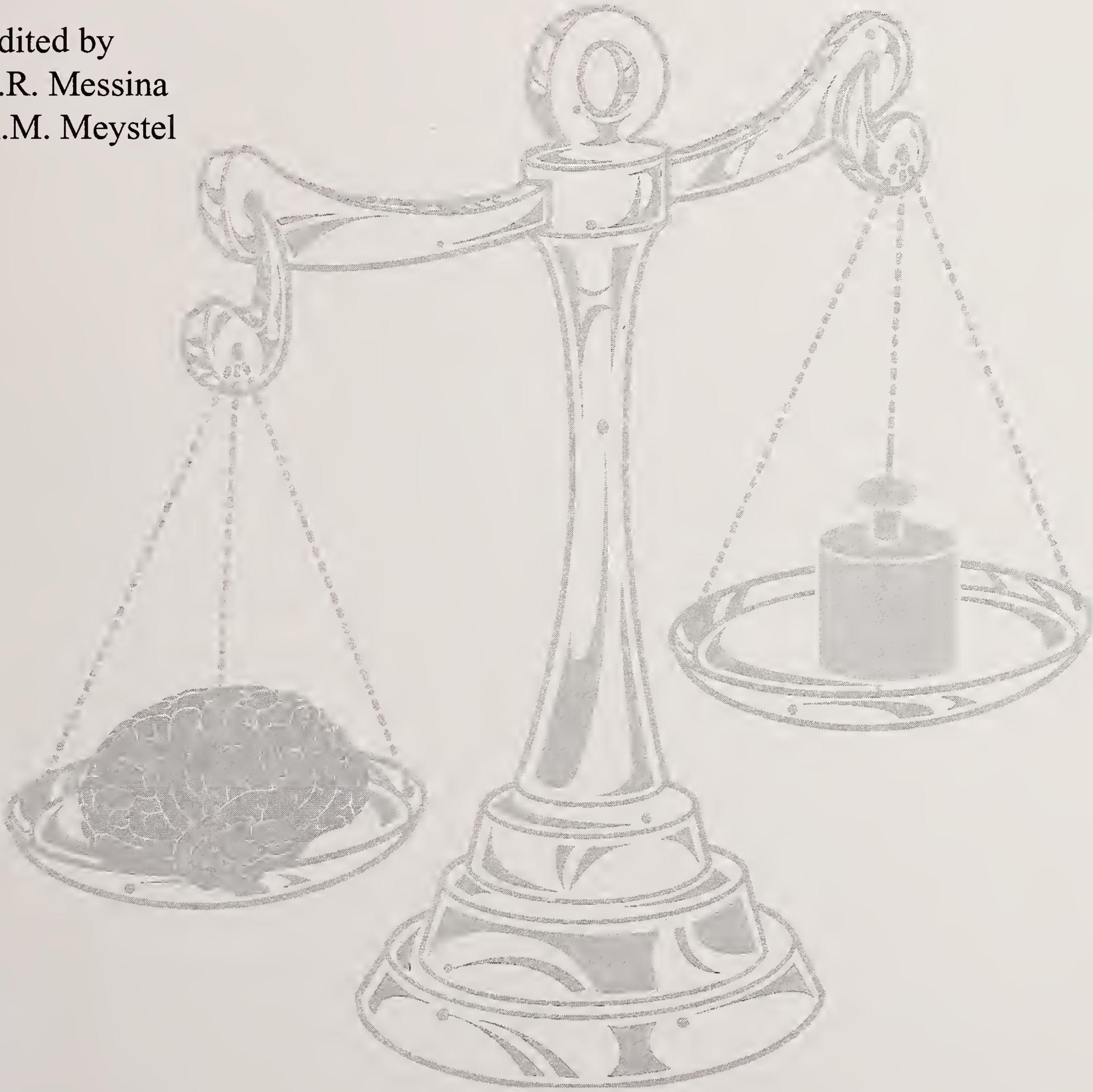
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NIST Special Publication 982

Measuring the Performance and Intelligence of Systems: Proceedings of the 2001 PerMIS Workshop September 4, 2001

Edited by
E.R. Messina
A.M. Meystel



NIST Special Publication 982

Measuring the Performance and Intelligence of Systems: Proceedings of the 2001 PerMIS Workshop September 4, 2001

Edited by
E.R. Messina
A.M. Meystel

Manufacturing Engineering Laboratory
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230

Issued June 2002

Co-Sponsored by:
National Institute of Standards and Technology
Defense Advanced Research Projects Agency
Institute of Electrical and Electronic Engineers Control Systems Society

In Cooperation with:
Institute of Electrical and Electronic Engineers Neural Network Council

National Institute of Standards and Technology Special Publication 982
Natl. Inst. Stand. Technol. Spec. Publ. 982, 153 pages (January 2002)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 2002

For sale by the Superintendent of Documents, U.S. Government Printing Office
Internet: bookstore.gpo.gov — Phone: (202) 512-1800 — Fax: (202) 512-2250
Mail: Stop SSOP, Washington, DC 20402-0001

TABLE OF CONTENTS

PART I

MEASURING PERFORMANCE AND INTELLIGENCE OF INTELLIGENT SYSTEMS

The White Paper

PERMIS'2001

White Paper

Measuring Performance and Intelligence of Intelligent Systems[1]

1. Is Testing of Intelligent Systems different from Testing of Non-intelligent Systems?

Testing of performance pertains to evaluation of the potential and actual capabilities of a system to satisfy the expectations of the designer and the users via exploration of its functioning. This includes determining how well the system performs its declared "job," how efficiently and effectively it does so, how robust it is, and so forth. The "job" and expected performance must therefore be defined at the outset. Efficiency is defined as how well the system does things right, effectiveness is defined as how well the system does the right thing , and robustness is defined as "the degree to which a system ... can function correctly in the presence of invalid inputs or stressful environmental conditions." [Finklestein, 00]

Furthermore, the tests under consideration are not meant to be broad-based general evaluations of the system's knowledge or the full spectrum of its capabilities. In particular, we are not striving to ascertain whether a system has common-sense generic knowledge applicable to general-purpose problem solving. The system being evaluated has a given sphere of responsibility and known abilities and tasks that it is able to undertake under its specifications.

Comments regarding the testing of intelligent versus non-intelligent systems are not meant to underestimate the difficulty of testing non-intelligent systems. Testing robustness, efficiency, and even functionality of *non-intelligent* software systems is difficult enough, e.g., see [Mukherjee 97]. Since the software execution can follow a myriad of combinations of paths through the code, it is impossible, in typical practice to exhaustively test all the possible combinations. In non-deterministic real-time systems, the problem is compounded by the uncertainty in the execution times of various processes, the sequence of events, asynchronous interrupts, etc [Butler, 93].

In general, the evaluation of intelligent systems (IS's) is broader than testing of non-intelligent systems (NIS). A system that has intelligence should in general be able to perform under a wider range of operating conditions than one that does not have intelligence. In fact, it should learn from its experiences and either improve its results within the same operating conditions or extend its range of acceptable conditions. What does this mean? Let's look at the main elements typically found in an intelligent system: Behavior Generation, Sensory Processing, and World Modeling (Knowledge Representation) [Meystel, 00].

2. Behavior Generation

Dealing With General and/or Incomplete Commands

An IS is given a job to do (task, mission, set of commands). *The job definition for IS is expected to be less specific than in an NIS.* A system with intelligence ought to have the capability to interpret incomplete commands, understand a higher level, more abstract commands and to supplement the given command with additional information that helps to generate more specific plans internally. The IS should understand the context within which the command is given. For example, instead of telling a mobile robot to go to a specific location in world coordinates "GO_TO(X, Y)," the command could be "Go to the window nearest to me." The robot should understand what a window is and know that it needs to find one which is the minimum distance away from me and move to that location. It also has a nominal proximity that it maintains from the goal location. Notice, that the command did not determine how close the robot needs to

[1] This paper is written by E. Messina, A. Meystel, and L. Reeker.

get to the window. It is expected that the robot knows where to stop the motion in similar cases, or the distance from the window should allow for convenient performance of other, or consequent movements.

Ability to Synthesize the Alternatives of Decisions and to Choose the Best One

There was time, when the processes of decision making and planning were understood and reproduced as choosing from the preprogrammed lists and menus. This time has passed. Now, it became clear that most of the decisions should be synthesized on line. It becomes increasingly clear that most of the planning procedures require searching. It was discovered that the advantages of search algorithms can be achieved when the space is represented and search is organized in a multiresolutional fashion. (See Meystel, 98).

Ability to Adjust Plans, Reschedule, and Re-plan

All job definitions interpretable by IS should be more abstract than would be given to an NIS. The command may encapsulate multiple individual actions, but it is the IS's business to figure that out. A mobile robot could be told to get the necessary signatures for a document. (This assumes that electronic signatures on the document are not an option.). The robot would have to understand which signatures are necessary (for example, if this is for a purchase, the purchase amount dictates what level of management needs to sign off), locate the individuals, interact with them to ask for their signature, and perform the intricate physical maneuvers necessary to present the document for signature. The individuals might not be in their office, hence the robot may have to search for them in alternative locations or try to arrange to meet them at some other time (re-scheduling). If someone is out of the office, the robot will have to decide whether to get the signature from someone else with equivalent signature authority or wait until the original person returns. Contrast this type of behavior with explicit instructions where the individuals and their locations are precisely given. If one of the individuals is not available, a non-intelligent robot would have to consult its human supervisor about how to proceed next.

The ability to adjust plans (re-plan) when the original ones are no longer valid is another crucial aspect that must be considered. It is one thing to create very elaborate plans to carry out a task (and the plans may even be derived from high level, abstract commands), but it is another matter to be able to deal with situations that are not as anticipated. Therefore, the intelligent system must be tolerant of changes as it is executing its plan and be able to react to the changes. In the bureaucratic robot introduced above, the change may occur if the vice president refuses to sign until he is given more information. The robot would then create another set of plans for itself to address the request, going to the originating individual to get background information or to the web to print out the specifications of the system being purchased along with alternatives that were not chosen. It would return to the vice president and present the information, and proceed to reintroduce the document to be signed. Obviously, all of this requires using appropriate architectures of knowledge representation, in particular, appropriate ontologies, as discussed in the subsequent sections.

3. Sensory Processing

Choosing the adequate set of sensors

The system receives signals from the real world through whatever sensors it may have. Note that a system may inhabit a software world, in which case "sensing" involves perceiving what exists external to itself, even if that is additional pieces of software. It must determine how to interpret the sensed signals in order to accomplish its tasks: the required actions are not prescribed in advance. Multiple sensors may be necessary and the system must be able to fuse information from them, collecting them into a registered, meaningful world model. Different sensors may give conflicting reports reports due to different interpretations of the world given their sensing modalities. Sensors may fail in certain circumstances or give insufficient information. The intelligent system should determine that it needs to utilize an additional or different sensor or process the signals it has differently. For example, it may be using a range sensor and a CCD camera as it navigates a house. It may hypothesize that instead of facing a wall or door, it may be confronted with a curtain hung in a doorway. In this case, it may need to apply additional or different processing algorithms in order to see if it can discern fabric (or something soft) from a planar, rigid surface. It may have to utilize a tactile sensor, if one is available.

Recognizing the unexpected

A system with intelligence (IS) ultimately must understand what its sensors are discerning. It must perform all of the requisite sensor or image processing to identify items in its environment to the level appropriate to the task. The requirements to processing will vary, depending on the situation and task. It may need to distinguish between certain types of tall weeds if it is an off-road vehicle, and it can drive only through certain leafy plants (not woody ones), or it would look unintelligent if it skirts around patches of tall grass. However, if it is a civilian car that should stay on roads, it probably doesn't need to identify what type of vegetation is growing on the side of the road, just that it is vegetation and not likely to jump out into the middle of the road. It will be directed by the behaviors to look for specific objects it may need in order to localize itself or find the object it is to act on. For example, it may look for a specific intersection as it navigates around a city or it may try to find a specific tool. The system's perception algorithms will have to be tolerant of a wide variation in the location and appearance of objects. Not all chairs look alike. A wrench may be on the floor or on a table, in a random position. Contrast this with non-intelligent systems that have limited tolerance for variations in their surroundings or in the objects with which they interact.

Dealing with unknown phenomena

The intelligent system will have to perceive entities and objects as it encounters them. It will classify and recognize items in its field(s) of view. It may classify a portion of the space in front of itself as a chair, or may have to deal with this as with an unknown object that might be interpreted as an obstacle. The sensory processing system, in conjunction with the world modeling system, must therefore know what it doesn't know about, and determine whether it needs to focus attention on the unknown in order to classify and identify. This ability to recognize the functional implications of unknown objects should be one of the major properties of IS. It is not impossible (in the future) to integrate multiple perceptions of an unknown object in various situations and eventually label it and deal with it as with a regular "known" object. Movements of unknown blobs can be interpreted with implication to possible planned maneuvers of the robot under consideration.

Multiresolutional Sensory Processing

The intelligent system will have to perceive entities and objects as it encounters them. However, sensory processing typically would require considering representation at multiple level of resolution. In all cases it provides for efficient computing. It is possible to demonstrate that this would correspond to the multiresolutional systems of knowledge representation (multiresolutional ontologies) and multiresolutional systems of decision making (multiresolutional planning) [Messina, 00].

4. World Modeling

Knowledge Representation

In most intelligent systems, an internal model of the world and/or a long-term knowledge store are utilized as a part of the overall knowledge representation system (KR). The long-term knowledge store (repository, or knowledge base) contains fairly invariant information, such as street maps or machining rules. An enabling aspect of the system's intelligence is the a priori knowledge it has and knows how to use. The internal model of the world is used to formulate a subset of KR that would allow the robot for planning expeditiously the required responses to the environment and situation. The sensory processes (discussed above) update and populate the current world model. The model might not be a single, monolithic one, but should rather comprise a set containing different types of information and/or different representations of perhaps the same information. The long-term knowledge may have to be merged with the in situ generated knowledge. For instance, the local sensors detect a road and some landmarks, such as buildings (using the knowledge base maps). The knowledge base supplies the name of the road, which is kept in the current world model.

The locally sensed information is obviously more current than that in the long-term store. Therefore, it must supercede what is in the knowledge base if there's a conflict. If a road has been closed, the system will plan around it and should, if appropriate, update the long-term maps. Obviously, these processes of updating our knowledge of the world belong to different levels of granularity, require different scale for interpretation and serve for supporting different resolutions of planning. It becomes a commonplace that most of intelligent systems either have or can be substantially improved by using multiresolutional systems of representation (including multiresolutional ontologies).

Multiple types of information

The intelligent system must be able to utilize a variety of types of information about the world in which it is functioning. If it is mobile, it must understand 2D or 3D space and have an adequate representation that enables it to move to the desired location efficiently while avoiding obstacles. It may need to take into consideration aspects beyond simple support surface (terrain or floor) geometry and obstacles. The type of terrain and traversability characteristics may be important as it determines which way it can go and how difficult it will be. So, for instance, if maintaining line-of-sight with a communications station may be necessary, the IS must be able to model the world so that it can perform the supporting computations to plan its movements.

Commonsense knowledge

An intelligent system should be able to have generic models available that guide it as it interacts with the world. This is as opposed to non-intelligent systems, where the environment is constrained to fit within the system's expectations (limited knowledge about *what is possible*). Although all possible situations cannot be predicted, the system should be prepared to handle many of them by a sub-store of *commonsense knowledge*. For example, the system may have to recognize and model stairs and elevators if it needs to go between floors. Not all stairs have the same geometry or configuration. It must know how elevators work, if that is appropriate to its job, namely, how to call an elevator, determine that one is available going in the right direction, selecting the floor, waiting until the right floor is reached and the door is open, etc. There is a general model of how to use an elevator, but there is tremendous variability in the actual elevator experience. The intelligent system has to be able to map between the generic and the specific.

Knowledge Acquisition: Updating, Extrapolating, and Learning

The updating of all sub-stores is conducted as the new information arrives. This information is frequently incomplete as far as satisfying the documents and models used by IS. An intelligent system must also be able to fill in gaps in its knowledge. If a moving object appears behind a robotic vehicle, the vehicle notes that it has an unknown entity that must be identified. Is it an emergency vehicle that must be given the right of way or an aggressive driver? It has to extrapolate or interpolate based on what it knows and what it discovers. All these knowledge acquisition activities require taking into account the uncertainty about what it does know. When driving down a road, if it is about to crest a hill, it cannot see the road beyond the hill. Rather than stopping, it should be able to assume that the road continues, and extrapolate based on the local geometry to forecast where the road exists even if it can't see it.

Related to this is the concept of predicting what will happen in the future. A machine tool that has a model of tool wear should forecast when a particular cutter will need to be replaced. A mobile vehicle will have to estimate its own trajectory and that of others with which it could potentially collide. The multiresolutional planning processes use various horizons of anticipation (larger at lower resolution and smaller at higher resolution.

The ability to anticipate will be amplified by learning new phenomena and control rules from experience. An intelligent system should become better at performing its job as it learns from its experiences. Therefore, one aspect that should be part of the testing or evaluation is the evolution and improvement in the system's functioning. The IS should have an internal measure of success as it performs its job. It can use the measure to evaluate how well a particular approach or strategy worked. Just as humans build expertise and become more efficient and effective at doing a certain job, the intelligent systems should have some means of improving their performance as well.

Requirements for Testing Intelligent Systems

Based on the discussion above, there is an initial set of requirements for testing intelligent systems that arise. The tests should therefore be designed to measure or identify at least the following abilities:

1. to interpret high level, abstract, and vague commands and convert them into a series of actionable plans
2. to autonomously make decisions as it is carrying out its plans
3. to re-plan while executing its plans and adapt to changes in the situation
4. to register sensed information with its location in the world and with a priori data
5. to fuse data from multiple sensors, including resolution of conflicts
6. to handle imperfect data from sensors, sensor failure or sensor inadequacy for certain circumstances
7. to direct its sensors and processing algorithms at finding and identifying specific items or items within a particular class

8. to focus resources where appropriate
9. to handle a wide variation in surroundings or objects with which it interacts
10. to deal with a dynamic environment
11. to map the environment so that it can perform its job
12. to update its models of the world, both for short-term and potentially long-term
13. to understand generic concepts about the world that are relevant to its functioning and ability to apply them to specific situations
14. to deal with and model symbolic and situational concepts as well as geometry and attributes
15. to work with incomplete and imperfect knowledge by extrapolating, interpolating, or other means
16. to be able to predict events in the future or estimate future status
17. the ability to evaluate its own performance and improve

Most of the items on the list allow for a numerical evaluation. However, non-numerical domains play a substantial role in evaluating intelligence and performance of IS.

5. Performance Evaluation in Non-numerical Domains

This theme focuses upon the aspects of intelligent system performance that are not directly quantifiable, but which should be subject to meaningful comparison. An example of an analogous aspect of human performance is the term "intelligent" itself. The notion of quantifying intelligence has always been controversial, even though people regularly use terms that ascribe some degree of intelligence. Terms ranging from smart, intelligent, or clever to dumb, stupid, or idiotic, with all sorts of degrees between, express people's judgments. But of course, these are often arbitrary judgments, without any basis for comparison or consistency of application. The notion of IQ, based on the widely used tests, was intended as a means of providing some consistency and quantification, but is still controversial.

So how might we do measurements for machines of the virtues that we associate with intelligence? First, we have to encapsulate the notion of what we mean by intelligence a little better. From the previous section one can see that the following properties are tacitly considered to pertain to intelligent systems:

- the ability to deal with general and abstract information
- the ability to deduce particular cases from the general ones
- the ability to deal with incomplete information and assume the lacking components
- the ability to construct autonomously the alternative of decisions
- the ability to compare these alternatives and choose the best one
- the ability to adjust the plans in updated situation
- the ability to reschedule and re-plan in updated situation
- the ability to choose the set of sensors
- the ability to recognize the unexpected as well as the previously unknown phenomena
- the ability to cluster, classify and categorize the acquired information
- the ability to update, extrapolate and learn
- being equipped with storages of supportive knowledge, in particular, commonsense knowledge

Then we need to find consistent measurements of what we consider to be the characteristics for each item on the list. We want these characteristics, like characteristics of software system performance quality in general, to provide us with goals to strive for in developing systems.

Ideally, the characteristics of value would be even more than engineering goals. They would be theoretical constructs in a "science of the artificial" [Simon, 69] – in this case, the science of Artificial Intelligence, or (being more specific) in the science of knowledge representation. As with other scientific fields, the constructs would be used in models (generally called scientific theories when they have been combined with a means of generating hypotheses and the hypotheses have been tested enough that the models are widely trusted). Some theoretical constructs may be easily judged from behavior of systems ("surface constructs"), but as in natural sciences, they might also be deeply hidden from view, within very complex models ("deep constructs" [see Reeker, 00]). In general, the depth of the construct is determined by the level of resolution accepted in a particular representation. In a multiresolutional system of knowledge representation, each level of resolution can be characterized by a particular "depth of the construct." These phenomena find their implementation in Entity-Relational Networks of words that are organized in the multiresolutional hierarchies of ontologies [Meystel, 01].

From the standpoint of human cognition, the components of intelligence are hidden deeply in the models of Cognitive Science (an interdisciplinary part of Psychology, which is also a developing science). This is one reason that IQ is still controversial: The model that backs up the measures is not complete. But it has nevertheless been possible to endow IQ with some consistency that *ad hoc* descriptions do not have. This is because there is some consistency in measurement and some predictive value in terms of future human behavior. We would like this to be true for measures of intelligence in artificial systems, too, and it may turn out that we have a distinct advantage over the cognitive scientists. This advantage is that we can, so to speak "get into the heads" of intelligent artifacts more readily than we can with humans.

Ontologies and Reasons for Comparing Them in Intelligent Systems

How do we proceed to compare intelligent systems in these non-numerical areas? As a beginning, it is suggested that we look at what is the core of an intelligent system (maybe of a human as well as an intelligent computer program) – the way in which a system conceives of the world external to itself, the internal representation of *what is* and *what happens* in the world. This is what has come to be called an *ontology* in recent years. Ontologies are closely connected to a number of basic constructs that are highly relevant to the performance of an intelligent system. They are clearly of importance in planning, making decisions, learning, and communicating, as well as sensing and acting. An *ontology* is used in a computer program along with a *logic*. The "control" or dynamic aspects of that logic may be embedded in the computer program itself, or it may be in a special program that manipulates a knowledge base of logical formulas, or a database manipulation system.

Whether an ontology is used within a computer program (or even the requirements statement of a planned computer program), a database (and its associated programs), a knowledge based system, or an autonomous artificially intelligent system, the ontology is indeed an informational core. As the architecture of the knowledge repository, the ontology (ontologies) are multigranular (multiresolutional, multiscale) in their essence because of multiresolutional character of the meaning of words [Rieger, 01]. In integrating systems, the presence of a shared ontology is what will allow interoperability. The term can be applied to the world-view of a human, too (in fact, is derived from a human study) though it may be easier to elicit it from the machine, as remarked above. (A fact related to the "knowledge acquisition bottleneck".) Thus it is an aspect of intelligent behavior that we may be able to compare from one system to another and correlate with the more general notion of intelligence in a system.

Returning to the best attempts to date to measure human intelligence, it is worth noting that a human's individual ontology might be explanatory for human intelligence, so it is not surprising that there are indirect measures of ontologies on IQ tests and achievement tests. They may give us an idea as to how to proceed with this aspect of an intelligent system. To measure the breadth of the person's intelligence, is it useful to ask if some people have "broader" ontologies than others. That is, do they cover more areas, or more subjects, or more aspects, or more details. Should we expect that these broader ontologies will manifest themselves in, say, a scholastic aptitude test (which in turn correlates with IQ)? Does the "broader" ontology testifies for the *breadth* of intelligence? Would that broader ontology influence the ability of the intelligent system (including humans) to make better decisions? For people, the answers seem to be "yes". It is tempting to imply that for machines, as well.

Undoubtedly some people have ontologies that make more adequate, at least more accurate distinctions among different activities and objects that are present in the world (we can call this a "deeper" ontology"). That makes it possible for them to reason with more precision. In other words, the breadth and the depth of the ontology entails more powerful knowledge representation system. So the evaluation of ontologies is, to some extent at least, not unreasonable in gauging human cognitive performance. Is it a reasonable measure for machines? If so, how is the measure to be utilized? These are questions to be examined at PERMIS'2001.

A Human View of Ontology

In this subsection, we would like to describe a view of a human ontology further, with the purpose of expanding the analogy to intelligent systems.

Humans use their ontologies (ON) (and actually, the whole system of knowledge representation) to label, categorize, characterize, and compare everything -- every object, every action. If a human learns the meaning of some new entity, it is because a label for this thing is put into the knowledge representation (KR) system, and eventually into a place in the ontology that relates it to the rest of the human's knowledge. If a human learns more about that entity, it is because more of its attributes, bounds, and relationships are specified in an Entity-Relational Network (ERN) of the knowledge representation (KR) where the ontology resides. The person does not have to bring all of its understanding of that same entity to conscious attention all the time, as it would be a distraction. So, the ontology is usually accessed only as much as needed to make the decision, or to communicate ideas and understand ideas communicated by others. Stripping off the details allows people to note resemblances and make comparisons.

A human's Knowledge Representation (KR) system (which the ontology provides some meaning) reflects reality to the extent that it helps the human to deal with the world external to the human's mind in a way that enables good decisions and accurate predictions. If it does not, the person should be able to change it so that it better reflects reality, by learning that enriches the ERN of KR. That is one way in which an organism worldview must depend on its experiences. The experiences themselves depend on actions that have been taken, sensory information that has been absorbed and communications that have been received and understood. Each person's ontology is therefore unique to that person, since each has different experiences, and maybe also different ways to learn from those of another person. Each discovers new ideas and makes new distinctions in ways that nobody fully comprehends and they become a part of my ERN-ON-KR system.

The relationship between the ontology and direct experiences of a sensory nature, coupled with activity and what it accomplishes is a part of the property called *grounding* which is a part of the process of *symbol grounding* [Harnad, 90]. When I learn language or learn the external world, this constantly extends my symbol grounding, since information might be conveyed that affects the ontology. There may be innate tendencies that provide symbol grounding, such as the fact that we can store information and access it and have a sense of sequence, but it is not our specific purpose to inquire about these.

The rational interpretation of things communicated to an individual (or discovered by one) is affected by and affects that individual's ontology. The organism may encounter "raw" pains, perceptions, and emotions that are not fully understood, but even these may be refined and contextualized by an existing ON. If an organism is to successfully communicate to others, it must encode, in a shared language, things that are in its ontology and shared to at least some degree in the ontologies of those receiving the communications. Questions, context, and conversations help to facilitate this sharing.

Decisions that lead to a high probability of success in dealing with the external world can only be made in the light of an individual's KR-based understanding of the facts surrounding the decision. If that individual does not have alternative actions characterized by information in an ontology, that individual cannot compare thee alternatives, and therefore cannot consider them in rational decision processes. If an organism's ontology does not reflect reality, the organism will make irrational and perhaps unsuccessful decisions. Complex decisions involve problem solving, and I must be able to access methods for solving problems.

The issue of such methods as part of ontologies is developed more deeply in a paper authored by Chandrasekaran, Josephson, and Benjamins [see Chandrasekaran, 99]. There it is pointed out that a decision-making system requires both a subject matter ontology and a problem solving method ontology. It is possible – and may be needed - to imagine even a larger ontology of activities.

If a person is to learn, it will be guided by the person's ontology in the learning process. Maybe natural linking mechanisms in sensory processes can be brought to bear in certain learning tasks, so a path through the woods or a list of words can be learned in seemingly built-in ways. This *rote learning* can be improved upon by relating items within an existing ontology. If a person is to classify items, it must be do so based on attributes, which are in the person's ontology. To search memory, that person needs to do so based on shared attributes, related activities, and other sorts of relationships. To learn by reinforcement, a system needs to associate the reinforcements with actions, objects, features, bounds, and relationships. To transfer learning from one task to another, it is necessary to use an ontology to find mappings from one action or object to another.

Objects in an ontology can be composed of other objects. An action may involve many objects (with their attributes, bounds and relationships) and other actions that somehow get "hooked together". An object may be defined by attributes that include defining actions.

Measuring Non-Numerical Aspects of Intelligent Systems Related to Ontologies

Can we exploit the idea of the human ontology above as a "core" of intelligence to characterize and compare intelligent behavior is machines based on a machine's ontology, built-in or acquired? Like a human, a machine may have sensors connected to subsystems of sensory processing. The machine may be able to take certain actions that provide grounding for the ontology. If it can learn, perhaps it can extend its ontology. How can we characterize that ontology in a way that will allow us to characterize the machine's capabilities? How can we characterize its ability to change the ontology? If it has an ability to communicate to other machines or people, how does this ability add to its capabilities (and to its ontology)? These are some of the ideas to be explored in PERMIS'2001.

6. Evaluation: Mathematical and Computational Premises

Consider a general situation: there is a set of goals (G_1,..., G_n) and a set of IS (or intelligent agents) to achieve these goals. Different intelligent systems, or agents might have different goals, or they might put different weights on the various goals. Further, they might be better or poorer at pursuing those goals in differing contexts. That is, they might have different components of intelligence (I_1, I_2,..I_s) and these would be more or less important in the different contexts (C_1,..., C_q) that should also be known.

This dependence on the context determines that agents might be good at one set of matters, but bad in others. The agent might be good at trying and learning about recognizing new objects in the surrounding world, but poor at doing anything risky. It is typical for humans to have a portfolio of "intelligences" as well as "goals." It would give some value to all the different goals, and would have some value to each dimension of intelligence. One agent might be characterized as an explorer, while another is very good in performing repetitive routines. Which agent should be evaluated as a preferable one? Obviously, this would depend on the goal and the context. An unequivocal answer might be impossible at a single level of resolution because the true result depends on the distribution of the types of agents and the contexts that the groups of agents find themselves in. Thus, the "intelligences" as well as "goals" might require representing them as a multiresolutional system.

A brief summary of the notation described then is
{G_1,..., G_n }- set of goals, i=1, ... , n
{IS_m} - set of intelligent agents to achieve these goals, p=1,...,m
{I_1, I_2,..I_s } - different components of the vector of intelligence, j=1, ... , s
{C_1,..., C_q} - different contexts, k=1, ..., q.
VI – vector of intelligence

where i are indices for goals and
j are indices for the components of the vector of intelligence

Multiresolutional Vector of Intelligence (MVI)

What should be measured to evaluate intelligence? The Multiresolutional Vector of Intelligence (MVI), and the level of success of the system functioning when this success is attributed to the intelligence of the system. The need to construct a MVI and determine their success emerges in many areas. It is not clear whether "success" is (or should be) correlated with "reward" and "punishment."

What constitutes the appropriate scope and levels of details in an ontology is practically driven by the purpose of the ontology. The ability to dynamically assume one level of detail among many possible details is important for an intelligent system. It might depend on the purpose of a system. In that sense the long term purpose of the system is different from its short term or middle term goals. Clearly, the long term purpose and the multiple term goals are goals belonging to different levels of resolution and should be treated in this way. This brings us back to the measures of intelligence through success: is intelligence to be measured by the ability of a system to succeed in carrying out its goals? Can the highly successful functioning at one level of resolution co-exist with the lack of success at another? Are the "successes "nested" or independent one from another?

Evaluation of intelligence requires our ability to judge the degree of *success* in a multiresolutional system of multiple intelligences working under multiple goals. This means that if success is defined as producing a summary of the situation (a generalized representation of it), the latter can be computed in a very non-intelligent manner especially if one is dealing with a relatively simple situation. Indeed, in primitive cases, the user might be satisfied by composing a summary defined as "list the objects and relationships among them" i.e. a subset of an entity-relational network (ERN). On the other hand, the summary can be produced intelligently by generalizing the list of objects and relationships to the required degree of quantitative compression with the required level of the context related *coherence.* Thus, *success* characterizes the level of *intelligence* if the notion of *success* is clearly defined.

The need in determining levels or gradations of intelligence is obvious: we must understand why the probability of success increases because somebody is supposed to provide for this increase, and somebody is supposed to pay for it. This is the primary goal of our effort in developing the metrics for intelligence. The problem is that we do not know yet is the basis for these gradations and are not too active in fighting this ignorance. What are these gradations, how should they be organized, what are their parameters that should be taken in account? We can introduce parameters such that each of the parameters affects the process of problem solving and serves to characterize the faculty of intelligence at the same time.

Multiresolutional Architecture of Ontology is a part of the Multiresolutional Vector of Intelligence. The following list of 25 items should be considered an example of the set of coordinates for a possible Multiresolutional Vector of Intelligence (MVI):

(a) memory temporal depth
(b) number of objects that can be stored (number of information units that can be handled)
(c) number of levels of granularity in the system of representation
(d) the vicinity of associative links taken in account during reasoning of a situation, or
(e) the density of associative links that can be measured by the average number of ER-links related to a particular object, or
(f) the vicinity of the object in which the linkages are assigned and stored (associative depth)
(g) the diameter of associations ball (circle)

The association depth does not necessarily work positively, to the advantage of the system. It can be detrimental for the system because if the number of associative links is excessively large the speed of problem solving can be substantially reduced. Thus, a new parameter can be introduced

(h) the ability to assign the optimum depth of associations

This is one more example of recognition that should be performed, in this case, within the knowledge representation system. Obviously, the ability "h" is tightly linked with the ability of IS to deal with incomplete commands and descriptions (see Section 1).

Functioning of the behavior generation module, for example, evokes additional parameters, properties and features:

(i) the horizon of extrapolation, and the horizon of planning at each level of resolution
(j) the response time
(This factor should not be confused with a horizon of prediction, or forecasting which should combine both planning and extrapolation of recognized tendencies).
(k) the size of the spatial scope of attention
(This corresponds to the vicinity of the associative links pertinent to the situation in the system of knowledge representation)
(l) properties and limitations of the aggregation and decomposition of conceptual units.

The latter would characterize the ability to synthesize alternatives of decisions and choosing one of them (see Section 1).

The following parameters of interest can be tentatively listed for the sensory processing module:
(m) the depth of details taken in account during the processes of recognition at a single level of resolution
(n) the number of levels of resolution that should be taken into account during the processes of recognition

(o) the ratio between the scales of adjacent and consecutive levels of resolution
(p) the size of the scope in the most rough scale and the minimum distinguishable unit in the most accurate (high resolution) scale

It might happen that recognition at a single level of resolution is more efficient computationally than if several levels of resolution are involved. A more fine system of *inner* multiple levels of resolution can be introduced at a particular level of resolution assigned for the overall system. The latter case is similar to the case of unnecessarily increasing the number of associative links during the organization of knowledge.

Spatio-temporal horizons in knowledge organization as well as behavior generation are supposed to be linked with spatio-temporal scopes admitted for running algorithms of generalization (e.g. clustering). Indeed, we do not cluster the whole world but only the subset of it which falls within our scope. This joint dependence of clustering on both spatial relations and the expectation of their temporal existence can lead to non-trivial results.

One should not forget that generalization (the ability to come up with a "gestalt" concept) is conducted by recognizing an object within the chaos of available spatio-temporal information, or a more general object within the multiplicity of less general ones. The system has to recognize such a representative object, event, or action if they are *entities*. If the scope of attention is too small, the system might not be able to recognize the entity that has boundaries beyond the scope of attention. However, if the scope is excessively large, then the system will perform a substantial and unnecessary job (of searching and tentatively grouping units of information with weak links to the units of importance).

Thus, any system should choose the value of the horizon of generalization (that is the scope of the procedure of *focusing of attention*) at each level of resolution (granularity, or scale).

All of these parameters characterize the realities of the world and the mechanisms of modeling that we apply to this world. These parameters do not affect the user's specifications of the problem to be solved in this system. The problem is usually formulated in the terms of hereditary modeling that might not coincide with the optimum modeling, or with the parameters of modeling accepted in the standard toolbox of a decision-maker.

The problem formulated by a user often presumes a particular history of the evolution of variables available for the needs of the intelligent system. Simultaneously, the user requests a particular spatio-temporal zone within which the solution of the problem is desirable. However, the input specifications often do not require a particular decomposition of the system into resolution levels and the intelligent system of CSA is free to select it in an "optimal" way. In other cases, the user comes up with already existing decomposition of the system that appeared historically and must not be changed (like the organizational hierarchy of a company and/or an Army unit). Sometimes, it is beneficial to combine both existing realistic resolution levels and the "optimal" resolution levels implied by the optimum problem solving processes.

The discrepancy between these decompositions requires a new parameter of intelligence

(q) an ability of problem solving intelligence to adjust its multi-scale organization to the hereditary hierarchy of the system, this property can be called "a flexibility of intelligence"; this property characterizes the ability of the system focus its resources around proper domains of information.

In the list of specifications of the problem the important parameters are

(r) dimensionality of the problem (the number of variables to be taken in account)
(s) accuracy of the variables
(t) coherence of the representation constructed upon these variables
For the part of the problem related to maintenance of the symbolic system, it is important to watch the
(u) limit on the quantity of texts available for the problem solver for extracting description of the system
and this is equally applicable for the cases where the problem is supposed to be solved either by a system developer, or by the intelligent system during its functioning.

(v) frequency of sampling and the dimensionality of the vector of sampling.

Most of the input knowledge arrives in the form of stories about the situation. These stories are organized as a narrative and can be considered *texts*. In engineering practice, the significance of the narrative is frequently (traditionally) discarded. Problem solvers use knowledge that has been already extracted from the text. How? Typically, this issue is never addressed. Now, the existing tools of text processing allow us to address this issue systematically and with a help of the computer tools of text processing

Finally, the user might have its vision of the cost-functions of his interest. This vision can be different from the vision of the problem solver. Usually, the problem solver will add to the user's cost-function of the system an additional cost-function that would characterize the time and/or complexity of computations, and eventually the cost of solving the problem. Thus, additional parameters:

(w) cost-functions (cost-functionals)
(x) constraints upon all parameters
(y) cost-function of solving the problem

This contains many structural measures. We need to trace back from an externally perceived measure of "success" or intelligence to a structural requirement. E.g, the construction codes specify thickness of structural members, but these dimensions are related to the amount of weight to support – the performance goal is the lack of building collapse.

Important properties of the Intelligent Systems are their ability to learn from the available information about the system to be analyzed. This ability is determined by the ability to recognize regularities and irregularities within the available information. Both regularities and irregularities are transformed afterwards into the new units of information. The spatio-temporal horizons of Intelligent Systems turn out to be critical for these processes of recognition and learning.

Metrics for intelligence are expected to integrate all of these parameters of intelligence in a comprehensive and quantitatively applicable form. Now, the set $\{VI_{ij}\}$ would allow us even to require a particular target vector of intelligence $\{VI_T\}$ and find the mapping $\{VI_T\} \rightarrow \{VI_{ij}\}$ and eventually, to raise an issue of design: how to construct an intelligent machine that will provide for a minimum cost (C) mapping.

$$[\{V_{PT}\} \rightarrow \{VI_{ij}\}] \rightarrow \min C$$

where

$\{VI_{ij}\}$ – vector of intelligence
$\{VI_{PT}\}$ – a particular target vector of intelligence (vector of intelligence that we are trying to develop within a system)

By the way, has this ever been done for the systems that are genuinely intelligent? Of course, this question is not related to design, just to measurement.

The Tools of Mathematics

The following areas of mathematics should be considered belonging
The following tools are known from the literature as proven theoretical and practical carriers of the properties of intelligence:

- Using Automata as a Generalized Model for Analysis, Design, and Control
- Applying Multiresolutional (Multiscale, Multigranular) Approach
  1. Resolution, Scale, Granulation: Methods of Interval Mathematics
  2. Grouping: Classification, Clustering, Aggregation
  3. Focusing of Attention
  4. Combinatorial Search
  5. Generalization
  6. Instantiation
- Reducing Computational Complexity
- Dealing with Uncertainty by

1. *Implanted compensation at a level (feedback controller)*
2. *Using Nested Fuzzy Models with multiscale error representation*

- Equipping the System with Knowledge Representation
- Learning and Reasoning Upon Representation
- Using bio-neuro-morphic methodologies
- General Properties of Reasoning
  —Quantitative as well as qualitative reasoning
  —Generation of limited suggestions, as well as temporal reasoning
  —Construction both direct and indirect chaining tautologies (inferences)
  —Employing non-monotonic as well as monotonic reasoning
  —Inferencing both from direct experiences as well as by analogy, and
  —Utilizing both certain as well as plausible reasoning in the form of
    1. *Qualitative Reasoning*
    2. *Theorem Proving*
    3. *Temporal Reasoning*
    4. *Nonmonotonic Reasoning*
    5. *Probabilistic Inference*
    6. *Possibilistic Inference*
    7. *Analogical Inference*
    8. *Plausible Reasoning: Abduction, Evidential Reasoning*
    9. *Neural, Fuzzy, and Neuro-Fuzzy Inferences*
    10. *Embedded Functions of an Agent: Comparison and Selection*

Each of the tools mentioned in the list allows for a number of comprehensive embodiments by using standard or advanced software and hardware modules. Thus a possibility of constructing a language of architectural modules can be considered for future efforts in this direction.

The Tools of Computational Intelligence

Proper testing procedures should be associated with the model of intelligence presumed in the particular case of intelligence evaluation. It seems to be meaningful to compare systems of intelligence that are equipped with similar tools. In this section we introduce the list of the tools that are known from the common industrial and research practice of running the systems with elements of autonomy and intelligence. It is also expected that these tools can be used as components of the intelligent systems architectures. Thus, they might help in developing and applying types of architectures that will be used for comparing intelligence of systems.

Learning.

We have separated this into an independent sub-section because of the synthetic nature of the matter. Learning is the underlying essence of all phenomena linked with functioning of an intelligent system. It uses all mathematical and computational tools outlined for all other subsystems. In the machine learning community, the attention is paid to three metrics: the ability to generalize, the performance level in the specific task being learned, and the speed of learning. From the intelligence point of view, the ability to generalize is the most important since the other two capabilities dwell on the ability to generalize. Systems can do rote learning, but without generalization, it is impossible, or at least very difficult to apply what has been learned to future situations. Of course, if two systems were equivalent in their ability to generalize, with the same resulting level of performance, then the one which could do this faster would be better.

7. References

[Butler, 93] Butler, R.W., and Finelli, G.B. The infeasibility of quantifying the reliability of life-critical real-time software. IEEE Trans. Softw. Eng 19, 1 (Jan. 1993), 3–12.

[Chandrasekaran, 98] Chandrasekaran, B., J. R. Josephson and V. R. Benjamins, "Ontology of Tasks and Methods," 1998 Banff Knowledge Acquisition Workshop. [Revised Version is cited above. It appears as two papers "What are ontologies and why do we need them?," *IEEE Intelligent Systems*, Jan/Feb 1999, 14(1); pp. 20-26; "Ontology of Task and Methods," *IEEE Intelligent Systems*, May/June, 1999.]

[Finklestein, 00] Finkelstein, R., "A Method for Evaluation the "IQ" of Intelligent Systems," Proceedings of the 2000 Performance Metrics for Intelligent Systems Workshop, Gaithersburg, MD, August, 2000.

[Harnad, 90] Harnad, S., "The Symbol Grounding Problem," *Physica* D, 42, 1990, p.335-46

[Meystel, 00] A. Meystel, Messina, E., "The Challenges of Intelligent Systems," Proc. of the IEEE Int'l Symposium on Intelligent Control, Patras, Greece, 2000

[Meystel, 98] Meystel, A., "Robot Path Planning," *The Encyclopedia of Electrical and Electronic Engineers*, Wiley, New York, 1998

[Meystel, 01] Meystel, A., "Principles of Word-based Knowledge Representation and Knowledge Processing," in ed. P. Wang, *Computing with Words*, Wiley, 2001, pp. 329-346

[Mukherjee 97] Mukherjee, A., Siewiorek, D. P.,Measuring software dependability by robustness benchmarking, Software Engineering, IEEE Transactions on , Volume: 23 Issue: 6 , June 1997, Page(s): 366 -378

[Reeker, 00] Reeker, L., "Theoretical Constructs and Measurement of Performance and Intelligence in Intelligent Systems," Proceedings of the Workshop on Performance Metrics for Intelligent Systems, August 14-16, 2000, NIST

[Rieger, 01] Rieger, B. B., "Computing Granular Word Meaning: A Fuzzy Linguistic Approach in Computational Semiotics," in ed. P. Wang, *Computing with Words*, Wiley, 2001, pp. 147-208

[Simon, 69] Simon, H. A. *The Sciences of the Artificial.* Third edition, Cambridge, MA, MIT Press, 1996. [First edition published 1969].

PART II

MULTIRESOLUTIONAL REPRESENTATION AND BEHAVIOR GENERATION: HOW DO THEY AFFECT THE PERFORMANCE OF INTELLIGENT SYSTEMS

Tutorial

Multiresolutional Representation and Behavior Generation: How Do They Affect the Performance of Intelligent Systems

A.Meystel
Professor, Drexel University
Guest Researcher, NIST

Abstract

In this tutorial, an outline of the theory of intelligent systems is presented as a sequence of the following issues. The term "Intelligent Systems" has a meaning implied by our usage of it within the domain related to the formidable phenomenon of Life and functioning of Living Creatures. However, neither for living creatures not for engineering devices this term cannot be presented through the list of functional properties and/or design specifications. Our theory is based upon two phenomena that should be considered in their interconnection: a) the existence of an Elementary Loop of Functioning (ELF) in all cases of systems with intelligence, and b) formation of Multiple Levels of Resolution (MR) as soon as ELF emerges. MR levels develop because of the mechanisms of joint Generalization and Instantiation due to the processes of grouping, focusing attention and combinatorial search (GFACS). The latter are explanatory for the subsystems of Learning/Imagining/Planning that are characteristic of all intelligent systems. This paper introduces the variety of mechanisms of disambiguation that pertain to functioning of intelligent systems.On the other hand, MR and ELF together lead to the development of Heterarchical Architectures. The above concepts are explanatory of the kinds of intelligence that are observed in reality and suggest how to test the performance of intelligent systems and what are the metrics that could be recommended.

1. Intelligent Systems: Invoking the Design Specifications

Multiple characterizations of intelligence and intelligent systems have been collected in [1, 2]. The meaning of the terms are instilled by our associations with human beings, or even with living creature in general. The desire to create similar properties in constructed systems has determined the tendency to *anthropomorphize* both faculties and functions gadgets and systems belonging to various domains of application. This starts with categorizing objects into ACTORS, or agents that produce changes in the state of the world by developing ACTIONS, and the OBJECTS OF ACTIONS, i. e. the objects upon which the ACTIONS are applied. ACTIONS are the descriptions of activities developed by the ACTORS.

Yet, this does not give an opportunity to exhaustively, or even simply adequately describe intelligent systems in the terms of design specifications. One reason for this is that specifications are never complete. They are never fully appreciated and understood either.

Example 1: Spot Welding Robot. These are the features that are frequently claimed for it:

- It has Basic Intelligence. The meaning of this assertion does not extend beyond simple salesman decorative phrase. Even in the universities, courses on binary logic and circuits with switches are called "Introduction to Intelligent Systems". Even a wall switch can be characterized as a carrier of intelligence of making the light "on" or "off".
- Programmed for specific task. Certainly the number of programmed functions is very limited in a robot. Yet, probably, any number of functions being pre-programmed is an evidence of intelligence (the one of the designer, the ability of the system to store information ("memorize things"). Memorization what should be done in a response to a particular command is considered a certain level of animal intelligence.
- No operator is needed. When you see this statement in the list of welding robot specifications, you should raise a question what is the quality of the results of welding comparing with welding by a human operator. Even now, the feedback system are

limited in their ability to eliminate the need in a good professional welder.

- Can only perform repetitive tasks without deviation from programmed parameters. No doubt about it: one should realize that this statement is rather a disclaimer than a claim of intelligent functioning.

Example 2: Mars Sojourner. The word "Mars" evokes associations of the machines of the future. However, no real faculties of intelligence could be listed (the welding robot was substantially "smarter").

- Remote Control – should not be considered a property of intelligence because by extending the distance between the operator and the machine we do not make the machine smarter, or more sophisticated, or capable of dealing with unexpected situations, or interpret illegible commands, etc.
- Light elements of autonomy. The specifications do not expand on this concept ("autonomy"). Probably, the ability to provide a feedback control can be (arguably) interpreted as elements of autonomy.
- Can Perform a variety of maneuvers (limited). This property seems to be similar to having preprogrammed functions.
- A particular maneuver is performed independently. All available maneuvers should be discussed and evaluated separately. Indeed, the maneuver of "turning right" and the maneuver "make a K-turn in a particular tight space" require different level of intelligence: from zero up to the substantial degree of perception-based autonomy.
- Not capable of deciding what to do next (no planning). Absence of "planning" in most cases means no intelligence.
- Problem:10 minute communication Lag Between earth and Mars (and probably, the guy does not know what to do next and does not dare to think about it!)

Example 3: Bomb Disposal Robot. This is another case of the device for remote performance (extention of capabilities of a human operator). These robots are called "intelligent" because of the importance of their mission, and also because the should be able to reproduce human movements with absolutely no mistakes.

- Remote Operation with high accuracy create the aura of respect. If the "increase in accuracy" could be claimed, this would be a very conspicuous demonstration of an intelligence.
- Requires very skilled operator. This is a claim of intelligence of the operator. However, it is an important assertion that this remote control device cannot substantially detriment the skills of the operator.
- Incapable of acting on its own (does not have any intelligence at all). This is related to most of the remote controlled devices.

Example 4. Intelligent Network. An example of the communication system with intelligent systems as the nodes of the network is shown in Figures 6 and 7 of [3]. The description of the communication network containing intelligent systems demonstrates that a) the concepts of closure within the intelligent node, b) multiresolutional distribution of information, and c) heterarchical networks are characteristic for this example. This was not observed in the Examples 1 through 3. Thus, one might assume that our dissatisfaction with Examples 1 through 3 was based upon an existing difference between classes of systems as far as the level of their intelligence is concerned.

In our further discussion, we will call all objects including ACTORS and OBJECTS OF ACTION by the term entity. The ACTION can be characterized and represented as a Discrete Event (DE). The concrete choice of the phenomena and objects as actors, DE and objects of action is determined by a combination of temporal and spatial resolution characteristic for a particular level. The structure of the object at a particular level of resolution is shown in Figure 1. The structure of the DE for a level of resolution can be introduced in a similar way. The structure is a recursive one because each "*part*" can be substituted by a similar structure, and the representation of objects will evolve into the high resolution domain. Similar evolution is possible into the low resolution domain: Figure 1 should be used for representing each of the *parents*.

Thinking about constructed intelligent systems brings the researcher to the ideas of autonomous robots that are capable of understanding incomplete assignments (commands), apply the general intention of the command to the particular situation at hand, etc. How about telling the robot: "Go to the window and alert me if something unexpected appears in the street?.." Apparently, this is the performance of an intelligent system that is justifiably expected in a market of intelligent systems soon enough. This popular demand is not far from its possible satisfaction. The designer's options include on-line or off-line learning from experience and using multiple tabulated alternatives together with efficient decision making procedures.

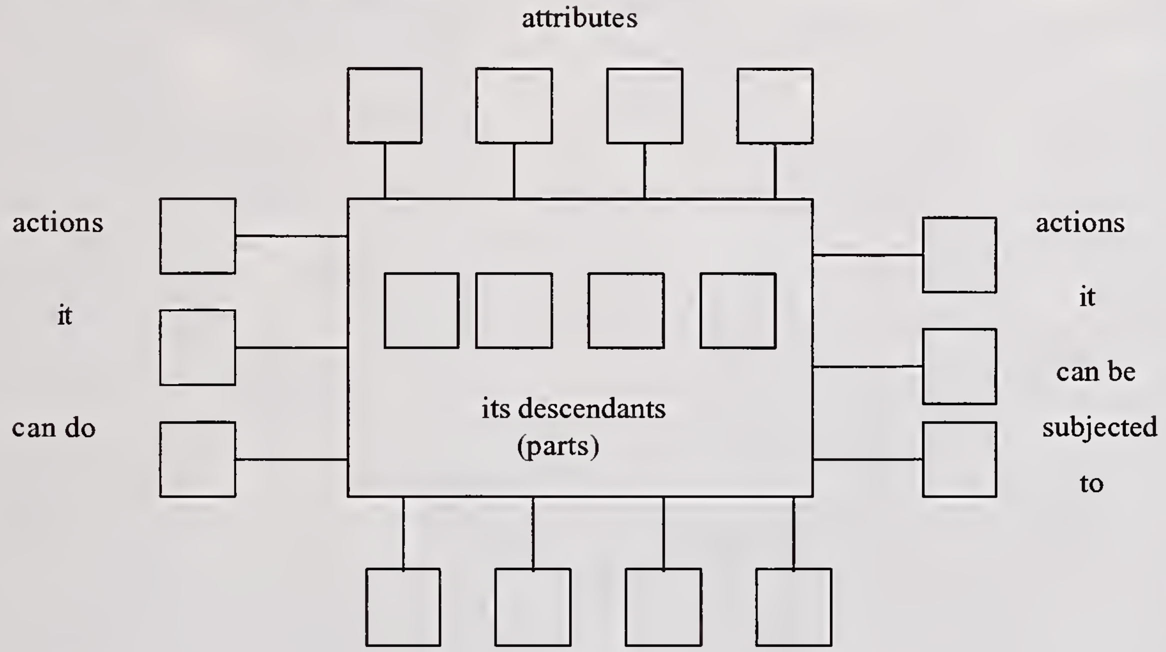


Figure 1. Structure of the Object

2. E L F: Elementary Loop of Functioning

The Law of Closure. Closure is the foremost property of Intelligent Systems (IS) and should be satisfied at all levels of its Architectures. The Elementary Loop of Functioning (ELF) of IS can be defined at each level of the IS and should be consistently closed in each communication link between the subsystems of ELF as described in [1, 2, 4]. Unlike the classical "feedback loop," the loop of ELF is not focused upon the deviation from the goal: it is focused upon the goal. As soon as we can explain for a particular scene and/or for a particular situation who are the ACTORS, what ACTIONS do they develop, and upon which OBJECTS OF ACTION their actions are applied – the Elementary Loop of Functioning has been found. In Figure 2. The subsystems of this loop determine basic properties of the intelligent system.

SENSORS (S) are characterized by their ultimate resolution and their scope of the information acquisition per unit of time. In SENSORY PROCESSING (SP), the primary clustering is performed (together with organization and bringing all available data to the total correspondence), and the resolution of clustered entities is evaluated. The WORLD MODEL,WM (or Knowledge Representation Repository, KRR) unifies the recently arrived and the earlier stored information within one model of representation that determines values of resolution for its subsets. Mapping the couples [goal, world model] into the sets of output commands is performed by BEHAVIOR GENERATION (BG) for the multiplicity of available ACTUATORS (A), actually maps the resolutions of the WORLD MODEL into the resolutions of output trajectory.

Closure of all these units (…➔W➔S➔SP➔WM➔BG➔A➔W➔…) is determined by the design of the system and the learning process of defining the languages of the ELF subsystems.

- The First Fundamental Property of Intelligent Systems Architectures (the property of the existence of intelligence), can be visualized in the law of ***forming the loop of closure.***

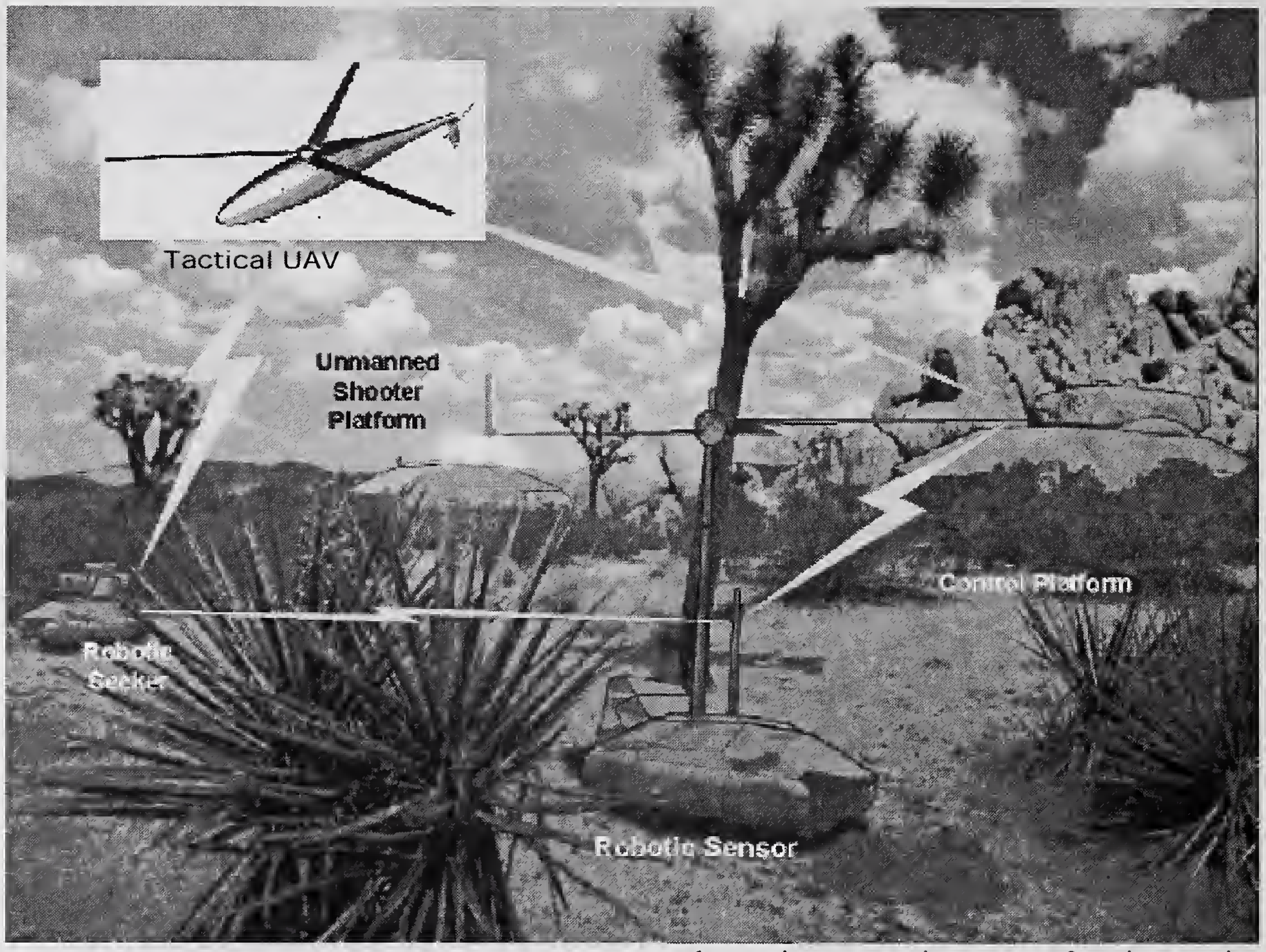


Figure 2. Design of a military situation (source: DARPA)

Closure is satisfied and the *consistency* of ELF holds when the unity of language (vocabulary and grammar) holds for each communication link between every pair of ELF subsystems.

- No matter what is the nature of the intelligent system, no matter what is the object-oriented domain under consideration, **the structure of closure is always the same.**

Statistical Closure. Functioning of the ELF cannot be impeccable because of noise and disturbances arriving from the external world and because of the errors of computations within ELF. Thus, as a result of mistakes, the property of closure is not satisfied impeccably. Thus, we should expect that only statistical closure can be satisfied reliably. The phenomenon of the time span between the "cause" and the "effect" is observed for both the closure of "in-level" functioning and the closure that is demonstrated for reduction of resolution when the information is integrated bottom-up. The following observations are important for interpreting reported information on the events in a system:

- The existence of closure at the lower (generalized) levels of resolution was considered a surprise and was even given a special term: "statistical closure" [5].

Now, it would not be difficult to understand that ***every closure is a statistical closure*** including closure reflected by the "in-level" functioning as well as closure obtained as a result of generalization of information tp the lower level of resolution.

- Obviously, there are no cause-effect events that happen simultaneously: if absence of the time span was reported, there is no basis for considering particular events of having "cause→effect" relationships.
- The time of any event is an ***integration*** of realistic or statistical results ***of the potential multiple experiments.*** This should be realized while determining whether the events are separated by a time span.

These observations can often protect us from a misinterpretation, but not in all cases. Even consistent ELFs are capable of generating misinterpretations related to causality. Example: it is known that 80% of patients with hip fracture die within a year not because of hip fracture complications but because they had another condition that brought them to fall (they had it prior to the hip fracture). Obviously, many of these misinterpretations ascend to the formation of the languages for the subsystems of an ELF. The *purpose* may not always be explicitly represented but it can always be explicated as the analysis of *causes.* Although, etiological analysis (contemplation of causes) is always presumed, it is seldom performed.

3. Levels of Resolution and Intentionality: Multiresolutional Analysis

We need to reduce the complexity of computations by grouping similar units (entities) into the larger formation that can satisfy the definition of an entity, too. The words "we need" are italicized because the issue of "need" is a critical one in the very emergence of this phenomenon: multiple levels of resolution. The *needed* entity is a "lower resolution" entity: the details of high resolution are unified together under a specific objective (representing the intentionality). The totality of lower resolution entities forms a "lower resolution world" of representation, or the "lower resolution level." Within the "scope of the world" considered at the higher resolution, we will have much smaller total number of entities, and for the same computational power, the scope of the world or the efficiency of computation can be substantially increased. This is why we are searching for the lower resolution entities and producing generalizations. Thus: the limitations in processing speed, memory size, and sensor resolution spur our creativity up.

There are numerous ways of representing information at a level of resolution. The most wide spread method presumes performing a sequence of the following steps as *the Algorithm of Information Organization*:

Step 1 (S1). Hypothesizing the entities within particular boundaries separating them from the background and other hypothesized entities. More than one hypothesis for an entity is expected to be introduced (a list of hypotheses is supposed to be formed and maintained)

Step 2. Searching for confirmation of the hypotheses {H} of Step 1 (HS1) and evaluation of current probabilities of HS1 being the "truth,"

Step 3. Hypothesizing a meanings of the hypothesized entities [$HS1_i \rightarrow M_i$]; call this couple "a meaningful entity." More than one hypothesis for the meaning is expected to be introduced (a list of hypotheses is supposed to be formed and maintained).

Step 4. For each hypothesized meaningful entity [$HS1_i \rightarrow M_i$] determine its plausible goal (objective)

{[$HS1_i \rightarrow M_i$] under the goal G}. This is associated with the ability to hypothesize (and verify) the "cause→effect" couples and hypothesize a purpose of events (etiological analysis).

Step 5. For each {[$HS1_i \rightarrow M_i$] under the goal G} determine its relationships with other meaningful entities of the "scene," going back to Steps 1 and 2; considering different hypotheses; converging to the maximum values of probabilities evaluation.

Step 6. Constructing the entity-relationship network for the scene (ERN_j)

Step 7. Search within ERN for islands-candidates for generalization into the entities of lower resolution. As the candidates has been determined consider them hypotheses of entities with particular boundaries similar to those mentioned in **Step 1** and **GO to Step 2**. If no new islands emerged, EXIT from the recursive search from entities and **GO to Step 8**.

Step 8. Submit the hierarchy of ERNs to World Model.

This sequence of steps can be applied to any type of information representation including visual, audio, verbal, etc. The sequence can be illustrated by using a set of multiresolutional images, for example, from [6].

One can see that some Logic is presumed to be introduced for dealing with the multiresolutional information at hand. Unlike the standard propositional and predicate calculi, this logic has to predicate various situations and related sub-situations by their goals (purposes, objectives) being important factors in the process of inference. We believe that the Intensional Logic of Entities (Objects) can be proposed for using in the system with multiresolutional ELFs. An important role is here allocated with the concept of alternative worlds (possible situations or possible worlds). This can be considered an extension of the known notion of the "world model". This allows looking for alternatives to

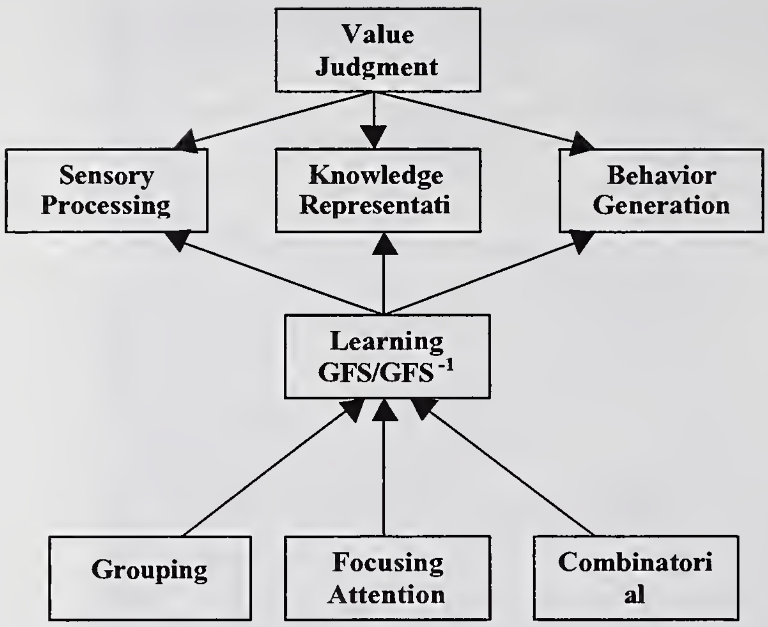


Figure 3. Combinatorics of GFACS/GFACS-1 functioning

the actual course of events in the world. On the other hand, adding the hypothesized purposes makes all statements *intentional* as well.

Intensional logic with explicated intentionality should become a basis for the introductory Multiresolutional Analysis (MA). The latter can be defined as constructing the representation and using it for the purposes of decision making. Using computational algorithms leads to taking advantage of representing the World as a set of sub-Worlds each with its individual scope and the level of detail.

The possibility and the need for MA is looming as can be seen from D. Dennett's Multiresolutional Stance where the property of considering many levels of resolution is being associated with intentionality:

> "To explain the intentionality of a system, we simply have to decompose the system into many, slightly less intelligent, subsystems. These subsystems can also be broken down into many more less intelligent subsystems. We can continue to break up these larger systems until eventually we find ourselves looking at individual neurons" [7].

Multiresolutional analysis boils down to purposeful development of multiresolutional heterarchies which

- protects us from paradoxes [e.g. of the pitfalls of self-referencing]
- allows for interlevel disambiguation
- determines true ontologies and definitions
- outlines symbol grounding activities

4. GFACS and GFACS^{-1}: Generalization and Instantiation by Using GFACS Operator

Both GFACS and GFACS-1 consist of the simpler procedures that are called "grouping", "focusing attention", and "combinatorial search". Most of the procedures that are being applied for computer vision and intelligent control systems are based upon the GFACS set of procedures. Examples: "Windowing" broadly aplied for selection of the representative part of the information set, is actually *searching* (combinatorially), CS. Masking irrelevant sub-entities is actually *focusing attention*, FA. On the other hand, the same "Windowing" contains a substantial component of "masking" and thus, can be interpreted as "focusing attention", FA in additional to searching combinatorially, CS. All algorithms of "clustering" can justifiably be interpreted as "grouping", G. Algorithms of "filtering" are "focusing attention", FA. Hypothesizing the entity in an image always includes all of the above: G, FA, CS.

4.1 Level-to-level Transformation: Generalizing by GFACS

The Algorithm of Information Organization presented above (see Section 3) contains the operator of generalization in its **Step 7**. It can be further decomposed into the following sub-steps:

7.1 Search within ERN for islands-candidates for generalization into the entities of lower resolution. This search will include forming tentative combinations of high resolution entities into sub-entities that allow for a consistent interpretation. Logic of this "combinatorial search" includes "focusing attention" upon the results of tentative "grouping" and determine properties of these tentative groups and their relations with each other.

7.2 As the candidates has been determined, finalize "grouping" and label the groups.

7.3 Consider these groups to be hypotheses of entities and analyze the corresponding ELFs.

Generalization is finished after the newly synthesized entity became a part of corresponding ERNs and ELFs.

4.2 Instantiations: GFACS^{-1}

In the inverse procedure, the system is searching for the plausible decomposition of a legitimate entity (that received a status of "group" as a result of prior "generalization"). Usually, this requires for performing several re-hypothesizing the components of entities and grouping them again to check whether they retain the meaning declared earlier. This features the following steps of instantiation: the hypotheses of instantiations are arriving from the adjacent level of lower resolution after hypothesizing (i.e. are arriving from "above") and should be verified by repeating the procedure of "grouping" at the level of higher resolution (i.e. "below"). In Figure 3, the richness of procedural capabilities is illustrated that is achieved in a single ELF as a result of GFACS/CFACS^{-1} functioning. From Figure 3, one can see that the generalization/instantiation couple can be considered a core of unsupervised learning [1]. This determines the need is a special logic of inference.

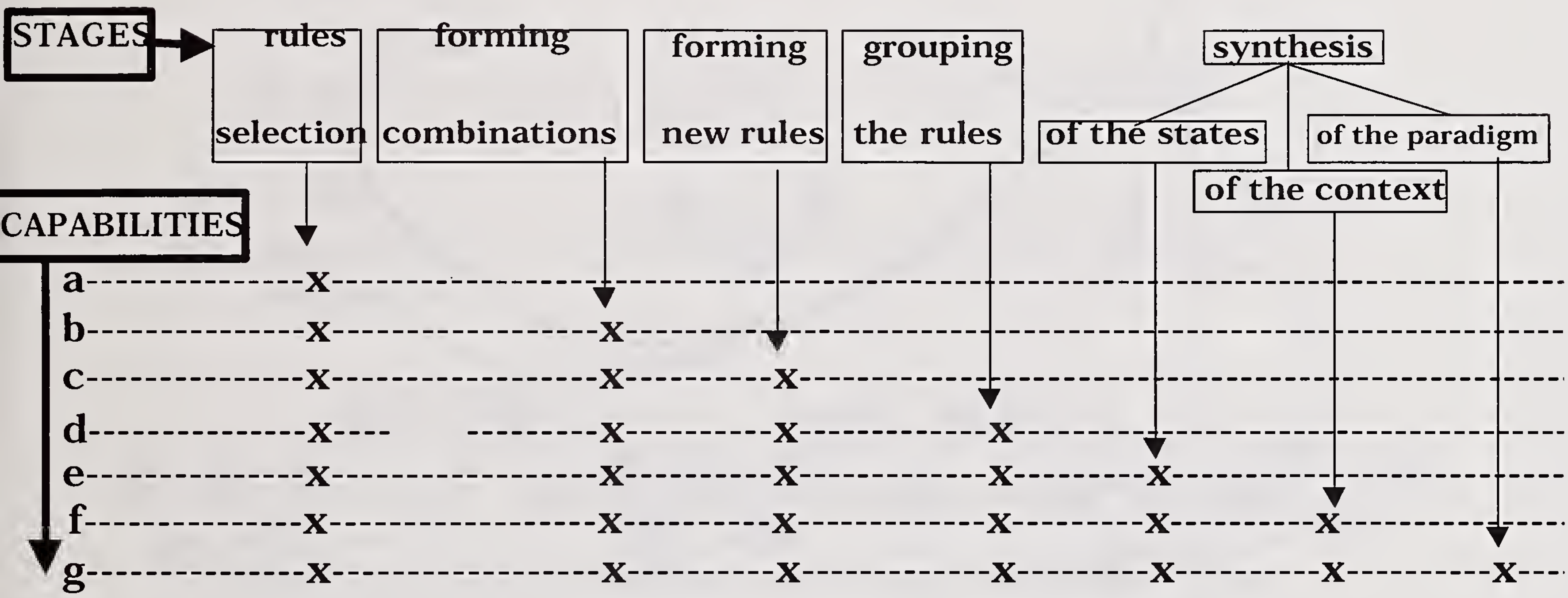


Figure 4. Logical Properties Acquired at Different Stages of the Intelligence Development

4.3 Advanced Logic Induced by Generalization/Instantiation

Indeed, the standard set of the inference tools taken from the arsenal of Propositional Calculus and Predicate Calculus of the 1st Order builds the inference processes primarily based on the undeniable conclusions that can be made from having a set of properties known for a particular class (ergo: belonging to this class), or conclusions that can be made from the fact of belonging to a particular class (ergo: having properties characteristic for this class). Forming new objects and/or new classes, growth of object and events hierarchies are new phenomena in the domain of inference. Even more powerful are the capabilities linked with new abilities to infer the purpose, construct hierarchies of goals, imply cause-effect relationships. In Figure 4, it is demonstrated that the introduction of logical capabilities and the enhancement of the ability to infer emerges as a result of incorporation of computational capabilities based upon equipping the system gradually by the new computational tools: including rule selection, forming combinations of rules, forming new rules (as a result of learning), grouping the ryles, forming combinations of the states and the context.

Unlike the symbolic logic that is supposed to be precise, free of ambiguity and clear in structure, the logic of multiresolutional system of ERN is limited in precision by the demands for ***associative disambiguation*** (see Section 7) that spreads into the adjacent levels of resolution (no *"logical atomism"* is presumed).

4.4 Learning, Imagining, and Planning: The Tools and Skills of Anticipation

Since the etiology enters the discussion, it would not be an exaggeration to state that the GFACS/CFACS^{-1} couples induce the knowledge of a Future, give the intelligent system the skill of anticipation. Thus, learning invokes imagining "what if" and various alternatives are being simulated to exercise the alternatives for estimating the Future and planning the Future as it was described and illustrated in [8] (see Figure 5). Actually, all types of intelligent processing of information are about the Future.

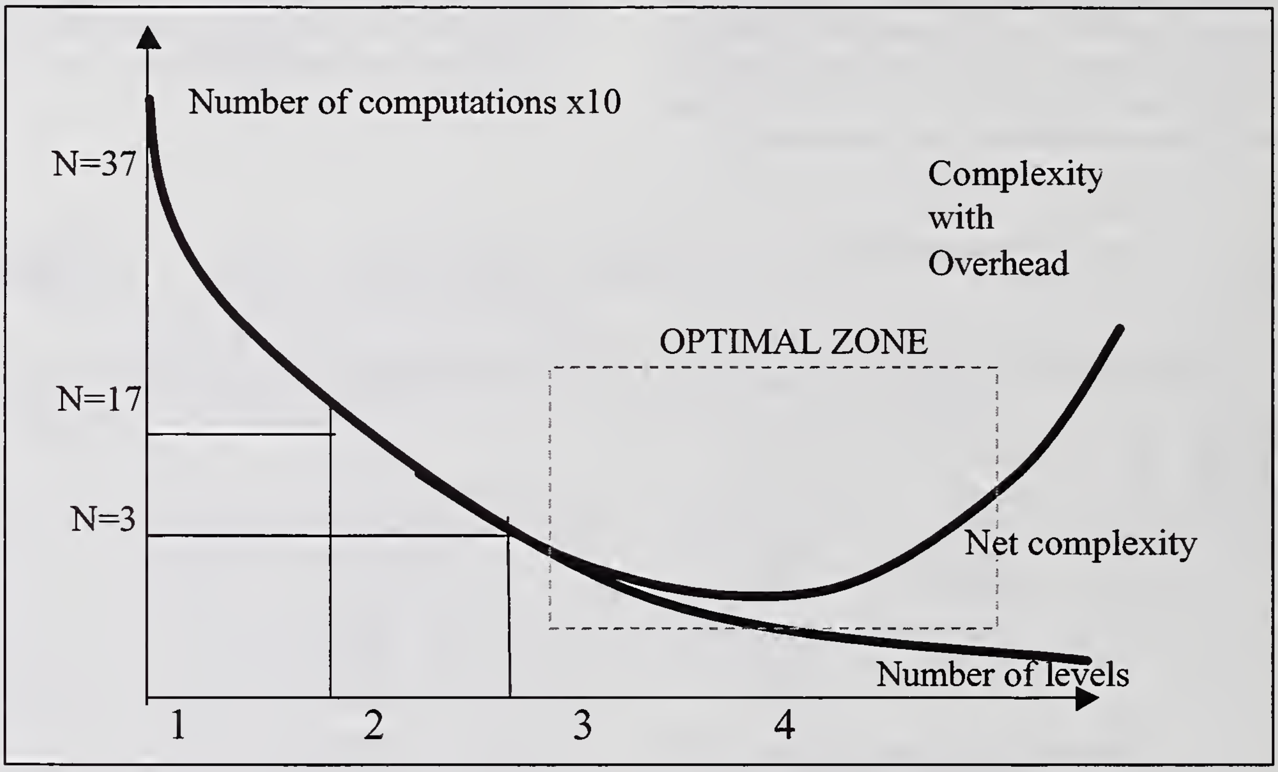


Figure 5. Computational complexity is reduced by introduction of additional levels of resolution

5. Intelligent Architectures and Kinds of Intelligence They Embody

5.1 More About Multiresolutional Combinatorial Search

Complexity in Multiscale Decision Support System depends on the number of levels of resolution. In Figure 5 the linkage between computational complexity and the number of resolution bottom up fits within the hierarchy of command, increase of the planning horizon and re-planning interval helps to bring the best properties of the system to a realization. The following are 4-D/RCS specifications for the planning horizon, re-planning interval, and reaction latency at all seven levels (see the table).

5.2 Existing Architectures

Multiresolutional processing is one of the important features of the reference architectures promulgated by NIST for application in levels of resolution is shown for a problem of path planning. The Example with DEMOIII would clarify how the levels of resolution differ in their parameters. Actually, lowering the intelligent systems. It is easily recognizable that heterarchies similar to shown in Figure 6 fit within the paradigm of large complex systems including intelligent autonomous robots, unmanned power plants, smart buildings, intelligent transportation systems including large automated bridges. It fits perfectly also to the DOD systems of command, control, communication and intelligence. It is characteristic of heterarchies that while having top-down and bottom-up hierarchical components, they are not hierarchies:

Table of specifications for parameters of multiresolutional planning in DEMOIII [1]

Level	Planning horizon	Replan interval	Reaction latency
1 Servo	50 milliseconds	50 milliseconds	20 milliseconds
2 Primitive	500 milliseconds	50 milliseconds	50 milliseconds
3 Subsystem	5 seconds	500 milliseconds	200 milliseconds
4 Vehicle	50 seconds	5 seconds	500 milliseconds
5 Section	10 minutes	1 minute	2 seconds
6 Platoon	2 hours	10 minutes	5 seconds
7 Battalion	24 hours	2 hours	20 seconds

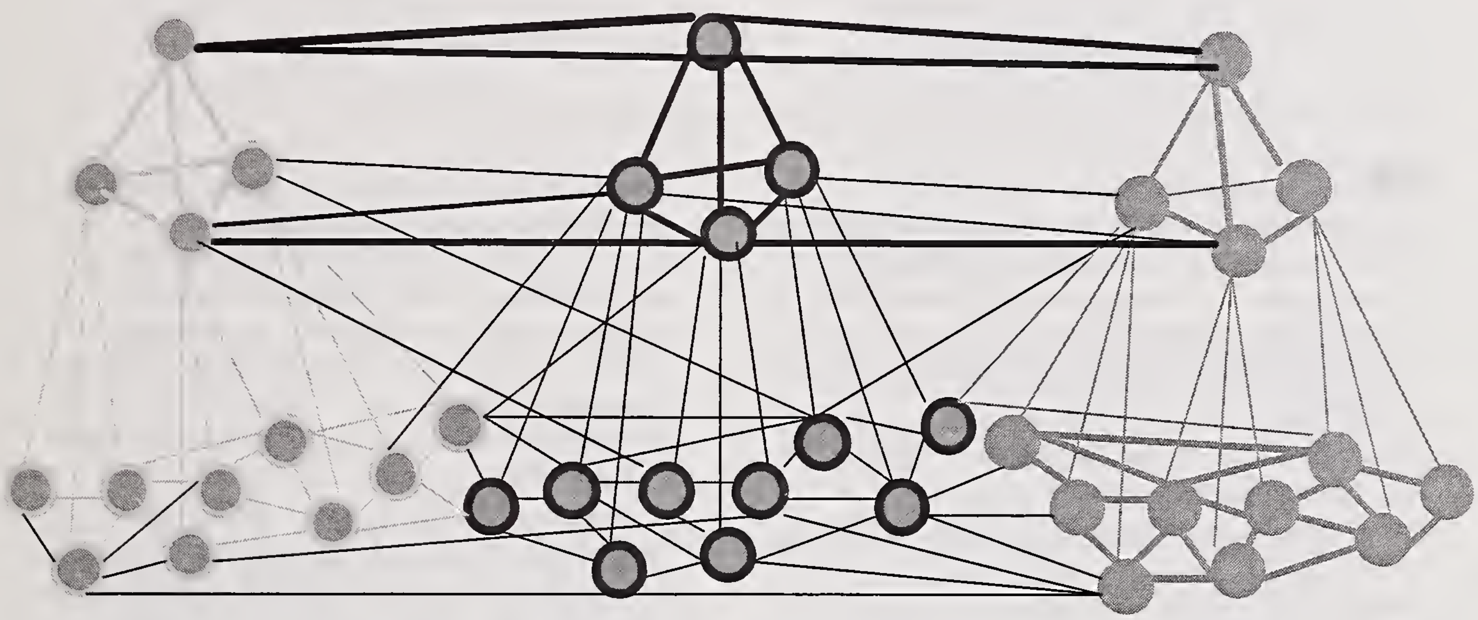


Figure 6. A Community of Interacting Heterarchies

heterarchies are not tree architectures. However, in each heterarchy, a multiplicity of hierarchies can be discovered and employed including heterarchies of Top/Down-Bottom/Up Processing heterarchies of "In-Level" Processing, and others. Similar relationships and transformations are characteristic of Entity-Relational Networks (ERN) that are obtained from semantic networks for using in Knowledge Representation Repositories.

5.3 Kinds of Intelligence

General Intelligence

Many and equally unclear definitions are known from the literature. We refer here to two definitions that seem to be both applicable and instrumental ones.

Definition 1 (Internal)

"An intelligent system has the ability to act appropriately in an uncertain environment, where an appropriate action is that which increase the probability of success, and success is the achievement of behavioral subgoals that support the system's ultimate goal" [9].

Definition 2 (External)

"Intelligence is a property of the system that emerges when the procedures of direct and inverse
generalization (including focusing attention, combinatorial search, and grouping) transform the available information in order to produce the process of successful system functioning." [8].

These definitions should be supplemented by a description of the trade-off to be achieved by any intelligent systems no matter whether they are oriented a) toward the goal achievement (articulation), b) toward sustaining oneself [realization of self], or c) toward "feeling better" (avoiding paradoxes, antinomies, contradictions). The trade-off is illustrated in the diagram 7.

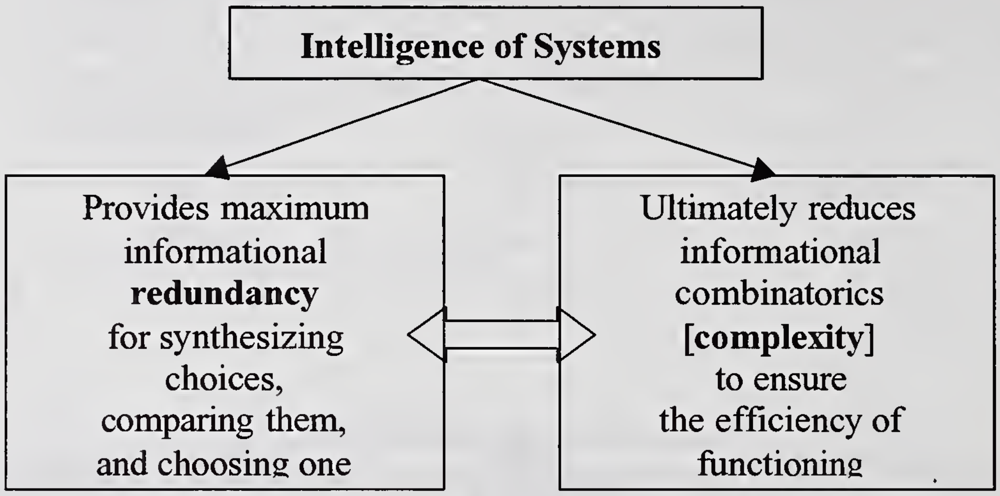


Figure 7. Trade-off achieved by intelligence of systems

Proprioceptive Intelligence
A special kind of intelligence presumes blending the carriers of elements of ELF into an inseparable construction. Proprioceptive intelligence presumes blending sensing devices with actuators of a system. This gives additional properties:

- An ability to modify behavior to maintain feeling comfortable
- An ability to use the working part of a system as a carrier of information

Contemplative Intelligence
All architectures of intelligence considered above are oriented toward pursuing clearly discernible objectives. In some situations this is not the case. The following activities are characteristic for a contemplkative intelligence: it ponders [thoroughly], theorises, cogitates, inquires, ruminates [repetitively], speculates, conjectures, deliberates [in the latter case, the intentionality is a primary issue].

6. Testing the Performance and Intelligence

The general lessons of the existing experience in testing performance of systems can be formulated as follows.

•Performance can be different for IS and non-IS. Breaches in communication that are taken care by human operators in non-IS, are covered by automated sub-systems in IS. However, all expected cases might not be reflected in the pre-programmed menu. Thus, learning is the only way to compensate for the inadequate pre-programming. Nevertheless, the failures in representation are expected to endanger the quality of operation even in the most intelligent systems. Another cause of the inevitable failures is the incomplete or inadequate goal specifications.

• We already discussed the fact that the main advantage of the intelligence is giving the ability to deal with unexpected predicaments. Because of this, the main advantages power that intelligence brings to the system is unspecified (and probably, unspecifiable). It should not be forgotten that many hings are NOT and frequently CANNOT be specified.

6.1 Testing Generic Capabilities of Intelligent Systems

The following capabilities can be checked and statistically validated via experimental testing in a functioning system on-line.

- All terms from the assignment are supposed to be supported by the high resolution, low resolution and associative knowledge.
- Each level must demonstrate its ELF consistency. Standard testing scenario can be constructed and exercised.
- Functioning is presumed the ability to work under incomplete assignment (including incomplete statement of what should be minimized or maximized).

- Functioning should be possible under not totally understandable assignment.
- Functioning should be possible under not totally interpretable situation.

6.2 Skills that can be checked off-line

Off-line testing allows for enabling better preparedness of the system for critical situations.

- Multiple channels of enabling functions (allows working under a condition that a part of the capabilities is disabled).
- The existence of the internal model of the world that is capable of planning and developing "the best" responses to the changing environment and dynamic situation by using simulated system.
- The ability to learn from experience of functioning: learning can be verified prior to the future situations of functioning..
- The ability to judge the richness of the MR ontologies. Indeed, the vocabularies and grammars of all levels allow for shaping and refining them prior to real operation.
- The ability to re-plan and/or adjust plans in important when the original ones are no longer valid; this is another crucial aspect that must be evaluated.

6.3 Understanding "Commander's Intent"

One of the important functions of intelligence is restoring of the *intent* of the node that is the source of the goal. In other words, a system with intelligence ought to have the capability to understand its higher level, i.e. the lower resolution level (where the "supervisor" or "commander" is situated). The incoming "goal" is frequently presented rather as an abstract combination of terms. The system should be capable of supplementing the submitted command with additional information (sometimes, contextual) that helps to generate more specific plans internally. This is almost equivalent to creating the goals for itself: the elements of future autonomy emerge in the intelligent systems as tools of performance improvement.

7. Conducting Disambiguation

We have addressed the need to verify the consistency of statements generated at a level by their compatibility with the adjacent levels above and below. Clearly, they should not violate generalizations creating objects and events of the level above, and the results of decomposition of the entities and events at a level of consideration should not violate consistency of the higher resolution representation and decision making.

The following capabilities are expected from the system of disambiguation.

- 1. Hypotheses should be formulated of generalizations for the upper level and instantiations for the lower level. These hypotheses are obtained by GFACS and $CFACS^{-1}$ within the context of the situation represented by the ELFs of three adjacent level under consideration.
- 2. When the hypotheses generation is completed (a ranked list of hypotheses is constructed) the consistency of the hypotheses should be verified an the i-th, [i+1]-th and [i-1]-th levels. Verification is done by checking whether the closure of each ELF still holds. This operation is an example of creating the "Tarsky's Hierarchy" that should eliminate the possible contradictions that are expected because of Godel's theorem of incompleteness.
- 3. The other hypotheses on the lists should be checked, too. We should observe what is the change in the situation when the hypothesis is changed, are the ELFs closures violated, what is the relative compatibility of other hypotheses to the BG solutions contemplated.

In Figure 8, an example of ambiguous situation is presented. The right alternatives are hypothesized, and the disambiguation is easily performed by the human viewers even not familiar with the original phenomenon (see http://www.ournet.md/~mythorm/LochNess.htm) . One can easily check that the activities for disambiguation performed in a natural way are similar to those presented in the above list (hypothesize the connectivity of all segments of the expected body of a living creature (H1), hypothesize the radius of the "underwater" part (H2), verify the H1 with available information of possible living creatures, verify H2 by comparing it woth the visible radius of the part above the surface of "water", etc.

8. Multiresolutional Metrics

The concept of *value judgement* introduced in [9] and expanded in [1, 2] is expected to be a useful component of the measuring performance of systems, in particular, intelligent systems.

Figure 8. Ceramics "Loch Ness Monster" on a poliched wooden surface

Although this concept seems to be almost trivial, coinciding with the concepts of *cost/reward* applied in one set of research results, and repeating the premises of *utility function* from another set of research results, it has more obscurities than can be allowed for applying this concept in practical cases. In this paper, the issues are listed that should be clarified, properly stated and resolved before using the concept of *value judgment* would be scientifically justified.

We have some light problem with the issues of VALUE and VALUE JUDGMENT. Indeed, value judgment system can evaluate what is good and bad, important and trivial, and can estimate cost, benefit, and risk of potential future actions. However, it is difficult to find objective evaluators. Indeed, scalar evaluators need a tool for assigning weights to various components of VJ. Vector evaluators intend to escape the the need for dealing with the idea of relative importance of the components of the vector. Actually, neither is achieved in practical cases. There

- are many factors of preferences that cannot be easily transformed into physical values or money.
- Preferability that is delivered by emotions is still a subject of discussion. It is unclear how to assign a numerical value to the degree of preferability brought by one's loyalty. Why does one care that the team of his/her school wins the game even if this game is beyond his/her interest and even simple curiosity?
- Even if the problem of computing the value judgment is resolved at a particular level of resolution, one cannot present any meaningful techniques of consolidating all measures into a single numerical value.
- The previous problem might be considered easier if at least we knew where to cut-off building representations of the next level of resolution from above and from below. These are silly but "fundamental" considerations: the limit of generalization from above is achieved when we stop blurring particular details since it affects the interpretation, the limit of instantiation below is considered to be achieved when we do not know how to make further decomposition of the representation.
- One of the areas containing multiresolutional analysis related results and intuitions is not sufficiently analyzed by scientists in multiresolutional representation and behavior generation: the on-standard analysis [10]. A. Robinson stops decimating space at the indistinguishability zone level (the limit of tessellation from below).

* It is possible to expect that Heisenberg's Uncertainty Principle is not bound by sub-atomic particles and quantum mechanics and can be applied for any level of resolution in the MR structures.

References

1. A. Meystel, J. Albus, *Intelligent Systems: Architectures, Design, Control*, Wiley, NY, 2001
2. J. Albus, A. Meystel, *Engineering of Mind: An Introduction to the Science of Intelligent Systems,* Wiley, NY, 2001
3. E. Dawidowicz, "Performance Evaluation of Network Centric Warfare Oriented Intelligent Systems " Proceedings of PERMIS'2001, Gaithersburg, MD, 2002
4. E. Messina, A. Meystel, "The Challenge of Intelligent Systems," Proceedings of the 2000 IEEE International Symposium on Intelligent Control, Eds. P. Groumpos, N. Koussoulas, M. Polycarpou, July 17-19, Patras, Greece, 2000, pp. 211-216
5. H. Pattee, "Physical basis and origin of control, in *Hierarchy Theory*, ed., Braziler, NY, 1987, [see p. 94]
6. *Powers of Ten*, by Morrisons and the Office of Charles and Ray Eaems, Scientific American Library, NY, 1982
7. D. C. Dennett, *Kinds of Minds: Toward an Understanding of Consciousness*, Basic Books, 1996
8. A. Meystel, "Evolution of Intelligent Systems Architectures: What Should Be Measured?" in Eds. A. Meystel, E. Messina, *Measuring the Performance and Intelligence of Systems*, Proceedings of the PERMIS'2000, August 14-16, 2000, Gaithersburg, MD 2000, pp. 361-382
9. J. Albus, "Outline for a theory of intelligence," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 3, 1991, pp. 473-509
10. A. Robinson, *Non-Standard Analysis*, Princeton U. Press, 1974

PART III

RESEARCH PAPERS

PART III
RESEARCH PAPERS

1. Mathematical Aspects of Performance Evaluation

Interval Mathematics for Analysis of Multiresolutional Systems

Vladik Kreinovich[1] and Richard Aló[2]

[1]Computer Science, University of Texas, El Paso, TX 79968, USA, vladik@cs.utep.edu
[2]CS and Math, U. of Houston-Downtown, Houston, TX 77002, USA, ralo@uh.edu

Abstract— **The more complex the problem, the more complex the system necessary for solving this problem. For very complex problems, it is no longer possible to design the corresponding system on a single resolution level, it becomes necessary to have multiresolutional systems. When analyzing such systems – e.g., when estimating their performance and/or their intelligence – it is reasonable to use the multiresolutional character of these systems: first, we analyze the system on the low-resolution level, and then we sharpen the results of the low-resolution analysis by considering higher-resolution representations of the analyzed system. The analysis of the low-resolution level provides us with an approximate value of the desired performance characteristic. In order to make a definite conclusion, we need to know the accuracy of this approximation. In this paper, we describe interval mathematics – a methodology for estimating such accuracy. The resulting interval approach is also extremely important for tessellating the space of search when searching for optimal control. We overview the corresponding theoretical results, and present several case studies.**

I. Multiresolutional Methods are Necessary: A Brief Reminder

The more complex the problem, the more complex the system necessary for solving this problem. For very complex problems, it is no longer possible to design the corresponding system on a single resolution level, it becomes necessary to have multiresolutional systems.

The methodology of multiresolutional search for the optimum solution of a control problem was first presented by A. Meystel in [40], [41]. These papers contributed to the broad interest in and dissemination of the multiresolutional approach to solving problems of the areas of intelligent control and intelligent systems.

Many algorithms based on this methodology were developed since then. The successful practical applications of these algorithms shows that multiresolutional approach are indeed necessary.

This empirical conclusion has been supported by many mathematical results; let us name a few recent ones:

• It has been proven that for general complex (NP-hard) problems, i.e., problems, for which no general feasible algorithm is possible, there always exists an appropriate "granulation" after which the problem becomes easy to solve. The fact that the problem is NP-hard means that there is no general algorithm for automatically finding such a granulation, this granulation requires an expert familiar with the particular problem that we are trying to solve [11].

• For noisy images $I(x)$ in which we do not know the exact statistical characteristics of the noise, only the upper bound on the noise, the optimal image processing requires representing this image as a linear combination of so-called *Haar wavelets* $e_i(x)$, i.e., functions which only take values 1 or 0. Such a wavelet representation is a known particular case of a multiresolutional representation [5], [6].

• In particular, when detecting a known pattern in a given image, it is provably better to use lower-resolution type techniques that look for the whole pattern as opposed to higher-resolution techniques which look for pieces of this pattern and then try to match found pieces together [64].

• Similarly to noisy images, for signal multiplexing under noise, the use of Walsh functions (similar to Haar wavelets) can be proven to be the optimal choice [2].

• In general, in function interpolation, clustering techniques – in which we combine the values into clusters before extrapolation – turn out to be optimal [34]. Such an interpolation is very useful in intelligent control, when we train a system by providing it with examples of control values used by expert human controllers in different situations.

• In general, in intelligent control, hierarchical fuzzy control is better in the sense that it requires fewer rules to describe the same quality control [35], [36], [77].

• Finally, it can be shown that for many systems, the optimal control is of "bang-bang" type, when there are finitely many preferred control values (or preferred fixed control trajectories), and the optimal control consists of optimally switching between these values (trajectories). This general result explains different empirical phenomena ranging from the empirical fact of discrete speed levels in traffic control to the phenomenon of sleep when it seems to be biologically optimal to always switch between several fixed levels of activity [29].

II. Interval Mathematics: A Methodology for Validated Analysis of Multiresolutional Systems

A. Validated Analysis of Multiresolutional Systems Naturally Leads to Interval Computations

When analyzing multiresolutional systems – e.g., when estimating their performance and/or their intelligence – it is reasonable to use the multiresolutional character of these systems: first, we analyze the system on the low-resolution level, and then we sharpen the results of the low-resolution analysis by considering higher-resolution representations of the analyzed system.

For example, instead of the original image with its numerous pixel-by-pixel brightness values, we consider a low-resolution image in which there is a small finite number of zones, and each zone is characterized by a single brightness value. After analyzing this image, we increase resolution, thus adding more details (more zones), etc.

The analysis of the low-resolution level provides us with an approximate value of the desired performance characteristic. In order to make a definite conclusion, we need to know the accuracy of this approximation. How can we estimate this accuracy?

In order to solve this problem, let us reformulate it in general mathematical terms. Instead of considering the *exact* system, we consider its *approximation*, analyze this approximation, and then we want to make a conclusion about the original system based on this analysis. The original system is characterized by the values of different parameters $x_1, \ldots, x_n$; e.g., for the image, these parameters are the brightness values at different pixels. We want to estimate some characteristic $q = f(x_1, \ldots, x_n)$ of the original system.

A low-resolution approximation can be usually described by fewer parameters $y_1, \ldots, y_m$, $m \ll n$; e.g., for the image, these parameters are the brightnesses of different zones. Each parameter x_i is approximated by one of the new parameters y_j; let us denote the corresponding parameter by $y_{j(i)}$. When each x_i is exactly equal to the corresponding value y_j, we get a simplified expression for q which only depends on $m \ll n$ values: $\widetilde{q} = \widetilde{f}(y_1, \ldots, y_n)$. In reality, the values x_i are somewhat different from y_j, and as a result, the estimate $\widetilde{q}$ is different from the actual value q of the desired characteristic. How can we estimate the corresponding approximation error $\widetilde{q} - q$?

In addition to the approximate model itself, we usually know, for each j, the upper bound on the error with which the value y_j approximates the corresponding values x_i. In other words, we know that the actual value of x_i belongs to the interval $\mathbf{y}_j = [y_j - \Delta_j, y_j + \Delta_j]$. Since each value x_i belongs to the interval $\mathbf{y}_{j(i)}$, the actual value of the desired characteristic belongs to the *range*

$$\mathbf{q} = f(\mathbf{y}_{i(1)}, \ldots, \mathbf{y}_{i(n)}) \stackrel{\text{def}}{=} \{f(x_1, \ldots, x_n) \,|\, x_i \in \mathbf{y}_{j(i)}\}$$

of the function f on these intervals. Thus, in order to estimate the accuracy of the lower-resolution estimate $\widetilde{q}$, we can estimate the above range.

The problem of estimating the range of the function $f(x_1, \ldots, x_n)$ when we know the intervals $\mathbf{x}_i$ of possible values of x_i is a known problem in areas where the inputs are not known precisely, be it numerical methods or data processing. This problem is called the problem of *interval computations*, and methods for solving this problem are called *interval mathematics* [1], [16], [17], [19], [20], [44], [75].

B. Interval Computations are Difficult

In general, the interval computation problem is NP-hard even for quadratic functions $f(x_1, \ldots, x_n)$; see, e.g., [26]. In plain English, this means that it is highly unprovable that we will be able to find a general feasible algorithm that computes the exact range for all functions f and all intervals $\mathbf{x}_i$ in reasonable time. Since we cannot compute the exact range, what can we do instead?

We wanted to compute the exact range $\mathbf{q}$ because we wanted to get an interval that is guaranteed to contain the desired value q, and the range definitely contains this value. If we cannot compute the *exact* range in reasonable time, we can compute the *approximate* interval $\mathbf{Q}$ for the range. The only way to guarantee that the new interval still contains q is to make sure that this new intervals contains the entire range $\mathbf{q} \subseteq \mathbf{Q}$, i.e., that this interval is an *enclosure* for the desired range.

In these terms, interval mathematics is an art of computing good narrow enclosures for the range of a given function $f(x_1, \ldots, x_n)$ on given intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

C. Methods of Interval Mathematics: A Very Brief Introduction

Interval mathematics started, in the 1950s, with the observation that for simple arithmetic operations $f(x_1, x_2) = x_1 + x_2$, $x_1 - x_2$, etc., the range can be computed explicitly; e.g.:

$$[x_1^-, x_1^+] + [x_2^-, x_2^+] = [x_1^- + x_2^-, x_1^+ + x_2^+];$$

$$[x_1^-, x_1^+] - [x_2^-, x_2^+] = [x_1^- - x_2^+, x_1^+ - x_2^-];$$

$$[x_1^-, x_1^+] \cdot [x_2^-, x_2^+] = [\min(x_1^- \cdot x_2^-, x_1^- \cdot x_2^+, x_1^+ \cdot x_2^-, x_1^+ \cdot x_2^+), \max(x_1^- \cdot x_2^-, x_1^- \cdot x_2^+, x_1^+ \cdot x_2^-, x_1^+ \cdot x_2^+)].$$

The corresponding expressions are called formulas of *interval arithmetic.*

It turns out that we can use these expressions to get reasonable enclosures for arbitrary functions f. Indeed, when the computer computes the function f, it *parses* the function, i.e., it represents the computation as a sequence of elementary arithmetic operations. It can proven, by induction, that if we start with intervals and replace each arithmetic operation with the corresponding operation of interval arithmetic, at the end, we get an enclosure for f. For example, if $f(x) = x \cdot (1 - x)$, represent f as a sequence of two elementary operations:
- $r := 1 - x$ (r denotes the 1st intermediate result);
- $y := x \cdot r$.

In the interval version, perform the following computations:
- $\mathbf{r} := 1 - \mathbf{x}$;
- $\mathbf{y} := \mathbf{x} \cdot \mathbf{r}$.

In particular, when $\mathbf{x} = [0, 1]$, compute the intervals $\mathbf{r} := [1, 1] - [0, 1] = [0, 1]$, and

$$\mathbf{y} := [0, 1] \cdot [0, 1] = [\min(0 \cdot 0, 0 \cdot 1, 1 \cdot 0, 1 \cdot 1), \max(0 \cdot 0, 0 \cdot 1, 1 \cdot 0, 1 \cdot 1)] = [0, 1].$$

The interval $[0, 1]$ is indeed an enclosure of the actual range $[0, 0.25]$.

D. *Modern Methods of Interval Mathematics and Their Potential Use in Tessellating the Search Space*

D.1 Methods Based on Mean Value Theorem

The enclosure obtained by using the above simple idea is often too wide. One of the main objectives of interval computations is to make this enclosure narrower. One way to do that is to use the *mean value* theorem, according to which $f(x) = f(x_0) + f'(\xi) \cdot (x - x_0)$ for some value ξ between x_0 and x. Thus, if we take, as x_0, the midpoint of the interval $\mathbf{x}$ of width w, we will have $|x - x_0| \le w/2$, $f'(\xi) \in f'(\mathbf{x})$, and thus, $f(\mathbf{x}) \subseteq f(x_0) + f'(\mathbf{x}) \cdot [-w/2, w/2]$. If we do not know the exact range $f'(\mathbf{x})$, we can use the enclosure for this range. Similar formulas can be easily written for the case of several variables.

D.2 Methods Based on Division into Subboxes and Their Relation with Multiresolutional Approach

In many cases, the above idea leads to a reasonable enclosure. If the enclosure is still too wide, we can divide the original box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$ into sub-boxes, compute the enclosure for each of these subboxes, and then take the union of the resulting enclosures.

It is worth mentioning that this idea is completely in line with the general multiresolutional approach: instead of considering the individual values of the function $f(x_1, \ldots, x_n)$ for all possible inputs $x_1, \ldots, x_n$, we divide the range of this function into a small number of zones, and consider the enclosure for each zone. In multiresolutional terms, we are thus considering a low-resolution approximation to the original function. If we want better results, we have to consider smaller zones, i.e., we have to consider higher-resolution approximations.

In other words, not only the *formulation* of the main problem of interval mathematics naturally comes from multiresolutional approach, but also the *methods* of interval mathematics are completely in line with this approach.

D.3 Interval Mathematics as a Method for Tessellating Search Space

The resulting interval approach is also extremely important for tessellating the space of search when searching for optimal control [19], [20]. The simplest way of using interval computations in to locate a maximum of the objective function $f(x)$ is as follows:

First, we compute the values of $f(x)$ in several points $x^{(1)}, \ldots, x^{(k)}$; we then now that $\max f(x) \ge M \stackrel{\text{def}}{=} \max(f(x^{(i)}))$. Then, we divide the original range into several zones Z_i, use interval computations to get an enclosure $\mathbf{F}_i = [F_i^-, F_i^+]$ of the range of $f(x)$ on each zone Z_i, and dismiss the zones for which $F_i^+ < M$ – because they cannot contain the global maxima.

Then, we subdivide the remaining zones into sub-zones, and repeat this procedure again – until we locate the global maxima. This idea leads to a reasonably efficient algorithms for global optimization, with can be further enhanced by using interval versions of gradient-based optimization methods.

Numerous similar methods exist for computing enclosures and optimization. Most of these methods are implemented in easily available software packages; see, e.g., [19], [20], [75].

D.4 Conclusion: Interval Mathematics Is Very Useful for Multiresolutional Approach

Based on the above, we can conclude that *interval mathematics is a good candidate for being "the" mathematics of multiresolutional systems.*

D.5 We Will Present Examples of Applying Interval Computations

In the following sections, we will describe two applications of interval mathematics in some detail. Before we go into the descriptions, we should mention that the above is the description of a "vanilla" situation. In many real-life cases, the situation is even more complex, because, in addition to a *quantitative* conclusion (about the value of some quantity q), we need to make a *qualitative* conclusion: e.g., in the following example, a conclusion on whether a plate has a hidden fault or not.

E. *Case Study: Non-Destructive Testing*

This case study is described, in detail, in [65], [72], [73], [74].

In many areas, e.g., in aerospace industry, in medicine, it is desirable to detect mechanical faults without damaging or reassembling the original system. For testing, we send a signal and measure the resulting signal. The input signal can be described by its intensity $r_1, \ldots, r_n$ at different moments of time. The intensities $s_1, \ldots, s_m$ of the resulting signal depend on r_i: $s_j = f_j(r_1, \ldots, r_n)$, where the functions f_j depend on the tested structure.

Usually, we do not know the exact analytical expression for the dependency f_j, so we can use the fact that an arbitrary continuous function can be approximated by a polynomial (of a sufficiently large order). Thus, we can take a structure, try a general linear dependency first, then, if necessary, general quadratic, etc., until we find the dependency that fits the desired data.

If a structure has no faults, then the surface is usually smooth. As a result, the dependency f_j is also smooth; we can expand it in Taylor series. Since we are sending relatively weak signals r_i (strong signals can damage the plane), we can neglect quadratic terms and only consider linear terms in these series; thus, the dependency will be *linear.*

A fault is, usually, a violation of smoothness (e.g., a crack). Thus, if there is a fault, the structure stops being smooth; hence, the function f_j stops being smooth, and therefore, linear terms are no longer sufficient. Thus, *in the absence of fault, the dependence is linear, but with the faults, the dependence is non-linear.* So, we can detect the fault by checking whether the dependency between s_j and r_i is linear. So, we send several different inputs, measure the values $r_i^{(k)}$ and $s_j^{(k)}$ corresponding to these inputs, and check whether the dependence is linear. In this case, the values $r_i^{(k)}$ and $s_j^{(k)}$ are the inputs $x_1, \ldots, x_n$, but the desired q is a qualitative (yes-no) variable: we simply want to know whether there is a fault or not. If there is a fault, then we would also like to make a quantitative conclusion of its size, location, etc., but the most important part of the analysis is to check whether there is any fault at all.

If the measurements were ideal, all we had to do was to check whether there are values a_{ji} for which, for all j and for all measurements k, we have:

$$a_{j0} + a_{j1} \cdot r_1^{(k)} + \ldots + a_{jn} \cdot r_n^{(k)} = s_j^{(k)}.$$

Solvability of a system of linear equations is easy to check.

In reality, the situation is more complicated. Measurement are usually imprecise: the result $\tilde{x}$ of measuring the actual value x is somewhat different from the actual value x. In many real-life situations, we do not know the probabilities of different values of measurement error $\Delta x = \tilde{x} - x$, we only know the upper bound Δ of the corresponding measurement error. As a result, the only information that we have about the actual value x of the measured quantity is that it belongs to the interval $\mathbf{x} = [x - \Delta, x + \Delta]$. So, in practice, instead of the exact values of $r_i^{(k)}$ and $s_j^{(k)}$, we have *intervals* $\mathbf{r}_i^{(k)}$ and $\mathbf{s}_j^{(k)}$ of possible values of these quantities. The question becomes: are these intervals consistent with the linearity, i.e., are there values $r_i^{(k)} \in \mathbf{r}_i^{(k)}$ and $s_j^{(k)} \in \mathbf{s}_j^{(k)}$ for which, for some values a_{ji}, the above linearity formulas hold.

In general, the solvability of the corresponding system of interval linear equations is an NP-hard problem [26], but for some cases, efficient algorithms have been developed. For example, when we have only one (non-negative) input and only one output, with non-intersecting intervals $\mathbf{r}^{(1)} < \mathbf{r}^{(2)} < \ldots$, the solvability of the corresponding system of linear equations can be proven to be equivalent to the following inequality:

$$\max_{k<l} \frac{s^{(l)-} - s^{(k)+}}{r^{(l)+} - r^{(k)-}} \le \min_{k<l} \frac{s^{(l)+} - s^{(k)-}}{r^{(l)-} - r^{(k)+}}.$$

We tested this method on the dependence of the energy E of the ultrasound response on the voltage V that causes the original ultrasound signal. The results show that non-linearity is indeed an indication of a fault:

- For faultless plates, the above inequality is indeed true, meaning that the measurement results are consistent with linearity.
- For plates with faults, this inequality is not satisfied, meaning that the dependence is non-linear.

F. Case Study: Reliable Sub-Division of Geological Areas

This case study is described, in detail, in [7], [8].

In geophysics, appropriate subdivision of an area into segments is extremely important, because it enables us to extrapolate the results obtained in some locations within the segment (where extensive research was done) to other locations within the same segment, and thus, get a good understanding of the locations which weren't that thoroughly analyzed. The subdivision of a geological zone into segments is often a controversial issue, with different evidence and different experts' intuition supporting different subdivisions.

For example, in our area – Rio Grande rift zone – there is some geochemical evidence that this zone is divided into three segments [39]:

- the southern segment which is located, approximately, between the latitudes $y = 29°$ and $y = 34°$;
- the central segment – from $y = 34.5°$ to $y = 38°$; and
- the northern segment – from $y = 38°$ to $y = 41°$.

However, in the viewpoint of many researchers, this evidence is not yet sufficiently convincing.

It is therefore desirable to develop new techniques for zone sub-division, techniques which would be in the least possible way dependent on the (subjective) expert opinion and would, thus, be maximally reliable. To make this conclusion more reliable, we use, instead of the more rare *geological* samples, a more abundant *topographical* information (this information, e.g., comes from satellite photos). We can characterize each part of the divided zone by its topography.

In topographical analysis, we face a new problem: of too much data, most of which is geophysically irrelevant. To eliminate some of this irrelevant data, we can use the Fourier transform; indeed, it is known that while (at least some) absolute values of the map (forming a so-called spectrum) are geophysically meaningful, the phases usually are random and can be therefore ignored. So, we should only use the spectrum.

Since we are interested only in the large-scale classification, it makes sense to only use the spectrum values corresponding to relatively large spatial wavelengths, i.e., wavelengths L for which $L \ge L_0$ for some appropriate value L_0. In particular, for the sub-division of the Rio Grande rift, it makes sense to use only wavelengths of $L_0 = 1000$ km or larger.

Also, for the Rio Grande Rift, we are interested in the classification of horizontal zones, so it makes sense to divide the Rio Grande Rift into 1° zones $[y^-, y^+]$ (with y from $y^- = 30$ to $y^+ = 31$, from $y^- = 31$ to $y^+ = 32$, …, from $y^- = 40$ to $y^+ = 41$). For each of these zones, we take the topographic data, i.e., the height $h(x, y)$ described as a function of longitude x and latitude y, compute the Fourier transform $H(\omega, y)$ with respect to x, combine all the spectral values which correspond to large wavelength (i.e., for which $\omega \le 1/L_0$), and compute the resulting spectral value

$$S(y^-) = \int_{y=y^-}^{y^+} \int_{\omega=0}^{1/L_0} |H(\omega, y)|^2 \, d\omega \, dy.$$

Since we are interested in comparing the spectral values $S(y)$ corresponding to different latitudes y, so we are not interested in the *absolute* values of $S(y)$, only in *relative* values. Thus, to simplify the data, we can *normalize* them by, e.g., dividing each value $S(y^-)$ by the largest $S_{\max}$ of these values. In particular, for the Rio Grande rift, the resulting values of $y^- = y_1, y_2, \ldots$ and $s_i = S(y_i)/S_{\max}$ are as follows:

TABLE I

y_i	29	30	31	32	33	34
s_i	0.28	0.24	0.21	0.16	0.20	0.29
35	36	37	38	39	40	41
0.31	0.35	0.46	1.00	0.80	0.96	0.74

Based only on these spectral values s_i, we will try to classify locations into several clusters ("segments").

From the geophysical viewpoint, the desired zones correspond to "monotonicity regions": in the first zone, the values s_i are (approximately) decreasing, in the next zone, they are (approximately) increasing, etc. So, we must look for the monotonicity regions of the (unknown) function $s(y)$.

The problem is that the values s_i are only approximately known, so we cannot simply compare the values to determine whether a function increases or decreases. The heights are measured pretty accurately, so the only errors in the values s_i come from discretization. In other words, we would like to know the values of the function $s(y) = S(y)/S_{\max}$ for all y, but we only know the values $s_1 = s(y_1)$, ..., $s_n = s(y_n)$ of this function for the points $y_1, \ldots, y_n$. For each y which is different from y_i, it is reasonable to estimate $s(y)$ as the value $s_i = s(y_i)$ at the point y_i which is the closest to y (and, ideally, which belongs to the same segment as y_i). For each point y_i, what is the largest possible error Δ_i of the corresponding approximation?

When $y > y_i$, the point y_i is still the closest until we reach the midpoint $y_{\text{mid}} = (y_i + y_{i+1})/2$ between y_i and y_{i+1}. It is reasonable to assume that the largest possible approximation error $|s(y) - s_i|$ for such points is attained when the distance between y and y_i is the largest, i.e., when y is this midpoint; in this case, the approximation error is equal to $|s(y_{\text{mid}}) - s_i|$.

If the points y_i and y_{i+1} belong to *the same* segment, then the dependence of $s(y)$ on y should be reasonably smooth for $y \in [y_i, y_{i+1}]$. Therefore, on a narrow interval $[y_i, y_{i+1}]$, we can, with reasonable accuracy, ignore quadratic and higher terms in the expansion of $s(y_i + \Delta y)$ and thus, approximate $s(y)$ by a linear function. For a linear function $s(y)$, the difference $s(y_{\text{mid}}) - s(y_i)$ is equal to the half of the difference $s(y_{i+1}) - s(y_i) = s_{i+1} - s_i$; thus, for $y > y_i$, the approximation error is bounded by $0.5 \cdot |s_{i+1} - s_i|$.

If the points y_i and y_{i+1} belong to *different* segments, then the dependence $s(y)$ should exhibit some non-smoothness, and it is reasonable to expect that the difference $|s_{i+1} - s_i|$ is much higher than the approximation error.

In both cases, the approximation error is bounded by

$$0.5 \cdot |s_{i+1} - s_i|.$$

Similarly, for $y < y_i$, the approximation error is bounded by $0.5 \cdot |s_i - s_{i-1}|$ if the points y_i and y_{i-1} belong to the same segment, and is much smaller if they don't. In both cases, the approximation error is bounded by

$$0.5 \cdot |s_i - s_{i-1}|.$$

We have two bounds on the approximation error and we can therefore conclude that the approximation error cannot exceed the smallest Δ_i of these two bounds, i.e., the value

$$\Delta_i = 0.5 \cdot \min(|s_i - s_{i_1}|, |s_{i+1} - s_i|).$$

As a result, instead of the *exact* values s_i, for each i, we get the *interval* $\mathbf{s}_i = [s_i^-, s_i^+]$ of possible values of $s(y)$, where $s_i^- = s_i - \Delta_i$ and $s_i^+ = s_i + \Delta_i$. In particular, for the Rio Grande rift, we get:

$$\mathbf{s}_1 = [0.26, 0.30], \mathbf{s}_2 = [0.225, 0.255], \mathbf{s}_3 = [0.195, 0.225],$$
$$\mathbf{s}_4 = [0.14, 0.18], \mathbf{s}_5 = [0.18, 0.22], \mathbf{s}_6 = [0.28, 0.30],$$
$$\mathbf{s}_7 = [0.30, 0.32], \mathbf{s}_8 = [0.33, 0.37], \mathbf{s}_9 = [0.405, 0.515],$$
$$\mathbf{s}_{10} = [0.80, 1.10], \mathbf{s}_{11} = [0.72, 0.88], \mathbf{s}_{12} = [0.88, 1.04],$$
$$\mathbf{s}_{13} = [0.63, 0.85].$$

We want to find regions of uncertainty of a function $s(y)$, but we do not know the exact form of this function; all we know is that for every i, $s(y_i) \in \mathbf{s}_i$ for known intervals $\mathbf{s}_i$. How can we find the monotonicity regions in the situation with such interval uncertainty? Of course, since we only know the values of the function $s(y)$ in finitely many points y_i, this function can have as many monotonicity regions between y_i and y_{i+1} as possible. What we are interested in is funding the subdivision into monotonicity regions which can be deduced from the data. The first natural question is: can we explain the data by assuming that the dependence $s(y)$ is monotonic? If not, then we can ask for the possibility of having a function $s(y)$ with exactly two monotonicity regions:

- if such a function is possible, then we are interested in possible locations of such regions;
- if such a function is not possible, then we will try to find a function $s(y)$ which is consisted with our interval data and which has three monotonicity regions, etc.

This problem was first formalized and solved in [68], [69], where we developed a linear-time algorithm for solving this problem. By applying this algorithm, we find three monotonicity regions: $[29, 34]$, $[31, 41]$, and $[37, 41]$ – in good accordance with the geochemical data from [39].

G. Other Applications: A Brief Overview

Other successful applications of interval techniques include:

- telemanipulation [9], [25], [65];
- robot navigation [65];
- analysis of multi-spectral satellite images [63], [65].

Since a fuzzy set can be naturally represented as a nested family of intervals (corresponding to different levels of certainty), methods of fuzzy data processing actively use interval computations and be considered as natural applications of interval techniques [22], [50], [54], [65].

III. Multi-D Generalizations of Interval Mathematics and Symmetry Approach

A. General Idea

In addition to the upper bound on the approximation error for each quantity x_i, we often have an additional information. For example, in some cases, in addition to the upper bounds Δ_i for the differences $\widetilde{x}_i - x_i$, we also know the upper bound on their distance between the vectors $\widetilde{x}$ and x, i.e., the upper bound on $\sqrt{(\widetilde{x}_1 - x_1)^2 + \ldots + (\widetilde{x}_n - x_n)^2}$. In this case, we know that the actual values of $x_1, \ldots, x_n$ belongs to the intersection of a box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$ and a ball. We may have more complex shapes. Processing complex shapes is computationally difficult (see, e.g., [32]), so we must find good approximations for such shapes. Ideally, we should find approximations which are *optimal* in some reasonable sense.

A similar problem of finding the optimal shapes arises in the selection of "clusters" (zones) corresponding to the low-resolution approximation. Here also, it is desirable to find the optimal zones.

Let us show, on the example of selecting zones on the plane, how this problem can be solved (a more general case is described in [47]).

Of course, the more parameters we allow, the better the approximation. So, the question can be reformulated as follows: for a given number of parameters (i.e., for a given dimension of approximating family), which is the best family?

For simplicity, we will restrict ourselves to families of sets have analytical (or piece-wise analytical) boundaries, i.e., boundaries that can be described by an equation $F(x,y) = 0$ for some analytical function $F(x,y) = a + bx + cy + dx^2 + exy + fy^2 + \ldots$ Since we are interested in finite-dimensional families of sets, it is natural to consider finite-dimensional families of functions, i.e., families of the type $\{C_1 \cdot F_1(x,y) + \ldots + C_d \cdot F_d(x,y)\}$, where $F_i(z)$ are given analytical functions, and $C_1, \ldots, C_d$ are arbitrary (real) constants. So, the question is: which of such families is the best?

When we say "the best", we mean that on the set of all such families, there must be a relation $\geq$ describing which family is better or equal in quality. This relation must be transitive (if A is better than B, and B is better than C, then A is better than C). This relation is not necessarily asymmetric, because we can have two approximating families of the same quality. However, we would like to require that this relation be *final* in the sense that it should define a unique *best* family A_{opt} (i.e., the unique family for which $\forall B\,(A_{\text{opt}} \geq B)$. Indeed:

- If none of the families is the best, then this criterion is of no use, so there should be *at least one* optimal family.
- If *several* different families are equally best, then we can use this ambiguity to optimize something else: e.g., if we have two families with the same approximating quality, then we choose the one which is easier to compute. As a result, the original criterion was not final: we get a new criterion ($A \geq_{\text{new}} B$ if either A gives a better approximation, or if $A \sim_{\text{old}} B$ and A is easier to compute), for which the class of optimal families is narrower. We can repeat this procedure until we get a final criterion for which there is only one optimal family.

It is reasonable to require that the relation $A \geq B$ should be invariance relative to natural geometric symmetries, i.e., shift-, rotation- and scale-invariant.

Now, we are ready for the formal definitions.

Definition 1. *Let $d > 0$ be an integer. By a d-dimensional family, we mean a family A of all functions of the type $\{C_1 \cdot F_1(x,y) + \ldots + C_d \cdot F_d(x,y)\}$, where $F_i(z)$ are given analytical functions, and $C_1, \ldots, C_d$ are arbitrary (real) constants. We say that a set is defined by this family A if its border consists of pieces described by equations $F(x,y) = 0$, with $F \in A$.*

Definition 2. *By an optimality criterion, we mean a transitive relation $\geq$ on the set of all d-dimensional families. We say that a criterion is final if there exists one and only one optimal family, i.e., a family A_{opt} for which $\forall B\,(A_{\text{opt}} \geq B)$. We say that a criterion $\geq$ is shift- (corr., rotation- and scale-invariant) if for every two families A and B, $A \geq B$ implies $TA \geq TB$, where TA is a shift (rotation, scaling) of the family A.*

Theorem [33], [71]. ($d \leq 4$) *Let $\geq$ be a final optimality criterion which is shift-, rotation-, and scale-invariant, and let A_{opt} be the corresponding optimal family. Then, the border of every set defined by this family A_{opt} consists of straight line intervals and circular arcs.*

For $d = 5$ and $d = 6$, we also get hyperbolas, parabolas, and ellipses [55].

A similar symmetry-based optimization technique can be used to find the optimal technique for subdividing boxes in interval range estimation and interval optimization; see, e.g., [21].

B. Case Studies: Brief Overview

B.1 Analyzing Cotton Images

The above approach has been very helpful in the automatic analysis of cotton images [55], [61]. Specifically, the above symmetry-based approach helps in classifying trash (bark, leaves, etc.) in ginned cotton and in classifying insects by their shapes. The symmetry approach enables us not only to find the optimal shapes, but also to find the optimal geometric characteristics for distinguishing between different shapes and different sizes of the same size. The same symmetry approach leads to the conclusion that the optimal approximations to sizes form a geometric progression; this conclusion is in good accordance with the actual insect sizes.

B.2 Half-Orders of Magnitude

A similar geometric progression result explains why, when people make crude estimates, they feel comfortable choosing between alternatives which differ by a half-order of magnitude (e.g., were there 100, 300, or 1,000 people in the crowd), and less comfortable making a choice on a more detailed scale, with finer granules, or on a coarser scale (like 100 or 1,000) [18]. This empirical fact is difficult to explain within standard uncertainty formalisms like fuzzy logic; see, e.g., [31].

B.3 Analyzing Geospatial Data II

Computer processing can drastically improve the quality of an image and the reliability and accuracy of a spatial database. A large image (database) does not easily fit into the computer memory, so we process it by downloading pieces of the image. Each downloading takes a lot of time, so, to speed up the entire processing, we must use as few pieces as possible.

Many algorithms for processing images and spatial databases consist of comparing the value at a certain spatial location with values at nearby locations. For such algorithms, we must select (possibly overlapping) sub-images in such a way that for each point, its neighborhood (of given radius) belongs to a single sub-image. In [3], we formulate the corresponding optimization problem in precise terms, and show (in good accordance with the above optimization result) that the optimal sub-images should be bounded by straight lines or circular arcs.

B.4 Analyzing Geospatial Data III

Geospatial databases often contain erroneous measurements. For some such databases such as gravity databases, the known methods of detecting erroneous measurements – based on regression analysis – do not work well. As a result, to clean such databases, experts use manual methods which are very time-consuming. In [70], we propose a (natural) multiresolutional (localized) version of regression analysis as a technique for automatic cleaning. Specifically, we subdivide the original image into zones, and apply regression analysis separately within each zone (on the high-resolution level) and between different zones (on a low-resolution level).

In this physical problem, natural requirements lead to the following optimality criterion for selecting zones: minimizing the zone's diameter (that describes the variance within the zone) under given area (that describes the number of measurements within the zones). The efficiency of the resulting optimal zones is shown on the example of the gravity database, where our algorithm not only detected all erroneous measurements found manually by the experts;, but it also uncovered several suspicious points that the experts overlooked.

B.5 Non-Destructive Testing II

A standard way of detecting faults is to measure a certain quantity x at different points on the analyzed plate, and to classify the point as faulty is when the value x of the measured quantity at this point differs from the average a of measurement results by more than two or three σ.

Based on the results of measuring a single quantity (e.g., ultrasonic signal), we often miss some faults. To improve the quality of fault detection, it is necessary to measure several different quantities, and combine the results of these measurements. A natural idea is to classify the point as faulty is one of the measurement detects a fault. However, one of the measurements may be erroneous, we would rather consider a point a fault location if at least one other measured quantity at this or nearby point indicates a fault.

In other words, to improve the quality of fault detection, we replace the original point-by-point analysis by a new method which involves high-resolution clustering. When the corresponding neighborhoods are selected in an optimal way, this replacement indeed improves the quality of fault detection [58], [59].

A further improvement in fault detections comes when we treat the physically different points near the plate's edge as a different zone, and classify a point as faulty only if the corresponding value x differs from the average a_z within this zone by more than two or three standard deviations σ_z measured within this zone z. In other words, a further improvement in fault detection comes when we supplement the above high-resolution technique by additional low-resolution subdivision into zones.

B.6 Why Two Sigma

In the above example, and in statistics in general, a two-sigma criterion is used. The normal justification for this criterion is that for $k \approx 2$, the dependence of the probability to be outside the $k \cdot \sigma$ interval $[a - k \cdot \sigma, a + k \cdot \sigma]$ on the (unknown) probability distribution is the smallest. In [52], [53], we provide a theoretical explanation for this empirical fact, and thus, for the "2σ" criterion.

For that, we take into consideration the fact that an arbitrary probability distribution can be represented as $f(\eta)$, where η is normally distributed, so the choice of a distribution is equivalent to the choice of a function $f(x)$. An symmetry-based approach similar to the one presented above leads to the family $f(x) = x^\alpha$, and for this family, in the vicinity of normal distribution (when $\alpha \approx 1$), the smallest dependence on α is indeed attained for $k \approx 2$.

B.7 Acupuncture Points

The above approach to describing optimal shapes can be successfully applied to finding a good approximation for the location of the acupuncture points, i.e., points in which acupuncture treatment is the most efficient [46].

B.8 Towards Optimal Image Compression

In the above image processing problems, we process the image as it appears. In many situations, we must store the image for future use, and there is not enough storage space to store all the images, so we need to compress the image. In other situations, there is not enough bandwidth to send the entire image, so again, compression is needed.

It is proven that finding the optimal compression of a given image, be it an optimal lossless compression or an optimal lossy compression with a given bound on allowable loss of information, is a computationally difficult problem [66]. Since we cannot find the optimal compression, a natural idea is to consider several compression techniques and find the best one. The problem is to quantify what "the best" means, especially in the situations when we may have several possible applications of the compressed image, and since we do not know where exactly this image will be used, it is difficult to quantify the quality of the compression. In [23], [49], we consider the optimal choice of quality metric most appropriate for a given problem. First, we use a similar-based optimization approach to find the optimal family of possible quality metrics (which turns out to be L^p-metrics), and then, we find p based on a specific problem.

B.9 Pattern Matching

In many real-life situations, we are interested in finding the known pattern in a given image. For example, in the analysis of geospatial data, we may be looking for certain geophysical patterns indicative of, say, presence of water. In [10], [12], [13], [14], [62], [78], a similar symmetry-based optimality approach is used to develop optimal FFT-based techniques for such matching.

B.10 Guaranteed Quality Estimation for Approximately Given Systems

Our final example bring us back to the original problem – of quality estimation for an approximately given system. Symmetry-based approach can help in designing optimal methods for such quality estimation for the situations when the system is treated as a "black box", a low-resolution approximation to the original system in which we are not allowed to use the high-resolution details [24], [67]. In particular, in [24], [67], we describe modified Monte-Carlo techniques which provide us with validated results even when we do not know the exact values of the statistical characteristics of the system – only intervals of possible values of such characteristics.

IV. Multiresolutional Approach to Reasoning and Logic: A Brief Overview

A. Reasoning and Logic: Successes and Problems

Multiresolutional approach can be applied not only to the systems themselves, but also to the way we reason about these systems, i.e., to the *logic* of human reasoning. Specifically, in many areas (medicine, geophysics, military decision-making, etc.), top quality experts make good decisions, but they cannot handle all situations. It is therefore desirable to incorporate their knowledge into a decision-making computer system.

Experts describe their knowledge by statements $S_1, \ldots, S_n$ (e.g., by if-then rules). Experts are often not 100% sure about these statements S_i; this uncertainty is described by the *subjective probabilities* p_i (degrees of belief, etc.) which experts assign to their statements. The conclusion C of an expert system normally depends on several statements S_i. For example, if we can deduce C either from S_2 and S_3, or from S_4, then the validity of C is equivalent to the validity of a Boolean combination $(S_2 \,\&\, S_3) \vee S_4$. So, to estimate the reliability $p(C)$ of the conclusion, we must estimate the probability of Boolean combinations. In this paper, we consider the simplest possible Boolean combinations are $S_1 \,\&\, S_2$ and $S_1 \vee S_2$.

In general, the probability $p(S_1 \,\&\, S_2)$ of a Boolean combination can take different values depending on whether S_1 and S_2 are independent or correlated. So, to get the precise estimates of probabilities of all possible conclusions, we must know not only the probabilities $p(S_i)$ of individual statements, but also the probabilities of all possible Boolean combinations. To get all such probabilities, it is sufficient to describe 2^n probabilities of the combinations $S_1^{\varepsilon_1} \,\&\, \ldots \,\&\, S_n^{\varepsilon_n}$, where $\varepsilon_i \in \{+,-\}$, S^+ means S, and S^- means $\neg S$. The only condition on these probabilities is that their sum should add up to 1, so we need to describe $2^n - 1$ different values. A typical knowledge base may contain hundreds of statements; in this case, the value $2^n - 1$ is astronomically large. We cannot ask experts about all 2^n such combinations, so in many cases, we must estimate $p(S_1 \,\&\, S_2)$ or $p(S_1 \vee S_2)$ based only on the values $p_1 = p(S_1)$ and $p_2 = p(S_2)$. There exist many possible "and"-operations $f_\& : [0,1] \times [0,1] \to [0,1]$ which transform the degrees p_1 and p_2 into an estimate $f_\&(p_1, p_2)$ for $p(S_1 \,\&\, S_2)$. Similarly, there exist many "or"-operations which transform degrees the p_1 and p_2 into an estimate $f_\vee(p_1, p_2)$ for $p(S_1 \vee S_2)$.

Many such operations have been successfully used in fuzzy logic and intelligent control; see, e.g., [22], [56]. In spite of the successes, there are still major problems with these operations:

- First, these operations are not perfect. Indeed, some of these operations, although very natural and useful at first glance, seem to violate natural commonsense requirements; we will give an example later).
- Second, there are so many different possible "and"- and "or"-operations that it is difficult to meaningfully select one of them. Any guidance for decreasing the class of possible operations is very welcome.

B. Reasoning and Logic: Multiresolutional Approach

In our viewpoint, the above problems of the existing logical methodologies come, to a large extent, from the fact that researchers often combine different degrees of certainty together. In reality, the degrees have a clear multiresolutional character, and if we fully take this character into consideration, we can make a large progress in solving the above problems.

Let us explain why expert degrees of uncertainty are multiresolutional. An expert rarely provides us with numbers describing his or her degrees of uncertainty. A more natural way for an expert to describe his/her degree of belief in a certain statement is to use a word from natural language such as "most probably" or "possibly", and then we translate this word into a number. There are only few such words, and these words form the lowest-resolution level of the uncertainty description. On this level, several different statements with slightly different degrees of uncertainty may be described by the same word and thus, lumped into a single cluster. To avoid this lumping, we may ask an expert to provide us with a more detailed description of the expert's degree, e.g., by using hedged combinations of words like "slightly less certain but still reasonably certain". The more details we ask, the more higher-resolution description we get.

Another possibility to describe the expert's degrees in numerical terms is to ask the expert to describe his/her degrees on a scale from, say, 0 to 10. We can start with a low-resolution scale, e.g., with a scale consisting of only two values "yes" and "no" that corresponds to the use of the classical (two-valued) logic. As we increase the number of elements on the scale, we get a higher- and higher-resolution description. Eventually, we get real numbers describing uncertainty.

In both cases, we get numbers as a result, but these numbers appear as a result of a multiresolutional procedure. It is therefore natural, when resolving the above problems – of seeming inconsistency with common sense and of too many options – to consider not only the resulting assignments of numbers, but also the multiresolutional approximations to these assignments. This consideration indeed helps in solving the above problems.

C. *Multiresolutional Character of Uncertainty Reasoning Resolves the Inconsistency Between Uncertainty Operations and Common Sense*

Let us give one example of such inconsistency and show how the multiresolutional character of human reasoning can help with this particular example. It is known that for given $p_1 = p(S_1)$ and $p_2 = p(S_2)$, possible values of $p(S_1 \,\&\, S_2)$ form an interval $\mathbf{p} = [p^-, p^+]$, where $p^- = \max(p_1 + p_2 - 1, 0)$ and $p^+ = \min(p_1, p_2)$; and possible values of $p(S_1 \vee S_2)$ form an interval $\mathbf{p} = [p^-, p^+]$, where $p^- = \max(p_1, p_2)$ and $p^+ = \min(p_1 + p_2, 1)$ (see, e.g., a survey [48] and references therein). So, in principle, we can use such interval estimates and get an interval $\mathbf{p}(C)$ of possible values of $p(C)$. Sometimes, this idea leads to meaningful estimates, but often, it leads to a useless $\mathbf{p}(C) = [0,1]$ [47], [57]. In such situations, it is reasonable, instead of using the entire interval $\mathbf{p}$, to select a point within this interval as a reasonable estimate for $p(S_1 \,\&\, S_2)$ (or, correspondingly, for $p(S_1 \vee S_2)$).

Since the only information we have, say, about the unknown probability $p(S_1 \,\&\, S_2)$ is that it belongs to the interval $[p^-, p^+]$, it is natural to select a *midpoint* of this interval as the desired estimate:

$$f_\&(p_1, p_2) \stackrel{\text{def}}{=} \frac{1}{2} \cdot \max(p_1 + p_2 - 1, 0) + \frac{1}{2} \cdot \min(p_1, p_2);$$

$$f_\vee(p_1, p_2) \stackrel{\text{def}}{=} \frac{1}{2} \cdot \max(p_1, p_2) + \frac{1}{2} \cdot \min(p_1 + p_2, 1).$$

This midpoint selection is not only natural from a common sense viewpoint; it also has a deeper justification. Namely, in accordance of our above discussion, for $n = 2$ statements S_1 and S_2, to describe the probabilities of all possible Boolean combinations, we need to describe $2^2 = 4$ probabilities $x_1 = p(S_1 \,\&\, S_2)$, $x_2 = p(S_1 \,\&\, \neg S_2)$, $x_3 = p(\neg S_1 \,\&\, S_2)$, and $x_4 = p(\neg S_1 \,\&\, \neg S_2)$; these probabilities should add up to 1: $x_1 + x_2 + x_3 + x_4 = 1$. Thus, each probability distribution can be represented as a point $(x_1, \ldots, x_4)$ in a 3-D simplex $\mathcal{S} = \{(x_1, x_2, x_3, x_4) \,|\, x_i \geq 0 \,\&\, x_1 + \ldots + x_4 = 1\}$. We know the values of $p_1 = p(S_1) = x_1 + x_2$ and $p_2 = p(S_2) = x_1 + x_3$, and we are interested in the values of $p(S_1 \,\&\, S_2) = x_1$ and $p(S_1 \vee S_2) = x_1 + x_2 + x_3$. It is natural to assume that *a priori*, all probability distributions (i.e., all points in a simplex $\mathcal{S}$) are "equally possible", i.e., that there is a uniform distribution ("second-order probability") on this set of probability distributions. Then, as a natural estimate for the probability $p(S_1 \,\&\, S_2)$ of $S_1 \,\&\, S_2$, we can take the conditional mathematical expectation of this probability under the condition that the values $p(S_1) = p_1$ and $p(S_2) = p_2$:

$$E(p(S_1 \,\&\, S_2) \,|\, p(S_1) = p_1 \,\&\, p(S_2) = p_2) =$$

$$P(x_1 \,|\, x_1 + x_2 = p_1 \,\&\, x_1 + x_3 = p_2).$$

The problem is that these operations are non-associative. Why is this a problem? If we are interested in estimating the degree of belief in a conjunction of three statements $S_1 \,\&\, S_2 \,\&\, S_3$, then we can either apply the "and" operation to p_1 and p_2 and get an estimate $f_\&(p_1, p_2)$ for the probability of $S_1 \,\&\, S_2$ and then, we apply the "and" operation to this estimate and p_3, and get an estimate $f_\&(f_\&(p_1, p_2), p_3)$ for the probability of $(S_1 \,\&\, S_2) \,\&\, S_3$. Alternatively, we can get start by combining S_2 and S_3, and get an estimate $f_\&(p_1, f_\&(p_2, p_3))$. Intuitively, we would expect these two estimates to coincide, but, e.g., $(0.4 \,\&\, 0.6) \,\&\, 0.8 = 0.2 \,\&\, 0.8 = 0.1$, while $0.4 \,\&\, (0.6 \,\&\, 0.8) = 0.4 \,\&\, 0.5 = 0.2 \neq 0.1$.

How can we solve this problem? Since we know that the numerical values are only an approximation, we can analyze how non-associative the above operations can be. If the difference is below the natural resolution level, then, from the practical point of view, the above operations are as good as associative ones. The following is true:

Theorem [15], [38].

$$\max_{a,b,c} |f_\&(f_\&(a,b),c) - f_\&(a, f_\&(b,c))| = \frac{1}{9};$$

$$\max_{a,b,c} |f_\vee(f_\vee(a,b),c) - f_\vee(a, f_\vee(b,c))| = \frac{1}{9}.$$

Each word describing a degree of belief is a "granule" covering the entire sub-interval of values. Thus, non-associativity is negligible if the corresponding realistic "granular" degree of belief have granules of width $\geq 1/9$. One can fit no more than 9 granules of such width in the interval $[0,1]$. This may explain why humans are most comfortable with ≤ 9 items to choose from – the famous "7 plus minus 2" law; see, e.g., [42], [43].

D. *Multiresolutional Character of Uncertainty Reasoning Helps to Drastically Narrow Down the Class of Possible Logics*

These results cover both the logics in which the set of different degrees is an interval $[0,1]$, and more complex logics.

D.1 $[0,1]$-Based Logics

For *numerical* operations, if we interpret the degree of belief in a statement S as (proportional to) the number of arguments in favor of S, then we arrive at a natural choice of "and"- and "or" operations: $f_{\&}(a,b) = a \cdot b$, $f_{\vee}(a,b) = a+b$, and $f_{\vee}(a,b) = b^a$. As one of the unexpected consequences, we get a surprising relation with the entropy techniques, well known in probabilistic approach to uncertainty [60].

A similar conclusion can be made if we require that the operations be consistent with their multiresolutional structure: namely, for a discrete low-resolution level, we define "derivatives" of these operations as finite differences, and then require that the corresponding continuous limit operations have exactly the same expressions for the derivatives [4].

The multiresolutional character of human reasoning also explains why in logic, only unary and binary operations are normally used: because although in principle, there exist ternary operations on $[0,1]$ (in the limit case) which cannot be represented as compositions of natural unary and binary ones, but on each resolution level, when we have only finitely many degrees, every operation can be naturally represented as such a composition [51].

D.2 More General Logics

The need for more general logics comes from the fact that just like experts are not sure about the statement S, they are also not sure about their own degrees of belief $d(S)$. Thus, instead of a single number $d(S)$, we can consider several possible numbers d, with degrees $d_2(d)$ describing to what extent these numbers are adequate descriptions of the original expert's uncertainty. This "second-order" approach has several successful applications. In principle, it is possible to go further and consider the fact that the degrees $d_2(d)$ are also not given precisely, so we seem to need the third-, fourth-order etc, approaches. However, in practice, such theoretically possible approaches turned out to be not useful. This fact can be explained if we take the multiresolutional character of reasoning into consideration:

• On the one hand, every "first-order" and "second-order" logic, in which the set of degree of belief is an ordered set, can be naturally described as a limit of an interval-related multiresolutional procedure [27], [28], [45], [76].

• On the other hand, if degrees come from words, then the third order is no longer necessary [30].

It is natural to select a continuous approach which best reflects the multiresolutional character of human reasoning, i.e., in which there is a qualitative difference between different pairs of degrees. A natural way to describe this difference in continuous case is to use the approach of non-standard analysis, with the actual infinitesimal elements (= lexicographic ordering). The optimal selection of such logics is described in [37], [54].

Conclusion

Interval mathematics is very helpful in the analysis of multiresolutional systems.

Acknowledgments

This work was supported in part by NASA under cooperative agreement NCC5-209 and grant NCC 2-1232, by the Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-00-1-0365, by NSF grants CDA-9522207 and 9710940 Mexico/Conacyt, and by the Grant No. W-00016 from the U.S.-Czech Science and Technology Joint Fund.

The authors are very grateful to Alex Meystel for his support, encouragement, and helpful discussions.

References

[1] G. Alefeld and J. Herzberger, *Einführung in die Intervallrechnung*, Bibliographisches Institut, Mannheim, Wien, Zürich, 1974.

[2] R. Al-Jamal, S. Manoli, A. E. Brito, and O. Kosheleva, "Interval + Communications = Walsh: For Signal Multiplexing Under Interval Uncertainty, Walsh Functions are Optimal", *Proc. the Joint 9th World Congress of the Int'l Fuzzy Systems Association and 20th Int'l Conference of the North American Fuzzy Information Processing Society IFSA/NAFIPS 2001*, Vancouver, Canada, July 25–28, pp. 1677–1681, 2001.

[3] J. Beck, V. Kreinovich, and B. Penn, "Geombinatoric aspects of processing large images and large spatial databases", *Geombinatorics*, Vol. 11, No. 4, pp. 6–12, 2001.

[4] B. Bouchon-Meunier, H. T. Nguyen, and V. Kreinovich, "Discrete (set) derivatives and 'algebraic' fuzzy logic operations", *Proc. FUZZ-IEEE'2001*, Melbourne, Australia, 2001 (to appear).

[5] A. E. Brito and O. Kosheleva, "Interval + Image = Wavelet: For Image Processing under Interval Uncertainty, Wavelets are Optimal", *Reliable Computing*, Vol. 4, No. 3, pp. 291–301, 1998.

[6] A. E. Brito, O. M. Kosheleva, and S. D. Cabrera, "Multi-Resolution Data Processing is Optimal: Case Study of Detecting Surface Mounted Devices", *Proceedings of the International Conference on Intelligent Systems and Semiotics (ISAS'97)*, National Institute of Standards and Technology Publ., Gaithersburg, MD, pp. 157–161, 1997.

[7] D. D. Coblentz, V. Kreinovich, B. S. Penn, and S. A. Starks, "Towards Reliable Sub-Division of Geological Areas: Interval Approach" *Proc. 19th Int'l Conference of the North American Fuzzy Information Society NAFIPS'2000*, Atlanta, Georgia, July 13–15, pp. 368–372, 2000.

[8] D. D. Coblentz, V. Kreinovich, B. S. Penn, and S. A. Starks, "Towards Reliable Sub-Division of Geological Areas: Interval Approach", In: L. Reznik and V. Kreinovich (eds.), *Soft Computing in Measurements and Information Acquisition*, Springer-Verlag, 2001 (to appear).

[9] L. O. Fuentes and V. Kreinovich, "Towards Intelligent Virtual Environment for Teaching Telemanipulation Operators: Virtual Tool Approach and its Interval-Based Justification", *Proceedings of the 2nd International Workshop on Intelligent Virtual Environments*, Xalapa, Veracruz, Mexico, September 11–12, pp. 8–11, 1998.

[10] F. Garcia, R. Araiza, and B. Rzycki, "Towards Optimal Mosaicking of Multi-Spectral Images", *Proceedings of the Second National NASA Student Conference on Earth Sciences, Aero-Space Technology, Human Exploration and Development of Space, Space Science, and Crosscutting Technologies*, Nashville, Tennessee, April 7–10, pp. 261–266, 2000.

[11] O. N. Garcia, V. Kreinovich, L. Longpré, and H. T. Nguyen, "Complex problems: granularity is necessary, granularity helps", In: Nguyen Hoang Phuong and Ario Ohsato (eds.), *Proceedings of the Vietnam-Japan Bilateral Symposium on Fuzzy Systems and Applications VJFUZZY'98*, HaLong Bay, Vietnam, 30th September–2nd October, pp. 449–455, 1998.

[12] S. Gibson, *An optimal FFT-based algorithm for mosaicking images*, Master Thesis, Department of Computer Science, University of Texas at El Paso, 1999.

[13] S. Gibson, O. Kosheleva, L. Longpre, B. Penn, and S. A. Starks, "An Optimal FFT-Based Algorithm for Mosaicking Images, With Applications to Satellite Imaging and Web Search" *Proceedings*

of the Joint Conferences in Information Sciences JCIS'2000, Atlantic City, NJ, February 27–March 3, Vol. I, pp. 248–251, 2000.

[14] S. Gibson, V. Kreinovich, L. Longpré, B. Penn, and S. A. Starks, "Intelligent Mining in Image Databases, With Applications to Satellite Imaging and to Web Search", In: A. Kandel, H. Bunke, and M. Last (eds.), *Data Mining and Computational Intelligence*, Springer-Verlag, Berlin, pp. 309–336, 2001.

[15] I. R. Goodman, R. A. Trejo, V. Kreinovich, J. Martinez, and R. Gonzalez, "An even more realistic (non-associative) interval logic and its relation to psychology of human reasoning", *Proc. Joint 9th World Congress of the International Fuzzy Systems Association and 20th International Conference of the North American Fuzzy Information Processing Society IFSA/NAFIPS 2001*, Vancouver, Canada, July 25–28, pp. 1586–1591, 2001.

[16] R. Hammer, M. Hocks, U. Kulisch, D. Ratz, *Numerical toolbox for verified computing. I. Basic numerical problems*, Springer Verlag, Heidelberg, N.Y., 1993.

[17] E. R. Hansen, *Global optimization using interval analysis*, Marcel Dekker, N.Y., 1992.

[18] J. R. Hobbs and V. Kreinovich, "Optimal Choice of Granularity In Commonsense Estimation: Why Half-Orders of Magnitude", *Proc. Joint 9th World Congress of the International Fuzzy Systems Association and 20th International Conference of the North American Fuzzy Information Processing Society IFSA/NAFIPS 2001*, Vancouver, Canada, July 25–28, pp. 1343–1348, 2001.

[19] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer, Dordrecht, 1996.

[20] R. B. Kearfott and V. Kreinovich (eds.), *Applications of Interval Computations*, Kluwer, Dordrecht, 1996.

[21] R. B. Kearfott and V. Kreinovich, "Where to Bisect a Box? A Theoretical Explanation of the Experimental Results", In: Goetz Alefeld and Raul A. Trejo (eds.), *Interval Computations and its Applications to Reasoning Under Uncertainty, Knowledge Representation, and Control Theory. Proceedings of MEXICON'98, Workshop on Interval Computations, 4th World Congress on Expert Systems*, México City, México, 1998.

[22] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, Upper Saddle River, NJ, 1995.

[23] O. Kosheleva, V. Kreinovich, and Y. Yam, "On the optimal choice of quality metric in image compression", *Proceedings of the International Symposium on Smart Structures and Microsystems (IS3M)*, Hong Kong, October 19–21, 2000, Paper C3-3.

[24] V. Kreinovich, "A simplified version of the tomography problem can help to estimate the errors of indirect measurements", In: A. Mohamad-Djafari (ed.), *Bayesian Inference for Inverse Problems*, Proceedings of the SPIE/International Society for Optical Engineering, Vol. 3459, San Diego, CA, pp. 106–115, 1998.

[25] V. Kreinovich and L. O. Fuentes, "Telemanipulation: The Virtual Tool Approach and Its Interval-Based Justification", In: G. Alefeld and R. A. Trejo (eds.), *Interval Computations and its Applications to Reasoning Under Uncertainty, Knowledge Representation, and Control Theory. Proceedings of MEXICON'98, Workshop on Interval Computations, 4th World Congress on Expert Systems*, México City, México, 1998.

[26] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational complexity and feasibility of data processing and interval computations*, Kluwer, Dordrecht, 1998.

[27] V. Kreinovich, M. Mukaidono, "Intervals (Pairs of Fuzzy Values), Triples, etc.: Can We Thus Get an Arbitrary Ordering?", *Proceedings of the 9th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'2000)*, San Antonio, Texas, May 7–10, Vol. 1, pp. 234–238, 2000.

[28] V. Kreinovich, M. Mukaidono, and K. Atanassov, "From Fuzzy Values To Intuitionistic Fuzzy Values To Intuitionistic Fuzzy Intervals etc.: Can We Get an Arbitrary Ordering?", *Notes on Intuitionistic Fuzzy Sets (NIFS)*, Vol. 5, No. 3, pp. 11–18, 1999.

[29] V. Kreinovich and H. T. Nguyen, "Granularity as an Optimal Approach to Uncertainty – A General Mathematical Idea With Applications to Sleep, Consumption, Traffic Control, Learning, etc.", *Proc. 19th Int'l Conference of the North American Fuzzy Information Society NAFIPS'2000*, Atlanta, Georgia, July 13–15, pp. 316–320, 2000.

[30] V. Kreinovich and H. T. Nguyen, "1st Order, 2nd Order, What Next? Do We Really Need Third-Order Descriptions: A View From A Realistic (Granular) Viewpoint", *Proceedings of the Joint 9th World Congress of the International Fuzzy Systems Association and 20th International Conference of the North American Fuzzy Information Processing Society IFSA/NAFIPS 2001*, Vancouver, Canada, July 25–28, pp. 1908–1913, 2001.

[31] V. Kreinovich and H. T. Nguyen, and W. Pedrycz, "How to Make Sure That "≈ 100" + 1 Is ≈ 100 in Fuzzy Arithmetic: Solution and Its (Inevitable) Drawbacks", *Proceedings of the Joint 9th World Congress of the International Fuzzy Systems Association and 20th International Conference of the North American Fuzzy Information Processing Society IFSA/NAFIPS 2001*, Vancouver, Canada, July 25–28, pp. 1653–1658, 2001.

[32] V. Kreinovich and J. Wolff von Gudenberg, "Arithmetic of complex sets: Nickel's classical paper revisited from a geometric viewpoint", *Geombinatorics*, Vol. 9, No. 1, pp. 21–26, 1999.

[33] V. Kreinovich and J. Wolff von Gudenberg, "An optimality criterion for arithmetic of complex sets", *Geombinatorics*, Vol. 10, No. 1, pp. 31–37, 2000.

[34] V. Kreinovich and Y. Yam, "Why Clustering in Function Approximation? Theoretical Explanation", *International Journal of Intelligent Systems*, 2001 (to appear).

[35] M. Krishna, *Using symmetries to reduce the size of a fuzzy knowledge base, with an application to fault detection*, Master Thesis, Department of Computer Science, University of Texas at El Paso, 1999.

[36] M. Krishna, V. Kreinovich, and R. Osegueda, "Fuzzy Logic in Non-Destructive Testing of Aerospace Structures", In: Jaime Ramírez-Angulo (ed.), *Proceedings of the 1999 IEEE Midwest Symposium on Circuits and Systems*, Las Cruces, New Mexico, August 8–11, 1999, Vol. 1, pp. 431–434.

[37] C. Langrand, V. Kreinovich, and H. T. Nguyen, "Two-dimensional fuzzy logic for expert systems", *Sixth International Fuzzy Systems Association World Congress*, San Paulo, Brazil, July 22–28, Vol. 1, pp. 221–224, 1995.

[38] J. Martinez, L. Macias, A. Esper, J. Chaparro, V. Alvarado, S. A. Starks, and V. Kreinovich, "Towards more realistic (e.g., non-associative) and- and or-operations in fuzzy logic", *Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference*, Tucson, Arizona, October 7–10, 2001 (to appear).

[39] N. J. McMillan, "Temporal and spatial magmatic evolution of the Rio Grande rift", *New Mexico Geological Survey Guidebook, 49th Field Conference*, Las Cruces County, 107–116, 1998.

[40] A. Meystel, "Planning in a hierarchical nested controller for autonomous robots," *Proc. IEEE 25th Conf. on Decision and Control*, Athens, Greece, pp. 1237–1249, 1986.

[41] A. Meystel, "Theoretical Foundations of Planning and Navigation for Autonomous Mobile Systems", *International Journal of Intelligent Systems*, Vol. 2, No. 2, pp. 73–128, 1987.

[42] G. A. Miller, "The magical number seven plus or minus two: some limits on our capacity for processing information", *Psychological Review*, Vol. 63, pp. 81–97, 1956.

[43] P. M. Milner, *Physiological psychology*, Holt, NY, 1970.

[44] R. Moore, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.

[45] M. Mukaidono, Y. Yam, and V. Kreinovich, "Intervals is All We Need: An Argument", *Proceedings of The Eighth International Fuzzy Systems Association World Congress IFSA'99*, Taipei, Taiwan, August 17–20, 1999, pp. 147–150.

[46] Hoang Phuong Nguyen, S. Starks, and V. Kreinovich, "Towards foundations for traditional oriental medicine", In: Nguyen Hoang Phuong and Ario Ohsato (eds.), *Proceedings of the Vietnam-Japan Bilateral Symposium on Fuzzy Systems and Applications VJFUZZY'98*, HaLong Bay, Vietnam, 30th September–2nd October, pp. 704–708, 1998.

[47] H. T. Nguyen and V. Kreinovich, *Applications of continuous mathematics to computer science*, Kluwer, Dordrecht, 1997.

[48] H. T. Nguyen, V. Kreinovich, and Q. Zuo, "Interval-valued degrees of belief: applications of interval computations to expert systems and intelligent control", *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems (IJUFKS)*, 1997, Vol. 5, No. 3, pp. 317–358.

[49] H. T. Nguyen, O. Kosheleva, V. Kreinovich, and L. Ding, "On the Optimal Choice of Quality Metric In Image Compression: A Soft Computing Approach", *Proceedings of the Sixth International Conference on Control, Automation, Robotics and Vision ICARCV'2000*, Singapore, December 5–8, 2000.

[50] H. T. Nguyen and V. Kreinovich, "Nested Intervals and Sets: Concepts, Relations to Fuzzy Sets, and Applications", In: R. B. Kearfott et al (eds.), *Applications of Interval Computations*, Kluwer, Dordrecht, pp. 245–290, 1996.

[51] H. T. Nguyen, V. Kreinovich, and I. R. Goodman, "Why

Unary and Binary Operations in Logic: General Result Motivated by Interval-Valued Logics", *Proceedings of the Joint 9th World Congress of the International Fuzzy Systems Association and 20th International Conference of the North American Fuzzy Information Processing Society IFSA/NAFIPS 2001*, Vancouver, Canada, July 25–28, pp. 1991–1996, 2001.

[52] H. T. Nguyen, V. Kreinovich, G. N. Solopchenko, and C.-W. Tao, "Why Two Sigma? A Theoretical Justification", In: L. Reznik and V. Kreinovich (eds.), *Soft Computing in Measurements and Information Acquisition*, Springer-Verlag, 2001 (to appear).

[53] H. T. Nguyen, V. Kreinovich, and C.-W. Tao, "Why 95% and Two Sigma? A Theoretical Justification for an Empirical Measurement Practice", *Proc. International Workshop on Intelligent Systems Resolutions: The 8th Bellman Continuum*, Taipei, Taiwan, December 11–12, pp. 358–362, 2000.

[54] H. T. Nguyen, V. Kreinovich, and B. Wu, "Fuzzy/probability $\sim$ fractal/smooth", *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems (IJUFKS)*, Vol. 7, No. 4, pp. 363–370, 1999.

[55] H. T. Nguyen, N. R. Prasad, V. Kreinovich, and H. Gassoumi, "Some Practical Applications of Soft Computing and Data Mining", In: A. Kandel, H. Bunke, and M. Last (eds.), *Data Mining and Computational Intelligence*, Springer-Verlag, Berlin, pp. 273–307, 2001.

[56] H. T. Nguyen and E. A. Walker, *First Course in Fuzzy Logic*, CRC Press, Boca Raton, FL, 1999.

[57] N. J. Nilsson, "Probabilistic logic", *Artificial Intelligence*, Vol. 28, pp. 71–87, 1986.

[58] R. A. Osegueda, C. Ferregut, V. Kreinovich, S. Seetharami, and H. Schulte, "Fuzzy (Granular) Levels of Quality, With Applications to Data Mining and to Structural Integrity of Aerospace Structures", *Proceedings of the 19th International Conference of the North American Fuzzy Information Society NAFIPS'2000*, Atlanta, Georgia, July 13–15, pp. 348–352, 2000.

[59] R. A. Osegueda, S. R. Seelam, A. C. Holguin, V. Kreinovich, and C.-W. Tao, "Statistical and Dempster-Shafer Techniques in Testing Structural Integrity of Aerospace Structures", *Proc. of International Conference on Intelligent Technologies*, Bangkok, Thailand, December 13–15, 2000, pp. 383–389.

[60] P. Santiprabhob, H. T. Nguyen, W. Pedrycz, and V. Kreinovich, "Logic-motivated choice of fuzzy logic operators", *Proc. FUZZ-IEEE'2001*, Melbourne, Australia, 2001 (to appear).

[61] M. Siddaiah, M. A. Lieberman, N. R. Prasad, and V. Kreinovich, "A geometric approach to classification of trash in ginned cotton", *Geombinatorics*, Vol. 10, No. 2, pp. 77–82, 2000.

[62] S. Srikrishnan, R. Araiza, H. Xie, S. A. Starks, and V. Kreinovich, "Automatic referencing of satellite and radar images", *Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference*, Tucson, Arizona, October 7–10, 2001 (to appear).

[63] S. A. Starks and V. Kreinovich, "Multi-spectral inverse problems in satellite image processing", In: A. Mohamad-Djafari (ed.), *Bayesian Inference for Inverse Problems*, Proceedings of the SPIE/International Society for Optical Engineering, Vol. 3459, San Diego, CA, pp. 138–146, 1998.

[64] S. A. Starks and V. Kreinovich, "Locating the whole pattern is better than locating its pieces: a geometric explanation of an empirical phenomenon," *Geombinatorics*, Vol. 8, No. 4, pp. 116–121, 1999.

[65] S. A. Starks and V. Kreinovich, "Aerospace applications of soft computing and interval computations (with an emphasis on multi-spectral satellite imaging)", In: M. Jamshidi, M. Fathi, and T. Furunashi (eds.), *Soft Computing, Multimedia, and Image Processing. Proceedings of the 2000 World Automation Congress WAC'2000, Maui, Hawaii, June 11–16, 2000*, TSI Press, Albuquerque, pp. 644–651, 2000.

[66] S. Subbaramu, A. Q. Gates, and V. Kreinovich, "Application of Kolmogorov Complexity to Image Processing: It Is Possible to Have a Better Compression, but It Is Not Possible to Have the Best One", *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, Vol. 69, pp. 145–150, 1999.

[67] R. Trejo and Vladik Kreinovich, "Error Estimations for Indirect Measurements: Randomized vs. Deterministic Algorithms For 'Black-Box' Programs", S. Rajasekaran, P. Pardalos, J. Reif, and J. Rolim (eds.), *Handbook on Randomized Computing*, Kluwer, 2001.

[68] K. Villaverde and V. Kreinovich, "A linear-time algorithm that locates local extrema of a function of one variable from interval measurement results," *Interval Computations*, No. 4, pp. 176–194, 1993.

[69] K. Villaverde and V. Kreinovich, "Parallel algorithm that locates local extrema of a function of one variable from interval measurement results", *Reliable Computing*, Supplement (Extended Abstracts of APIC'95: International Workshop on Applications of Interval Computations, El Paso, TX, Febr. 23–25, 1995), pp. 212–219, 1995.

[70] Q. Wen, A. Q. Gates, J. Beck, V. Kreinovich, and G. R. Keller, "Towards automatic detection of erroneous measurement results in a gravity database", *Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference*, Tucson, Arizona, October 7–10, 2001 (to appear).

[71] J. Wolff von Gudenberg and V. Kreinovich, "Candidate Sets for Complex Interval Arithmetic", In: Hrushikesha Mohanty and Chitta Baral (eds.), *Trends in Information Technology, Proceeedings of the International Conference on Information Technology ICIT'99, Bhubaneswar, India, December 20–22, 1999*, Tata McGraw-Hill, New Delhi, pp. 230–233, 2000.

[72] K. Worden, R. Osegueda, C. Ferregut, S. Nazarian, D. L. George, M. J. George, V. Kreinovich, O. Kosheleva, and S. Cabrera, "Interval Methods in Non-Destructive Testing of Aerospace Structures and in Mammography", *International Conference on Interval Methods and their Application in Global Optimization (INTERVAL'98), April 20–23, Nanjing, China, Abstracts*, pp. 152–154, 1998.

[73] K. Worden, R. Osegueda, C. Ferregut, S. Nazarian, E. Rodriguez, D. L. George, M. J. George, V. Kreinovich, O. Kosheleva, and S. Cabrera, "Interval Approach to Non-Destructive Testing of Aerospace Structures and to Mammography", In: Goetz Alefeld and Raul A. Trejo (eds.), *Interval Computations and its Applications to Reasoning Under Uncertainty, Knowledge Representation, and Control Theory. Proceedings of MEXICON'98, Workshop on Interval Computations, 4th World Congress on Expert Systems*, México City, México, 1998.

[74] K. Worden, R. Osegueda, C. Ferregut, S. Nazarian, D. L. George, M. J. George, V. Kreinovich, O. Kosheleva, and S. Cabrera, "Interval Methods in Non-Destructive Testing of Material Structures", *Reliable Computing*, Vol. 7, No. 4, pp. 341–352, 2001.

[75] website on interval computations
http://www.cs.utep.edu/interval-comp

[76] Y. Yam, M. Mukaidono, and V. Kreinovich, "Beyond [0,1] to Intervals and Further: Do We Need All New Fuzzy Values?", *Proceedings of The Eighth International Fuzzy Systems Association World Congress IFSA'99*, Taipei, Taiwan, August 17–20, pp. 143–146, 1999.

[77] Y. Yam, H. T. Nguyen, and V. Kreinovich, "Multi-Resolution Techniques in the Rules-Based Intelligent Control Systems: A Universal Approximation Result", *Proceedings of the 14th IEEE International Symposium on Intelligent Control/Intelligent Systems and Semiotics ISIC/ISAS'99*, Cambridge, Massachusetts, September 15–17, pp. 213–218, 1999.

[78] H. Xie, N. Hicks, G. R. Keller, H. Huang, and V. Kreinovich, "Automatic image registration based on a FFT algorithm and IDL/ENVI", *Proceedings of the ICORG-2000 International Conference on Remote Sensing and GIS/GPS*, Hyderabad, India, December 1–4, 2000.

An Autonomous Metric (Polytope-Convex Hull) For Relative Comparisons of MIQ

Daniel W. Repperger[∇]

[∇] Air Force Research Laboratory, AFRL/HECP, WPAFB, Ohio 45433, USA, D.Repperger@IEEE.ORG

Abstract. A means of measuring machine intelligence is presented. The technique is based on geometric procedures and works best on relative comparisons across different entities, rather than absolute comparisons of intelligence.

Key Words: Measuring Machine Intelligence

I. INTRODUCTION

Defining, evaluating, and obtaining viable metrics for the measurement of autonomy, machine intelligence quotient (MIQ), or intelligence, in general, is a nontrivial task [1-9]. It is generally agreed that intelligence must be a high dimensional vector involving multiple attributes of a human or machine (Meystel [1]). Defining the relevant dimensions is also not a trivial task and much controversy exists. Even the discussion on how testing on intelligence is performed with humans creates controversy on which mental abilities constitute intelligence. The relevant issues include whether the IQ obtained, e.g. by the Stanford-Binet Intelligence Scale or the Wechsler Scales, are fair measures. Additional controversy also exists that certain less privileged racial, ethnic, or social groups do not have fair representations on the test questions pertinent to their living environments.

Albus [2] defines intelligence as having many dimensions. He also recognizes degrees or levels of intelligence. Some of the influencing parameters in describing features of intelligence for unmanned ground vehicles include, but are not limited to:

(1) The computational power and memory capacity of the system's brain (or computer),

(2) The sophistication of the processes the system employs for sensory processing, world modeling, behavior generation, value judgment, communication, and,

(3) The quality and quantity of information and values the system has stored in its memory.

The measure of intelligence is also predicated on the success in solving problems, anticipating the future, and acting so as to maximize the likelihood of achieving goals. Obviously intelligence is goal oriented and related to success. The presumption is that different levels of intelligence produce dissimilar probabilities of success in the accomplishment of specific missions.

In studying autonomous systems [3], there are numerous (analogous) systems that can be examined for attributes both within and across processes that relate to autonomy. Some of these systems include living things (birds, fish, insects), intelligent highway vehicle systems, mobile robots, control of satellites in orbit, underwater vehicles systems, helicopters, tanks, human-machine interfaces, unmanned air vehicles, swarms of robots, and a host of other processes. In studying unmanned air vehicle systems (UAVs) [8,10] autonomy is desired since the goal is to maximize the ratio of UAVs/operators for a number of important reasons. The advantages include the significant reduction in cost, the elimination of the need to include a life support system (significantly reducing fuel and weight requirements), decreased vulnerability if the UAV is shot down or captured, enhancing reliability and robustness with multiple opportunities to achieve a mission, as well as other important traits. Again, in the design of UAVs, it is desired to have a metric to compare within and across different systems on the level of autonomy or intelligence designed in the aircraft.

It was pointed out in [4] that, at best, a measure of machine intelligence (MIQ) is a relative metric and it is difficult to have an absolute measure. This paper will discuss a relative means of determining how to contrast across different machines for comparative intelligence or autonomy. The goal is to have an objective measure to demonstrate that one machine has higher or lower degrees of intelligence or autonomy in comparison to another machine. Thus the designer can rate different machines in terms of their relative MIQ and investigate trade-offs between gain in MIQ versus cost and the benefits derived. It is cautioned that MIQ is very mission specific, and unless the mission can be accomplished with the appropriate level of success, then the machine may still not be appropriate. In other words, the appropriate tool has to be able to perform the given task. Success in a mission is the final measure that demonstrates that a machine has the appropriate MIQ for a given application. To understand the metric introduced here, some basics need to be reviewed and discussed to better grasp how the measure of MIQ was constructed herein.

II. Some Basic Definitions

To understand the ensuing definition of MIQ, some basic concepts need to be reviewed. We present the fundamental nomenclature via key definitions.

Definition 1 – Convexity:

A subset A of R^n is convex if, for any vectors **x** and **y** in A and scalars r and s with $r \geq 0$ and $s \geq 0$, $r + s = 1$, then every point r **x** + s **y** remains in A. In other words, if we have a convex set (2 dimensions) A with two points **x** and y, then if we draw a line from the point **x** to y, every point on the line remains inside the surface A.

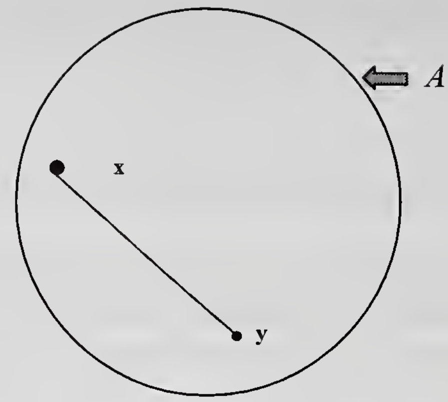


Figure 1 – The Convex Set A of a Circle

Figure 1 illustrates a circle in which the points **x** and **y** lie inside the circle. Drawing a line from the point **x** to the point **y** still remains inside the circle A. Also, every point along the line joining **x** to **y** also lies within the set A and no point on the line is outside the set A. Other examples of convex spaces in 3 dimensions include a cube, a sphere, etc. A cube is defined as follows:

$$\text{Cube} = A = \left\{ \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} : |x_1| \leq 1,\ |x_2| \leq 1,\ |x_3| \leq 1 \right\} \quad (1)$$

It is also worthwhile to look at a surface which is not convex. Example 1 describes a set of points, which is not convex.

Example 1–A set of points in a nonconvex set

The set A of points in R^2 defined by:

$$A = \left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} : x_1 \geq 0, x_2 \geq 0 \ \ni (x_1^{\frac{2}{3}} + x_2^{\frac{2}{3}})^{\frac{3}{2}} \leq 1 \right\} \quad (2)$$

Figure 2 is a plot of the nonconvex surface A. It is easily seen that a line cannot be drawn between any two points **x** and **y** in A and have every point on the line joining the points still lie in A. Thus the surface A in figure 2 is a nonconvex surface. Sometimes it is necessary to prove a surface is convex by the definition of its constituent elements. The following alternative definition is useful for this purpose.

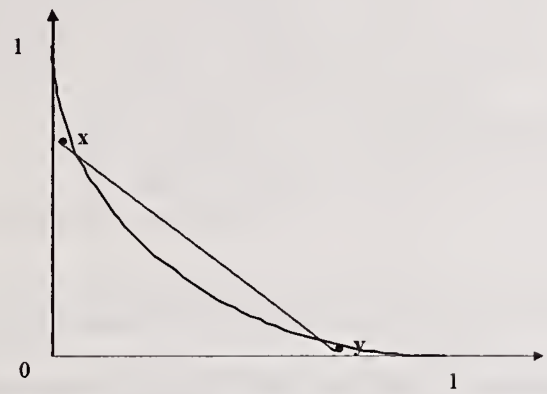


Figure 2 – A Nonconvex Set A

Alternative Definition of Convexity:
A function f(x) is convex if for all **x**, **y** and λ such that: $0 \leq \lambda \leq 1$,

$$f(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}) \quad (3)$$

The next three definitions will prepare for the appropriate definition of MIQ. Definition 2 refers to the outer surface (Convex Hull) that encloses the convex set.

Definition 2 – Convex Hull:
The convex cover (Convex Hull) of a convex set is what bounds the outside of the convex set. For figure 1, it is the circumference of the circle. For the cube of equation (1), the Convex Hull is the six surfaces of the cube. To define the Convex Hull more formally:
Let B be any subset of R^n and CH(B) is the convex hull of B if it contains all the convex combinations of the elements of B, i.e.

CH(B) = { **x** : there are elements $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ in B such that **x** is a convex combination of all of the $\mathbf{x}_i$ elements considered}

Hence the Convex Hull is the outside bounding surface of the convex set. The next definition generalizes this concept to multiple dimensions. Polytopes have many definitions, e.g. with respect to classes of polynomials [11], with respect to matrices [12], and also with reference to general convex-compact sets [13]. Here the choice is made to use the term polytope with respect to geometric figures. For a set of points in R^n where $n \geq 2$, the concept of convexity is now extended to multiple dimensions.

Definition 3 – Polytope:
Given the subset A of R^n which is a polytope if, for any vectors **x** and **y** in A and scalars r and s with $r \geq 0$ and $s \geq 0$, $r + s = 1$, then every point $r\,\mathbf{x} + s\,\mathbf{y}$ still remains in A. This generalizes for $n \geq 2$ and all points can be connected in A. Figure 3 illustrates a triangle as a 2 dimensional convex set and figure 4 generalizes this result to 3 dimensions. The goal is to increase n to any number greater than 2 and triangles or geometric figures with vertices will be used in each dimension.

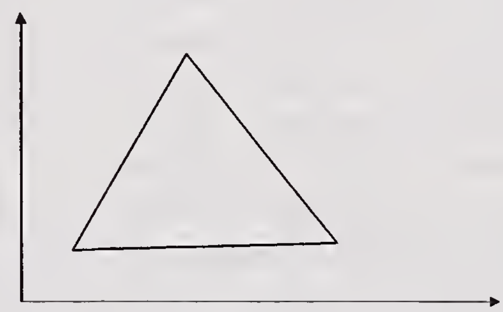


Figure 3 – A Triangle as a 2-dimensional Polytope

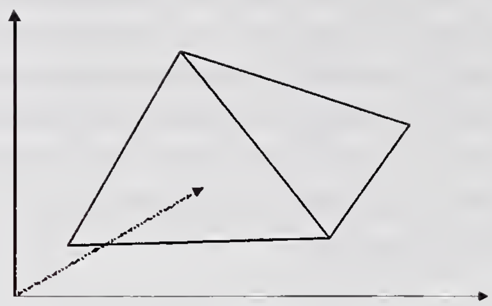


Figure 4 – A Triangle Extended as a 3-dimensional Polytope

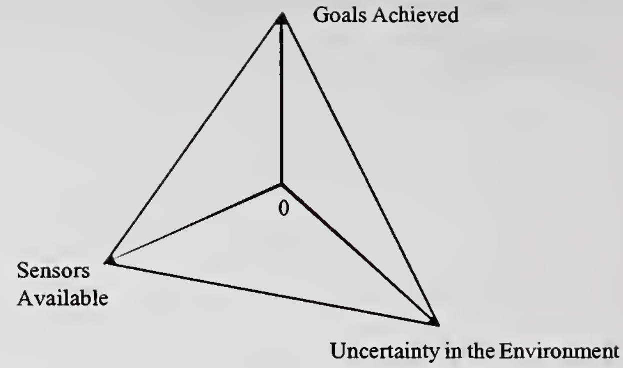


Figure 5 – First Definition of MIQ with 3 Axes (Intelligence Attributes)

Definition 4 – MIQ as a Polytope:

The prior definitions have provided some valuable tools to help in the definition of a measure of MIQ in multiaxes, as is necessary since intelligence is such a multidimensional process. There is a 3 step process in developing this methodology.

Step 1: Consider a minimum of 3 attributes for a 2 dimensional definition of MIQ.

Step 2: Generalize this result to 4 or more attributes in this 2-dimensional (planar space). In the two dimensional space, the map now extrapolates with any number of features necessary to complete the mission.

Step 3: The last step takes the generalization to a third or higher dimension. In all cases all the figures constructed are Convex Hulls or polytopes. Thus comparisons can always be made within any dimension involving two or more machines to be considered. To explain this better, figure 5 illustrates the Step 1 process with the 3 attributes of intelligence [2] being defined as: goals achieved (task performance), uncertainty in the environment, and sensors available. Figure 6 now extrapolates the previous figure to include a total of 5 attributes in the planar dimension with the addition of two more attributes of intelligence selected including: actuators controlled and *a priori* knowledge. Finally figure 7 generalizes to 3 dimensions with the addition of three additional intelligence attributes in the third dimension, including: accuracy level, time efficiency, and energy

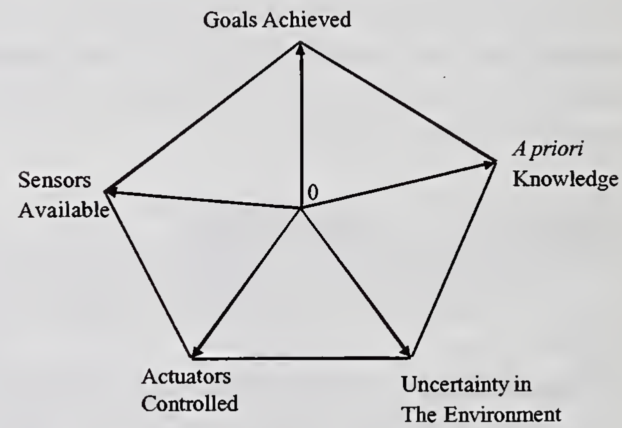


Figure 6 – Generalization of Figure 5 to now Include 5 Attributes

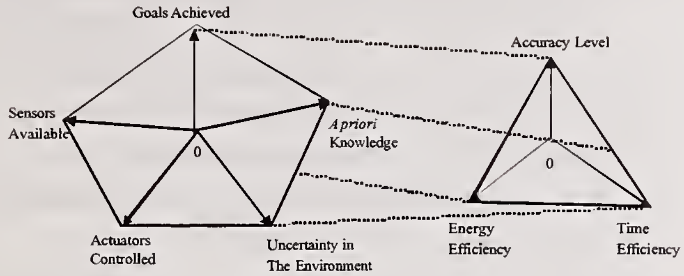


Figure 7 – Generalization of Figure 6 to 3 Dimensions With 8 Attributes.

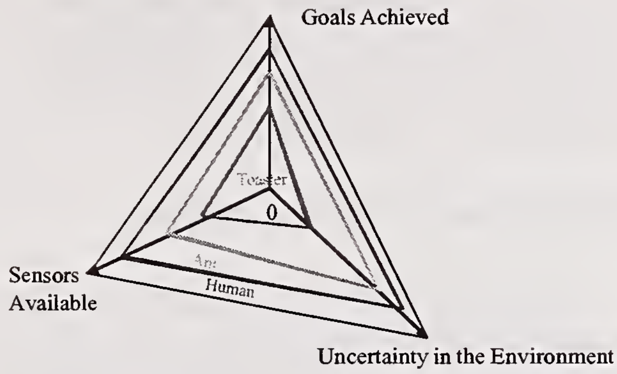


Figure 9 – Comparison of a Human, Ant and Toaster

efficiency [9]. To this point, the process has been an abstraction; in the next section a comparison is made of relative examples to illustrate how to use this methodology.

Methodology to Compare Across Machines

To illustrate how to use the methodology, four examples are considered with (presumed) increasing levels of intelligence (machine or nonmachine). They include:

(1) A toaster.
(2) A washing machine with fuzzy logic to detect quality of cleaning.
(3) An insect (ant).
(4) A human operator.

Due to the complexity of representation, figure 8 portrays a comparison of the washing machine with fuzzy logic to the toaster using the simplified planar representation

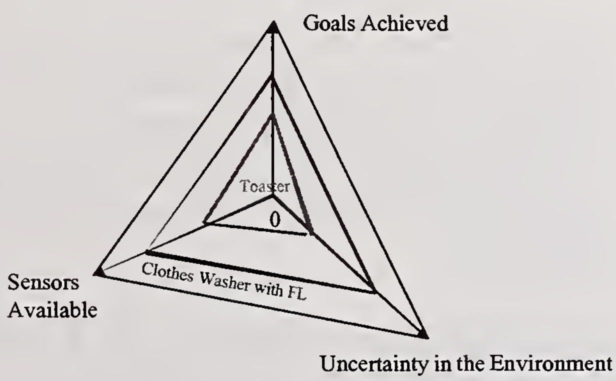


Figure 8 – Comparison of a Toaster and Clothes Washer With Fuzzy Logic

introduced in figure 5. Obviously the more intelligent machine is further displaced from the origin and due to the convexity of the polytope, it is seen that, in general, the fuzzy logic system appears to have greater machine intelligence (area measure). In figure 9, the evaluation of MIQ is now made between the mixture of living things and machines. The comparison involves a human, an ant, and a toaster. Here the relative hierarchy is specified by the amount of area or volume contained in each polytope. Thus the intelligence measure is very relative (not absolute) to compare across living things and machines. To summarize the results so far, the following paradigm is suggested on how to synthesize this MIQ metric:

Steps in Synthesizing the MIQ Paradigm:

(i) For the specific mission, define the axes of the polytope to be relevant to the performance of the mission under consideration (e.g. a toaster cannot clean a rug, nor can a washing machine toast a piece of bread).

(ii) Define the scales of each axes of the polytope relevant to the mission of interest.

(iii) Plot alternative machines on the same axes.

(iv) The hypervolume resulting will provide a relative (not absolute) comparison of the efficacy of a particular machine to perform certain missions.

Recall there is no absolute standard (however, an existing machine could be a baseline for comparison purposes) and, at best, the relevance of each machine to perform a specific mission can be better understood via this procedure.

III. Summary and Conclusions

Using properties such as convexity and relative measures of machine intelligence, the effectiveness to perform specific missions under various conditions can be determined. It is difficult to obtain an absolute measure of MIQ but by comparison to baseline or existing machines in use, there is some value in the relative comparison. The results can be extended to any level of complexity by considering convex polytopes in a multiple dimensional space.

References

[1] A. Meystel, "Measuring Performance of Systems with Autonomy: Metrics for Intelligence of Constructed Systems," *NIST Workshop on Performance Metrics for Intelligent Systems*, August 14-16, 2000, NIST, Gaithersburg, Maryland.

[2] J. S. Albus, "Features of Intelligence Required by Unmanned Ground Vehicles," *NIST Workshop on Performance Metrics for Intelligent Systems*, August 14-16, 2000, NIST, Gaithersburg, Maryland.

[3] D. W. Repperger, "Supervisory Control and Means of Designing Autonomy in Unmanned Vehicles," Tutorial, NAECON 2000, Dayton, Ohio, October, 2000.

[4] D. W. Repperger, "Using Intelligent Machines to Modify or Adapt Human Behavior," in *Intelligent Machines: Myths and Realities*, CRC Press LLC, 2000.

[5] A. Meystel, *Intelligent Systems – Annotated Bibliography and Survey*, Part 1-Systems with Intelligent Control, Part 2 – Systems with Learning, National Institute of Standards and Technology Report, 2001.

[6] H-J Park, B. K. Kim, and K. Y. Lim, "Measuring the Machine Intelligence Quotient (MIQ) of Human-Machine Cooperative Systems," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 31, No. 2, March, 2001, pp. 89-96.

[7] H. G. Kang and P. H. Seong, "Information Theoretic Approach to Man-Machine Interface Complexity Evaluation," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 31, No. 3, May, 2001, pp. 163-171.

[8] B. T. Clough, "Autonomous UAVs – Give Them Free Will and They Will Execute It!," *Proceedings of the AUVSI Unmanned Systems Meeting*, July, 2001.

[9] R. Finkelstien, "A Method for Evaluating the 'IQ' of Intelligent Systems," *NIST Workshop on Performance Metrics for Intelligent Systems*, August 14-16, 2000, NIST, Gaithersburg, Maryland.

[10] Tilden and Hasslacher, *From Living Machines*, Los Alamos National Laboratory, Report

[11] Y. Xiao, R. Unbehauen and X. Du, "Robust Hurwitz Stability Conditions of Polytopes of Bivariate Polynomials," *Proceedings of the 36th IEEE Conference on Decision and Control*, December, 1999. pp. 5030-5035.

[12] O. Bachelier, J. Bernussou, M. C. de Oliveira and J. C. Geromel, "Parameter Dependent Lyapunov Control Design: Numerical Evaluation," *Proceedings of the 36th IEEE Conference on Decision and Control*, December, 1999. pp. 293-297.

[13] C. E. de Souza, R. M. Palhares, and P. L. D. Peres, "Robust H-Infinity Filtering for Uncertain Linear Systems with Multiple Time-Varying State Delay: An LMI Approach," *Proceedings of the 36th IEEE Conference on Decision and Control*, December, 1999. pp. 2023-2028.

Decision-Making and Learning - Comparing Orthogonal Methods to Majority-Voting

Daniel W. Repperger[∇]

[∇] Air Force Research Laboratory, AFRL/HECP, WPAFB, Ohio 45433, USA, D.Repperger@IEEE.ORG

Abstract. A study on learning and decision-making methods was conducted by comparing an orthogonal methodology of manipulating data versus that of a majority-voting procedure. The latter method has recently become popular in the literature involving applications such as pattern recognition. To evaluate the differences between the proposed methods, data from a multidimensional paradigm involving decision-making and learning are analyzed. A number of basic concepts from estimation and information theory are first discussed to understand both the motivation and the underlining issues involved in conducting this study.

Key Words: Decision Making Methods

I. INTRODUCTION

Learning and decision-making are processes that adapt and are highly multidimensional [1]. Also when developing autonomous systems, there is considerable interest in adaptability as an intelligent means of modifying behavior as new data are acquired. Much like learning, decision-making to improve the quality of information has similar and related issues to designing intelligence in autonomous systems [1,2,3,4]. In a recent study [5], it has been demonstrated that it is possible to build a decision-making scheme from a "bottoms up" approach starting with a vector of orthogonal classifiers. Alternatively, a different approach involving classification and learning procedures occurs in pattern recognition schemes [6] where a scalar measure (majority-voting) can be compared to the hyperplane method as discussed in [5]. This paper will cover the basics of a decision-making process and how it can be generalized to learning by extrapolation of the techniques presented here. Both methods are highly adaptable, which is of interest in a number of special applications, and, in particular, for intelligent control methods involving the design of autonomy. First it is important to discuss some well-known results from estimation and information theory which motivate the orthogonal approach discussed here.

In estimation theory (e.g. in Kalman filtering) the concept of orthogonal projection is well-known. An optimal estimator is recognized as having its error vector orthogonal to the direction of the measurement signal. Another interpretation of this result is that the residuals (difference between the data and the estimator) should contain zero information (the residuals are random) and are not correlated with the state estimate [7]. Hence one can view learning as a process of making the residuals white (containing no information) and the error of a state vector remaining orthogonal to the measurement set. Thus learning can proceed, as new data are received, by updating the estimator, accordingly, so that the resulting residuals still contain minimal information. This is also consistent with information theory concepts in which the greatest information is contained in the most unlikely event and there is little new information in an expected event [8].

When multiple channels of data tell the observer their potential classification of a particular object, the decision can be predicated on the orthogonal approach or possibly on the majority vote of scalar classifiers. There are two distinct points of view:

(1) The first and traditional method (vector) is that an optimal estimator can be built which employs an orthogonal method described above. As new data arrive, the estimator is adapted so that the resulting error vector remains orthogonal to the measurement set. This methodology is not necessarily a scalar process and hyperplanes can describe the estimator when any number (n) of channels of data are available.

(2) The second possibility (scalar) is that a majority-voting scheme could be employed. This differs from the method (1) because of n (initially assumed to be odd) channels of data could each individually select (binary decision rule) their choice of a decision on the classification of an object. The overall decision is then based on the majority of the decisions. This second method is a scalar mapping; the first method involves a hyperplane or vector methodology. It has been shown mathematically [6] that the second method can be as effective or better than the first method in certain situations. This paper will examine the relevant details why learning or decision-making may benefit from a majority viewpoint in contrast to an orthogonal perspective. First the basics of each of these processes are reviewed.

II. Examples Considered

To better understand the relevant issues, the basics are reviewed utilizing well-known results involving information theory, Kalman Filtering, and orthogonal pattern recognition procedures. The goal is to compare both across and within different methodologies to see similarities and differences on why certain methods may help adapt in learning and why a majority-voting scheme has some merit. The first example arises from the basic mathematical discussion of orthogonal projection.

2.1 Optimality and Orthogonal Projection

To provide the background to this approach, it is first instructive to show the fundamental relationship between optimality and orthogonal projection. Given a linear space X with inner product $<x, y>$ defined for any two elements using the L_2 norm:

$$\| x \| = < x,x >^{1/2} \qquad (1)$$

A fundamental theorem is borrowed from the classical literature in this area [9].

Theorem 1: $\| x - \hat{y} \|$ is a minimum for all $y \varepsilon M$ (the measurement set), i.e.

$$\| x - y\| \geq \| x - \hat{y} \| \quad \forall\, y \varepsilon M \qquad (2)$$

if and only if $(x - \hat{y})$ is orthogonal to all $y \varepsilon M$, i.e.:

$$< x - \hat{y} ,y > = 0 \quad \forall\, y \varepsilon M \qquad (3)$$

Proof:

First assume equation (3) is valid, then for any $y \varepsilon M$,

$$\| x - y \|^2 = \| (x - \hat{y}) + (\hat{y} - y) \|^2 \qquad (4)$$

$$= \| x - \hat{y} \|^2 + 2 < (x - \hat{y}), \hat{y} - y > + \| \hat{y} - y \|^2 \qquad (5)$$

where each $(y - \hat{y}) \varepsilon M$. But from equation (3), the middle term of (5) vanishes yielding:

$$\| x - y \|^2 = \|(x - \hat{y})\|^2 + \| (\hat{y} - y) \|^2 \qquad (6)$$

$$\geq \| (x - \hat{y}) \|^2 \qquad (7)$$

with equality if and only if $y = \hat{y}$. To complete the proof, (assume (3) is not valid) and that $\hat{y}$ minimizes $\| x - y \|^2$ for all $y \varepsilon M$, hence there exists some $y_1 \varepsilon M$ such that:

$$< x - \hat{y}, y_1 > = \alpha \neq 0 \qquad (8)$$

Then: $\| x - \hat{y} - \beta y_1 \|^2 =$

$$\| (x - \hat{y}) \|^2 - 2\, \alpha \beta + \beta^2 \| y_1 \|^2 \qquad (9)$$

Thus it appears that by appropriate choice of β it is possible to make the combined total of the last two terms of (9) negative, thus contradicting the minimality of $\hat{y}$. Hence such an element y_1 of M cannot exist and this shows the optimality criterion.

Remark:

The relationship between optimality and orthogonality is immediately evident. The orthogonal component **y** clearly minimizes the function:

$$J_1 = \min \| \mathbf{x} - \mathbf{z} \| \qquad (10)$$

over the set of vectors **z** in M as illustrated in the proof of this theorem. Thus if the goal is optimality (in the sense of minimum distance), then the orthogonal projection provides a viable solution. Next, this concept is described in terms of the well-known Kalman filter and the principle of orthogonal projection.

2.2 An Example from Estimation Theory (Kalman Filter):

The well-known Kalman filter was derived using the concept of orthogonal projection [7,9,10]. For brevity, only the basic details are presented here. Let $\hat{x}$ denote the estimate of the state vector x as the solution of the optimal linear filtering problem. The error is $\tilde{x} = x - \hat{x}$. Using the expectation operator notation, the optimal estimator at time t_1, provided by measurements $z(t)$ up to time t, satisfies the following two important properties:

(a) $E\{ \hat{x}(t_1/t) \} = E\{ x(t_1) \}$

(b) $\min E\{ \| \tilde{x}(t_1|t) \|^2 \}_B$ is achieved.

The matrix B is a positive definite matrix. The orthogonal projection lemma relates to the above conditions as follows:

Orthogonal Projection Lemma for the Optimal Linear Estimator

The optimal estimator satisfying conditions (a,b) above also satisfies the following orthogonality condition [7,9,10]:

$$E\{ (\tilde{x}(t_1|t))(z(t_1)) \} = 0 \qquad (11)$$

Remark: The Optimal Linear Estimator can also be derived from Theorem 2 [10]:

Theorem 2:

A necessary and sufficient condition for the linear estimator $\hat{x}$ to be the least squares (minimum variance) estimate is that

$$E\{ \hat{x}(t_1|t)\} = E\{x(t_1)\} \qquad (12)$$

$$E\{ (\tilde{x}(t_1|t))(z(t_1)) \} = 0 \qquad (13)$$

In other words, if the estimator is unbiased (12) and orthogonal (13) to the measurement set, this is sufficient to minimize the least squares deviations. Hence orthogonality, linearity, and being unbiased are sufficient to guarantee optimality. We represent this concept in Figure 1 which portrays the error signal ($\tilde{x}(t_1|t)$), the measurement vector $z(t_1)$, and their orthogonal relationship. There is an interesting geometric interpretation in Figure 1 which elucidates the concept considered in this paper.

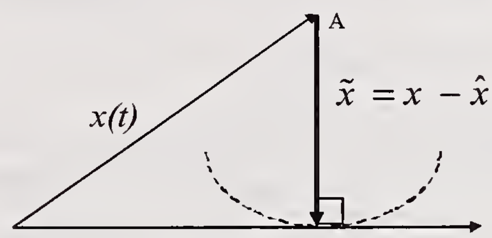


Figure 1 - Orthogonality Relationship between $z(t)$ and $\tilde{x}$

Geometric Interpretation of Figure 1:

In Figure 1, one can view optimality in terms of a distance measure. Starting at point A as a center, a radius is drawn with length $\tilde{x}(t_1|t)$ as indicated by the arc. It has been known since the time of early Greece that the shortest distance from point A to the measurement vector $z(t)$ (line) occurs if the radius is perpendicular to $z(t)$. Hence from a geometric perspective, the orthogonal projection is the minimum distance from a point to a line and the relationship between optimality and orthogonality is easily understood.

The next example is gleaned from information theory and insight is gained on how to relate this prior work on estimation theory to the information theory methods.

2.3 An Example from Information Theory

The approach here will be to synthesize a very complete model of an information channel to account for an assortment of possible losses and gains of information through a variety of processes [11]. The definition of the information $I(x\,;\,y)$ given by an observed event y about a hypothesis x can be specified in a probability sense as follows:

$$I(x\,;\,y) \;=\; \log_2 \frac{p(\,x|\,y)}{p(x)} \text{ (bits)} \qquad (14)$$

The input set of x's is defined as the discrete and finite set X, and the output set of y's, correspondingly, is defined as Y. In figure 2, a flow graph (the information channel is inside the dashed box) is constructed with the following variables defined, accordingly:

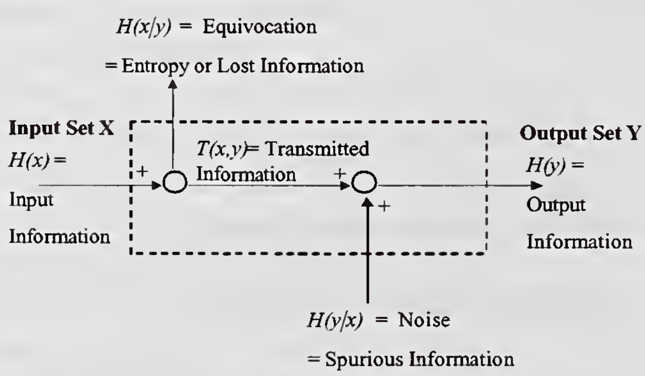


Figure 2 - The Flow of Information Through A Channel

H(x) = Input information in the set X (the information content of the set X).
H(y) = Output information in the set Y (the information content of the set Y).
H(y/x) = The noise added to the information channel (spurious information).
H(x/y) = The equivocation (entropy) which is the information about the input set X that might have been transmitted but was not.
T(x,y) = The transmitted information.

Some other interpretations of these key quantities can be stated. For example, *H(x)* is the input information provided in the source and *H(y)* is the output information received. The equivocation can be viewed as the average information still needed to specify an x exactly after the evidence y has been taken into account. The term average or expected value of information is derived from the fundamental definition of *H(z)* which is in the form of an expected value operation on information specified via:

$$H(z) \triangleq \sum_i p(z_i)\,\log_2 \frac{1}{p(z_i)} \text{ (bits)} \qquad (15)$$

Figure 2 displays the following equation representations of these different types of information measures:

$$H(x)-H(x\,|\,y)=T(x,y)=H(y)-H(y\,|\,x) \quad (16)$$

From figure 2, for a given information channel, the input information *H(x)* and the spurious information *H(y/x)* are generally fixed and specified. The best the designer can hope to accomplish is to reduce the uncertainty (*H(x/y)* = entropy or equivocation) by the choice of some design parameter or procedure. Two productive results occur if *H(x/y)* is reduced:

(a) The transmitted information *T(x,y)* is increased.
(b) The received or output information *H(y)* increases.

Hence reducing entropy or uncertainty, by any means possible, can only help to improve the quality of the decision-making or learning. For an autonomous or intelligent system, this can surely expand one dimension of intelligence by the means in which a decision is made. It will be shown in the sequel that the orthogonal procedure can also be viewed as an entropy reduction procedure.

To illustrate how decision-making can be realized from only an orthogonal approach, an example from pattern recognition is now introduced. Two approaches will be utilized to solve this problem. The first approach will be the construction of an orthogonal, hyperplane methodology. The second line of attack will introduce the procedure termed "majority-voting".

2.4 An Example from Pattern Recognition (Orthogonal Method)

A system is described which provides a means for improving the quality of information derived from a decision-making process by weighing certain multiple and alternative information channels. The method is applied to data estimating the cognitive workload state of a human operator dealing with a complex task using noninvasive sources of physiological data as a basis.

In recent years, as the proliferation of data becomes more and more persuasive, the challenge increases in designing systems that can process information in an innovative and efficient manner. The first system discussed in this paper has as a goal the improvement of the quality of information for making a decision from alternative (and multiple) sources of data. The potential data sources are first rank ordered in terms of their efficacy for making a binary decision. The next step is to combine two alternative data sources in a productive manner so as to glean out the highest quality information. By induction, the process then generalizes to multiple, alternative, data sources with the end goal of continuing to improve the decision-making process through the intelligent use of data. To illustrate the applicability of the approach, data relevant to the estimation of the state of an operator (human controlling an automated system) through the selection of certain, key, physiological signals provides a platform to test the efficacy of such a methodology [12].

As humans deal with highly automated and complex systems, it is sometimes desired to obtain estimates of elevated demands of cognitive workload as manifested by physiological signals that may be gleaned in a noninvasive manner. Once an identification of the operator in a high workload state is verified, the automation level of the system may be adjusted to maintain effectiveness of the mission [2,11]. Figure 3 illustrates the operator in a human-machine interaction system with physiological data being monitored. Figure 4 depicts the basis of the decision rule (low or high workload state) that will be investigated in this study with the goal of improving decision-making by using multiple channels of data in a productive sense. In Figure 4, the data displayed may be from as many as 43 possible physiological

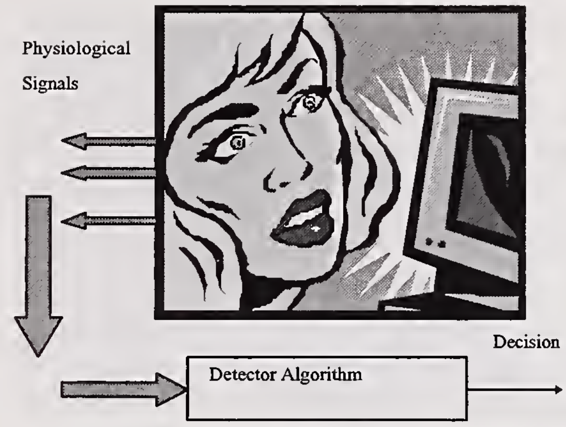


Figure 3- Physiological Signals to Detect Workload

signals, which are obtained in a noninvasive manner.

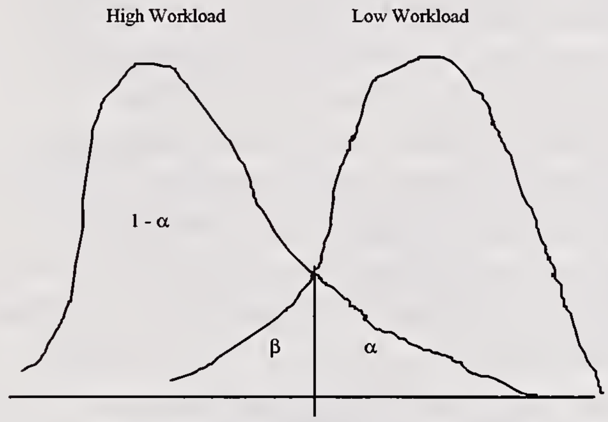


Figure 4- The Basis for The Decision Rule

2.5 The Statistical Decision Rule

Figure 5 portrays the ROC (relative operating characteristic) curve for data representative of figures 4 and 6.

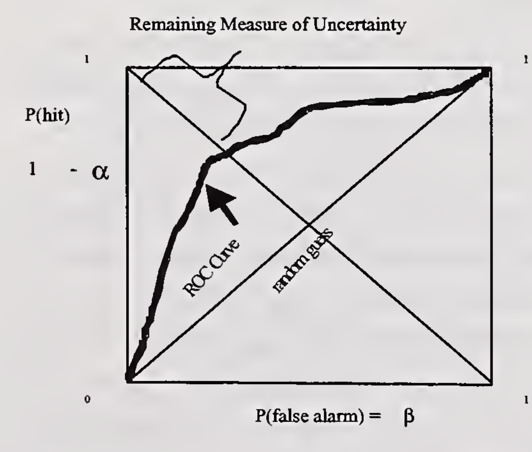


Figure 5 – The ROC Curve

The ROC was originally derived in signal detection theory, but has found widespread use in other areas. The plot in Figure 5 has

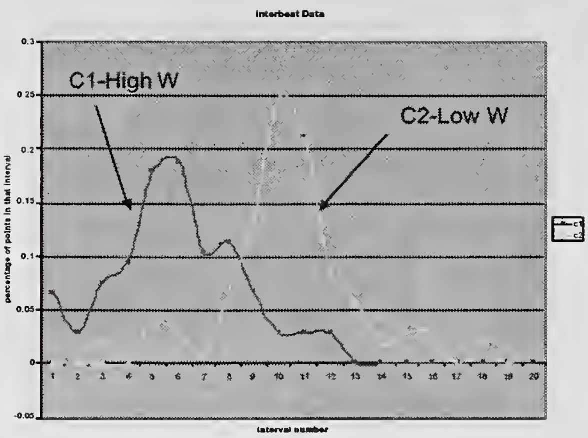


Figure 6 – Interbeat Heart Rate Data

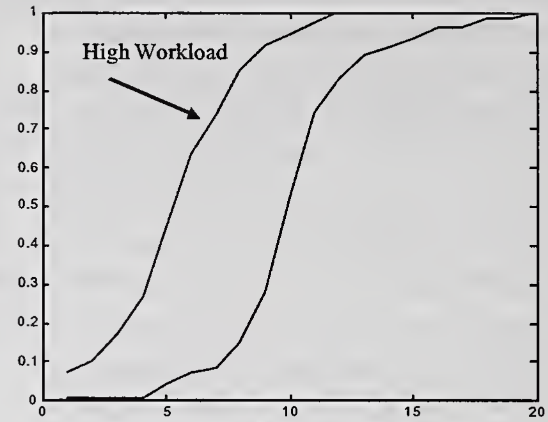


Figure 7 – CDFs for Figure 6 Data

as the dependent variable the term 1-α versus the independent variable β as derived from Figure 4. This may be viewed as the plot of the probability of a hit versus the probability of a false alarm in a binary decision rule [2,11,13] and can be shown to be the depiction of the two cumulative distribution functions of the densities of Figure 4. In an ideal decision-making process, the ROC curves moves upward to the left most diagonal (a measure of uncertainty, cf. Figure 5). Performance measures of such systems may be the minimum diagonal distance proximal to the upper left diagonal or the area under the ROC curve. An application to test the algorithm presented here is next described.

2.6 Testing the State of the Human Operator

From [12] there exist 43 possible data channels including physiological variables such as interbreath, interheart beat, and various electrode signals obtained as an operator performs a difficult task. Figure 6 illustrates the interbeat data for the two-workload conditions (high and low) and Figure 7 is the resulting cumulative distribution functions. Figure 8 is the corresponding ROC curve. Since the ROC curve is above the diagonal (random guess), this data variable is useful for predicting the state of the operator. The challenging problem discussed here is how to use two or more alternative data channels to improve upon the decision-making capability. After this procedure is illustrated for two channels, by induction, the process then generalizes to n channels.

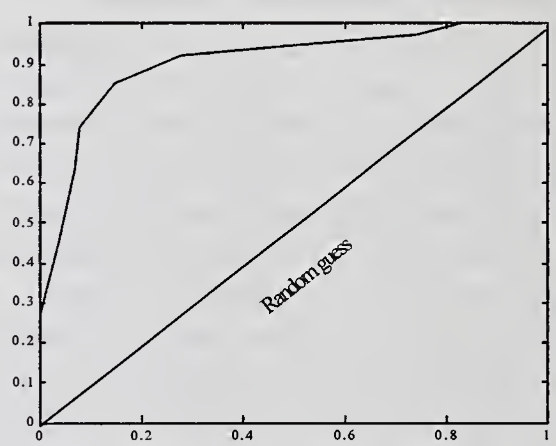


Figure 8 – Resulting ROC Curve for Figure 6 Data

2.7 The Orthogonal Algorithm

The algorithm to develop the decision rule has two steps:

Step 1: Rank order all data variables using the ROC curve.

Step 2: Select two or more data variables that yield a productive ROC curve, and then develop cross plots of the distributions. The decision rule is the hyperplane that separates the two distributions in an appropriate manner. Appropriate is based on an orthogonal projection between the centroids of the candidate distributions [14].

2.8 Implementation

Step 1 was implemented by plotting 43 ROC curves for all the data variables of

interest. The efficacy (objective metric) was the minimum distance along the diagonal from the upper left corner to the ROC curve (cf. Figure 5). Thus all 43 data channels could be rank ordered, according to their ability to improve on the binary decision rule.
Step 2 was implemented by developing cross plots of two candidate distributions. The centroids were then calculated for each distribution. A line was drawn between the centroids. A perpendicular line was then constructed to separate the two distributions at a point determined by a ratio involving the distance of the respective ROC curves from their upper left corner on the diagonal in Figure 5. This decision rule then generalizes to a hyperplane as more variables are included. The overall decision rule (cf. Figures 9 and 10, for example) is that the selection is made of the high workload condition if the points fall below the hyperplane. Above the hyperplane is considered the low workload condition. The results then generalize to multiple channels of data and the decision rule is a vector based on ROC curves and hyperplane surfaces as shown in Figure 10 for any number of data channels. Also this method can be viewed as a means of reducing entropy by expanding the dimension set. In multiple dimensions, the entropy (lost information) is constantly reduced when the hyperplane includes more discriminate points in an n dimensional space.

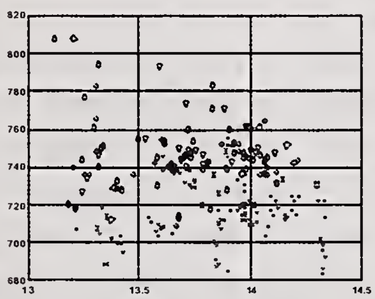


Figure 9 – Separating The Workload Data

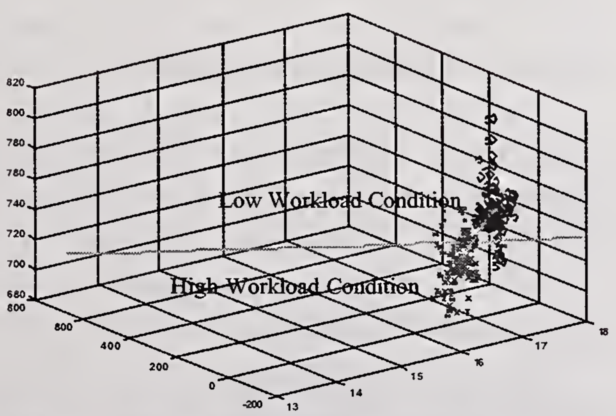


Figure 10 – Construction of A Decision Hyperplane

2.9 An Example from Pattern Recognition (Majority-Voting Procedure)

It has been shown mathematically [6] that a highly simple (scalar) algorithm can perform as well or better than an orthogonal scheme just described. Figure 11 displays a bank of classifiers (n is assumed to be an odd number). Each classifier makes an individual

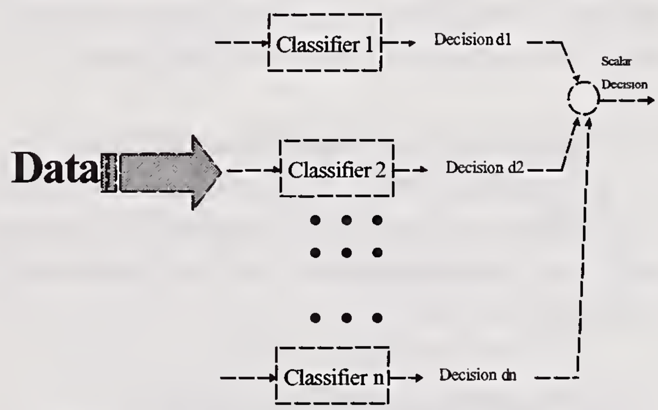


Figure 11 – Majority-Voting – A Scalar Decision-Making Process

decision on the binary decision rule. The overall decision is simply the majority vote of these n classifiers. The advantages and disadvantages of this procedure are briefly described:

2.10 Advantages of the Majority-Voting Procedure

Obviously, simplicity and the scalar nature of the process described in figure 11 is attractive, since computationally this process is much easier. Simplicity usually includes the attributes of reliability and robustness.

2.11 Disadvantages of the Majority-Voting Procedure

The disadvantage of the configuration in figure 11 occurs if the number of classifiers is small or does not fully represent the probability space concerning the important variables required in making a decision. If the number of classifiers $n \rightarrow \infty$, then it is obvious that the appropriate variables will be considered. This is analogous to the problem of "persistence excitation" in adaptive control theory. If, however, the system does not fully exploit the entire information set, then erroneous results may occur. Hence incorrect outcomes will occur if n is sufficiently small or does not include relevant information for making a key decision. We study the results with the application discussed previously.

III. Application to Experimental Data

Using data from [12] workload estimation of the human operator, the orthogonal method will be compared to a majority-voting scheme.

3.1 Comparison of the Orthogonal Approach to Majority-Voting

The comparison between these two sets of classifiers was conducted by studying three classifiers with a different data set as input to each classifier. This system was tested in an orthogonal sense as well as with the majority-voting scheme. The three selected physiological data sets from the 43 possible included: (1) interbeat (heart rate data), (2) electrode zero- alpha (the alpha brain wave from an electrode denoted as zero), and (3) electrode one- delta (the delta brain wave from the electrode denoted as number 1). It is noted that there were three nonelectrode data channels (interbeat, interbreath, and eyeblink) and 8 electrodes with 5 channels each of brain-wave data recorded. This gave a total of 43 channels of data possible to detect whether the operator was in a state of high or low workload. As these data were collected, the operator performed tasks, which were known to elicit a state of high or low workload by the task's relative complexity and subjective comments collected.

The ROC curves of figure 5 were determined for all three data sets. The variable σ will be used to measure the distance from the diagonal to the upper left hand corner of the ROC curve along the vertical axis. Note $0.5 \geq \sigma \geq 0$ because a random guess line is described by the diagonal that goes from the (0,0) point to the (1, 1) in figure 5 and the efficacy of the estimator is the proximity of the ROC curve intersecting the diagonal going from (0,1) to (1,0). Four tests were performed. The classifiers were rank ordered by their σ values (the smaller σ is a better estimator). The orthogonal method and the majority voting method were both utilized to classify 210 points (106 in the high workload case and 104 in the low workload case). Table 1 shows the efficacy of the classifiers, alone. It lists the data utilized and the σ value for each classifier.

Table1–Efficacy of A Classifier Acting Alone

Classifier Number	Data Variable Utilized	σ from the ROC Curve
Classifier - 1	Interbeat (heartrate) data	0.15
Classifier - 2	Electrode 1- delta wave	0.27
Classifier - 3	Electrode – 0 – alpha wave	0.32

Thus as the classifier number increases, its ability to perform accurate decision-making degrades accordingly. The performance of these classifiers is now evaluated in both an orthogonal sense as well as in a majority-voting scheme. In Table 2, the errors e_1 represent the data points that were high workload but were wrongly classified as low workload. The errors e_2 represent the data points that were low workload but were wrongly classified as high workload. The errors e_3 were the errors the majority voting scheme wrongly classified in either case. The overall performance results are displayed in Table 2. For two classifiers, the majority-voting scheme was considered inaccurate if both classifiers did not reach the same conclusion.

Table 2 – Performance of The Orthogonal Method versus Majority-Voting

Tests and Classifiers	e_1 errors	e_2 errors	e_3 errors
Test 1: C1 + C3	12	24	30
Test 2: C1 + C2	14	21	28
Test 3: C2 + C3	8	0	8
Test 4: C1+C2+C3	4	0	6

IV. Discussion of Results

From Table 2, some interesting results appear. When two classifiers are considered, the majority-voting scheme performs as well or better than the orthogonal method. As we go to higher dimensions, however, (Test 4), the combined effect of e_1 and e_2 errors is less for the orthogonal method as compared to the majority scheme. Also the Test 3 results are interesting because this is a poor estimator, yet the orthogonal projection scheme seems to include the relevant aspects of the decision-making space. The benefits of increasing the dimension of the orthogonal classifier seem to outweigh the benefits derived from the majority-voting scheme. As n gets larger, it appears this effect is more pronounced. Studies on ongoing to further investigate the dimensionality effect both within and across these candidate classifiers.

References

[1] A. Meystel, *Intelligent Systems – Annotated Bibliography and Survey*, Part 1- Systems with Intelligent Control, Part 2 – Systems with Learning, National Institute of Standards and Technology Report, 2000.

[2] D. W. Repperger, "Using Intelligent Machines to Modify or Adapt Human Behavior," Chapter 9 in *Intelligent Machines: Myths and Realities*, CRC Press LLC, 2000.

[3] H-J Park, B. K. Kim, and K. Y. Lim, "Measuring the Machine Intelligence Quotient (MIQ) of Human-Machine Cooperative Systems," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 31, No. 2, March, 2001, pp. 89-96.

[4] H. G. Kang and P. H. Seong, "Information Theoretic Approach to Man-Machine Interface Complexity Evaluation," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 31, No. 3, May, 2001, pp. 163-171.

[5] D. W. Repperger and C. A. Russell, "A System to Improve the Quality of Information Gained from Multiple Data Sources," *Proceedings of the IEEE 44th International Midwest Symposium on Circuits and Systems,* August 14-17, 2001, Dayton, Ohio.

[6] L. Lam and C. Y. Suen, "Application of Majority-Voting to Pattern Recognition: An Analysis of Its Behavior and Performance," *IEEE Transactions on Systems, Man, and*

Cybernetics – Part A: Systems and Humans, Vol. 27, No. 5, September, 1997, pp. 553-568.

[7] T. Kailath, *Linear Systems*, Englewood Cliffs, New Jersey: Prentice-Hall, 1980.

[8] M. Mansuripur, *Introduction to Information Theory*, Prentice Hall, Inc., 1987.

[9] R. C. K. Lee, *Optimal Estimation, Identification, and Control*, The MIT Press, 1964.

[10] M. S. Grewal and A. P. Andrews, *Kalman Filtering - Theory and Practice*, Prentice Hall, 1993.

[11] T. B. Sheridan and W. R. Ferrell, *Man-Machine Systems – Information, Control, and Decision Models of Human Performance*, The MIT Press, 1974.

[12] Greene, K. A., Bayer, K. W., Kabrisky, M., Rogers, S. K., Russell, C. A., and Wilson, G. F., "A Preliminary Investigation of Selection of EEG and Psychophysical Features for Classifying Pilot Workload" *In Intelligent Engineering Systems through Artificial Neural Networks*, vol. 6. of *Proceedings of Artificial Neural Networks in Engineering International Conference*, 1996, pp. 691-697.

[13] E. Riess, "Automatic Monitoring of Electrical Parameters in the Semiconductor Industry Based on ROC," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 30, No. 6, November, 2000, pp. 853-857.

[14] J. Yen and R. Langari, *Fuzzy Logic – Intelligence, Control, and Information*, Prentice Hall, 1999.

A TOP DOWN THEORY OF LOGICAL MODELING

John Shosky, Ph.D.

AmericanUniversity, USA

INTRODUCTION

Scientists, logicians, mathematicians, and linguists are among those who employ models. Yet, there are various views of models. For example, Quine has defined models as "a sequence of sets"[1] and van Fraassen sees them as "specific structures, in which all relevant parameters have specific values."[2] Harré argues that they can be either theoretical, as in a "set of sentences which can be matched with sentences in which the theory is expressed" or iconic, "some real or imagined thing, or process, which behaves similarly to some other thing or process, or in some other way than in its behaviour is similar to it."[3] The variation in these definitions reflects the many uses of models. The common ground between these definitions is that a model is an analogy, or a "relationship between two entities, processes, or what you will, which allows inferences to be made about one of the things..."[4] Traditional models share a mapping function in which the model and the system it compares stand in an analogical relationship, inviting horizontal comparisons and analysis. Models have been important in the development of logic, especially modal logics.[5] In science, they are "the [1]very basis of scientific thinking."[6]

Yet, the dangers of such "bottom up" analogical approaches are well known and lurking in the background of any serious discussion about the appropriate use of modeling. The analogy of the system under examination is always an artificial construct. Various competitors rival the model, with success based on the best analogy. Hence, analogy becomes the primary task, and problem. A model is developed through a theory-laden process that involves assumptions about initial conditions and applicable laws. It is hard to separate out those positive areas of the model that are similar to the system under analysis from the

---

negative areas that do not correspond to the system. Comparing the properties of the two systems is not enough. Analogical reasoning does not occur in a vacuum. Also, trivial and non-trivial modeling invites difficulties because structural isomorphism is not enough to account for similarity. There many be an endless number of systems that exhibit similarity. Then, of major concern, the appearance of possible counterfactuals may doom the modeling enterprise.[7]

But modeling is vital, often indispensable. Modeling can help provide knowledge not directly accessible in the real world. For instance, some models may provide a powerful even superior, substitute for reality. Theordoric of Freibourg's famous use of glass globes to simulate the role of raindrops in the formation of a rainbow show that models may provide the only possible means of studying an otherwise unresearchable process.[8]

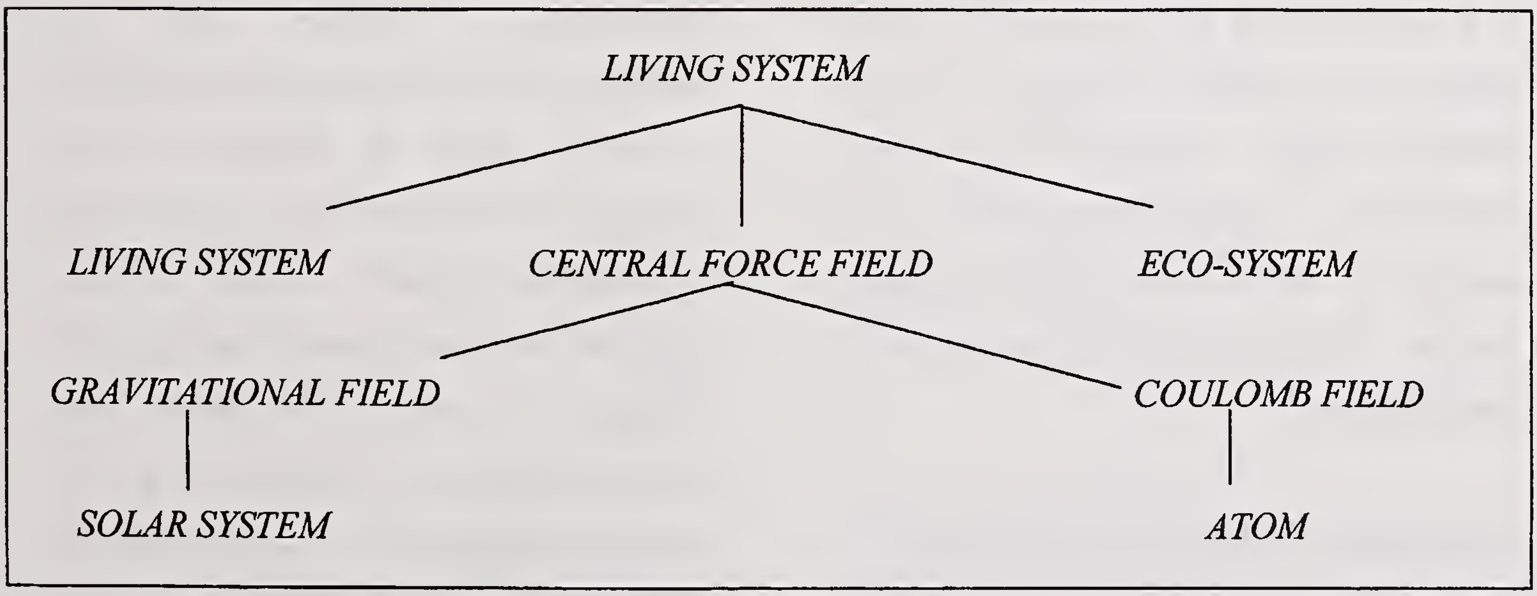
TYPE THEORY MODELING

In this short paper I will argue for a top down theory of modeling, as presented by Aronson, Harré, and Way. In this view, "theories are not thought of in terms of the hypothetico-deductive structure. Instead...theories are thought of as essentially involving chunks of type-hierarchies..."[9] If this is so, then theory-laden models already have types imbedded within the theoretic framework. Often, the type provides the direction, cohesion, and focus of the theoretical construct. So the types are already there within the theory. They simply have to be identified and used.

In the traditional comparison theory of bottom up modeling, a potential model is examined against the actual world, whether the real world is viewed logically, linguistically, or scientifically. The model functions to emulate or duplicate aspects of the real world, if not completely picture it. Because the bottom up model is not the actual world, but merely a representation, it may be locked into a deductive structure that is less elastic than the real world. This allows for avoidable difficulties in discussing possible worlds. The bottom up model also may generate counterfactuals that are known not to be true in the real world.

However, for Aronson, Harré, and Way, theories are descriptions of families of models that are metaphysical devices for expressing the ontology of our world. Our understanding of the real world is theory-laden, and therefore bottom up modeling invites comparisons which are problematic from the beginning, inherently damaged by a search for similarity that may tell us little about the actual world. Rather, they argue that the theoretic nature of our ontology must be recognized and accepted. If so, then we must look at what theories share in common. Often, a model and the system it attempts to emulate are sub-types of a larger type. The larger type is a concept that is the genesis of many ways of looking at the world. This larger type can function as a source from which hierarchies may be generated.

On this top down view, type theory becomes crucial for modeling. Type identification and analysis are prior to any comparison of models. By correctly identifying the larger type or class for examination, models are generated from the type itself. For example, if one wanted to ask if the solar system is "like an atom," one must recognize that the type under discussion is a notion of a complex system. Therefore, if a solar system is a complex system and an atom is a complex system, then the question is answered, not by comparison of the two, but through an inherited relationship that is found in any complex system. The following diagram illustrates the inheritance of relationships from the type at the top.[10]



The structure of the hierarchy generates the similarity, which is the answer to the question about the atom and the solar system. Given the view that both are complex systems, then the solar system is like an atom. The inheritance of the relationship is the vital factor in answering the question. The top down theory presents a modeling system and a system being modeled as "the lowest subtypes in a hierarchy."[11] The explaining theory incorporates them both.

Of course, the weakness of this top down view is the difficulty in identifying the proper type for discussion. The focus of modeling would shift to this issue. But the type-hierarchy model is a recognition of advances in the generation of appropriate paradigms for scientific research and a sophisticated use of modal logic.[12] The use of a type-hierarchy model can help to filter positive from negative analogies in a non-arbitrary manner. Similarity is a derived relationship. Counterfactuals based on analogy are side-stepped, thereby becoming benign. Analysis is primarily a function of classification.[13]

CROSS-DISCIPLINARY DISCOURSE

The top down theory was extensively analyzed in two conferences on cross-disciplinary discourse in 2001 and 2002. Sponsored by the Physical Science Laboratory at New Mexico State University, these conferences brought together scholars from a variety of disciplines, from literature, history of science, mathematics, biology, philosophy, robotics, computer sciences, psychology, logic, and linguistics. Each speaker discussed current issues and uses of methodology within a discipline, and then attempted to visualize cross-disciplinary applications of other methodologies. For example, Stuart Kauffman from Bios Group discussed the application of complex systems in biology and logical consistency. Dan Rothbart from George Mason University examined various uses of scientific instrumentation in the development of new methodologies. Michael Apter of Goergetown University presented his findings in reversal theory as relevant to both psychology and decision theory. Luis Arata of Quinnipiac University outlined a cross-disciplinary approach between literature and philosophy. A total of 44 papers were presented at these two conferences. A third conference will be held in January, 2003. A new journal, the *Journal of Models and*

Modeling, will showcase papers from these conferences.

Based on discussions at these conferences, there seem to be many ways to visualize cross-disciplinary modeling. One possible way to construct cross-disciplinary models is on the second-order level. This is where a top down theory could be most helpful. Consider the case of someone trying to forge a common model from sociology and physics. The search for similarity is the basis of most modeling. A category could be selected as the starting point of a top down approach, allowing for the construction of a type hierarchy. Second order levels and higher levels are accommodated by such an approach, as the hierarchy simply expands downward. On the meta-level, a top down theory demands attention to such concepts as "category", "type", "similarity", and "inheritance". The philosophical debate about these concepts will actually add to the discussion, showing new ways to find commonality or to pass down inheritance. Logic and mathematics emerge as even stronger candidates for the structure and language of models.

CONCLUSIONS

1. In a top down view of modeling horizontal analogical comparisons are eliminated.
2. Commonalities between type-hierarchies are inherited relationships.
3. The relevant focus for discussion of models becomes the shared or unshared type that generates or fails to generate two or more models.

REFERENCES

[1] Willard van Orman Quine, "Immanence and Validity," *Selected Logical Papers*, Enlarged Edition, Harvard University Press, 1995, p. 245.

[2] Bas C. van Fraassen, *The Scientific Image*, Clarendon Press, 1980, p. 44. [For an extension of this view to modal logic, *see* Graeme Forbes, *The Metaphysics of Modality*, Clarendon Press, 1985, pp. 7-8.]

[3] Rom Harré, *The Philosophies of Science*, Oxford University Press, Second Edition, 1985, pp. 173-4.

[4] *Ibid.*, p. 172.

[5] "Each possible world can be thought of as being essentially a model for the original non-modal language ..." Graeme Forbes, *Languages of Possibility*, Basil Blackwell, 1989, p. 2.

[6] *Ibid.*, p. 174.

[7] David Lewis, *Counterfactuals*, Harvard University Press, 1973, and Saul Kripke, *Naming and Necessity*, Harvard University Press, 1972, especially pp. 16-20.

[8] Rom Harré, *Great Scientific Experiments*, Oxford University Press, 1983, pp. 84-94.

[9] Jerrold L. Aronson, Rom Harré, and Eileen Way, *Realism Rescued: How Scientific Progress Is Possible*, Open Court Press, 1995, p. 88.

[10] The diagram is taken from *ibid*, p. 109.

[11] *Ibid.*, p. 110.

[12] Rom Harré, *Varieties of Realism*, Basil Blackwell, 1986, pp. 317-343.

[13] Bertrand Russell, Mathematical Logic as Based on a Theory of Types," *American Journal of Mathematics*, 1908, Volume 30, pp. 222-262, and Alfred North Whitehead and Bertrand Russell, *Principia Mathematica*, Volume I, Second Edition, Cambridge University Press, 1927, pp. 37-65. *Also, see* my "Ryle's Debt to Russell," *Linacre Journal*, 1999, Number 3, pp. 55-69.

Performance Evaluation of Network Centric Warfare Oriented Intelligent Systems

Edward Dawidowicz, *Member, IEEE*

Abstract

The concepts of Network Centric Warfare [Alberts et. al 1999] and its sibling Knowledge[1] Centric Warfare are critical elements in achieving so-called Information Superiority. Both of these concepts are not limited to military applications only, but are also suitable in the areas of business or daily life. For the latter however, we should remove the term "warfare" to suggest more appealing applications. The Knowledge Centric aspect is critical in achieving effective Information Superiority *"To transfer knowledge, the receiver's context and experience must be taken into account. The intended result is information is transferred in context instead of with no context.*[Harris, D.B. 1996]

The main question remains not only what Network Centric (NC) and Knowledge Centric (KC) are but also how these concepts can effectively be used to pragmatically achieve Information Superiority. The purpose of this paper is to discuss the NC and KC aspects including network configuration, functions of different nodes of the network, the intelligence required to facilitate KC by providing contextual information dissemination. The discussion of the key infrastructure elements will provide the foundation for exploring the performance evaluation of NCW oriented intelligent systems.

The warfighter desires the 'right' information at the 'right' time. Such information can be defined as contextual. The solution for contextual information dissemination requires intelligent information processing within the nodes of the communication network. The architecture required to support such intelligent nodes is described in this paper.

Edward Dawidowicz U.S. Army Communications-Electronic Command (CECOM), Research, Development and Engineering Center (RDEC) Fort Monmouth, NJ Tel (732) 427 - 4122, edward.dawidowicz@mail1.monmouth.army.mil

[1] Knowledge 1 obsolete: COGNIZANCE 2 a (1) : the fact or condition of knowing something with familiarity gained through experience or association (2) : acquaintance with or understanding of a science, art, or technique b (1) : the fact or condition of being aware of something (2) : the range of one's information or understanding <answered to the best of my knowledge> c : the circumstance or condition of apprehending truth or fact through reasoning : COGNITION d : the fact or condition of having information or of being learned <a man of unusual knowledge>. From Merriam Webster's Dictionary on line http://www.webster.com/cgi-bin/dictionary

I. Introduction

The definition of the problem space must be declared before evolving a solution to a particular problem within the scope of Knowledge Centric Warfare. For the purpose of this discussion the problem space can be decomposed into four main components:

1. **The battlespace** – the topology of the physical space where the action is taking place, the physical laws, the involved equipment and the entities' physical attributes
2. **The doctrine**, rules of engagement, and policies,
3. **The communication networks** – where information to support coordination of effort and execution of moves is transported,
4. And finally, **contextual information packaging and dissemination.**

A. The World, Battlespace, and Battlespace Decomposition

The battlespace is a model consisting of the geography of the region, the position and capability of friendly, neutral and opposing units or entities. The entities are expressed as sets of physical and cognizant properties including models of maneuver, tactics, and combat capability. Based on physical and cognizant properties and commander's goals, these entities may assume either combat or combat support postures. These entities are the players within the battlespace. The battlespace problem is a collection of issues, which the players must overcome to achieve mission successes or to win a war.

The battlespace is partitioned into domains. The domains are decomposed to reflect functional responsibility of a particular entity. The entities responsible for these domains are dispersed throughout the battlespace and have a need to communicate and collaborate. The battlefield problem space is complex and subject to constant change due to various factors such as weather, new threats, new tasks, and unavailability of planned resources. These entities need an information environment, which facilitates a capability for dynamic configuration/reconfiguration in order to meet their need to rapidly form different mission–specific teams, to be aware of their changing environment, and to have contextually pertinent information temporally reflecting the fluidity of the battlespace.

B. Network Centric and Knowledge Centric

Metcalfe's Law[2] suggests the power of information dissemination contained within a fully connected network, however it says nothing about the quality and contextual relevance of the information such network can provide. This power manifests itself in the large amount of potentially available information accessible at the nodes of a network. The question we must ask ourselves is what is more desirable, a large volume of information, what ever it might be, or a short but contextually relevant extraction from that large volume.

Large volumes of redundant or irrelevant information will overburden the communication channel rendering the NC aspect less effective or useless. Prioritizing and disseminating information based on the need to know and as recipient's task critical requirement can further save the communication bandwidth. Determining information pertinence and packaging the information within a specific level of granularity, required by the recipients, becomes therefore paramount in implementing the paradigm.

To analyze the NCW and KCW approaches we have to consider current and evolving topological architectures of tactical networks. However, the topology of the network is a "parcel delivery infrastructure" and while it erroneously seems to have no bearing on the actual context it is important for multilevel modeling. The success of KCW specifically depends on the contextual information dissemination. To achieve contextual information dissemination requires intelligent information processing at every node of the network, except routers or similar functioning devices, where information is received and sent.

C. Communication Network of the Battlespace

Shown below in Figure 1 are representations of possible network configurations. Fig.(b) is best suited to depict a typical military network, which represents for example, communication between ground force companies, battalions, or navy ships at sea. The hubs of the network, shaded gray in Figure 1 b, may also represent unit clusters consisting purely of sensors, robots, and people or a heterogeneous composition. For example, an M1A1 tank can be viewed as a hybrid of sensors, weapons, and people and can also represent one node in an armor company network.

[2] Metcalfe's Law, which states that the usefulness, or utility, of a network equals the square of the number of users. Named after Robert Metcalfe, the founder of 3Com Corporation and designer the Ethernet protocol.

The NC paradigm suggests the topology of Figure 1 (c), however such topology is very difficult to achieve for several reasons;

- Unavailability of required electromagnetic bandwidth,
- Line of sight limitations
- Doctrinal, echelon dependent communication requirements.

The topology of a network for brigade and below is shown in Figure 2. Additional battalions were omitted for simplicity.

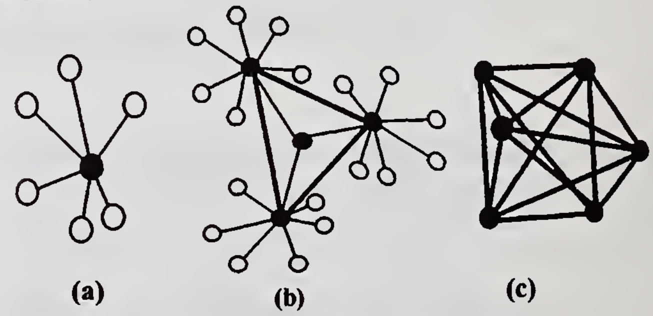


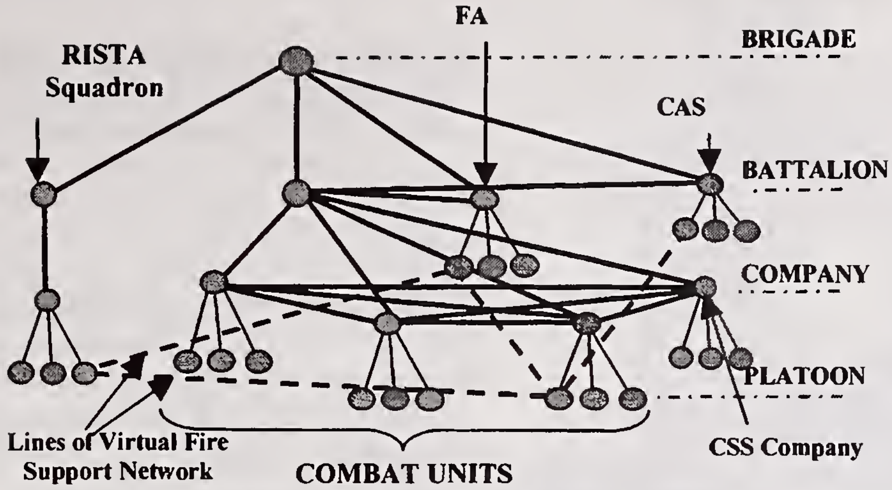
Figure 1. Network configurations
(a) Simple star, (b) Cluster of stars,
(c) Fully connected

The topology of Figure 2 lacks connectivity between battalions and companies of adjacent brigades. The elements of battalions are highly mobile and frequently come within weapons range of each other and must be aware of each other presence to avoid fratricide. The problem is further exacerbated when these elements also belong to different brigades. The situation awareness information, of units belonging to this brigade, must travel up to the level of the first brigade, must later be transmitted to the second brigade, and finally must be disseminated to the lower echelons. Whether the network topology remains the same or changes, **the need for intelligent processing at the nodes is critical to contextually evaluate the information about who done what and who needs to know about that first.**

D. Knowledge Centric Network

Understanding the information requirements for individual recipients is essential to achieve effective contextual information dissemination within the KC network. It is outside the scope of this paper to explore all the requirements for all potential individual recipients on the battlefield, however a general architecture must be defined. In order to be effective, the architecture must answer the following questions:

1. **What is the echelon of the recipient**
2. **What duties does the recipient have at a specific instance of time**
3. **What is the state of battlefield variables**
4. **What information must be sent first**



Reconnaissance Intelligence Surveillance and Targeting Acquisition (RISTA)
Combat Service Support (CSS)
Close Air Support (CAS)
Field Artillery (FA)

Figure 2. Communications network for brigade and below

5. **What is the level of granularity of the information required**
6. **When must the information be sent**
7. **What does the recipient already knows**

The major elements of the KC architecture are based on knowledge about the area of responsibility or the duties and tasks assigned and the echelon level of the individual. Such profiling is doctrinally driven and available in field manuals. The content of the information set is modeled on those attributes. The required information profile is not a template, or a table to be filled out to meet the information requirements, but is a mapping function, which transforms raw information and data into the information requirements for individual recipients (Figure 3).

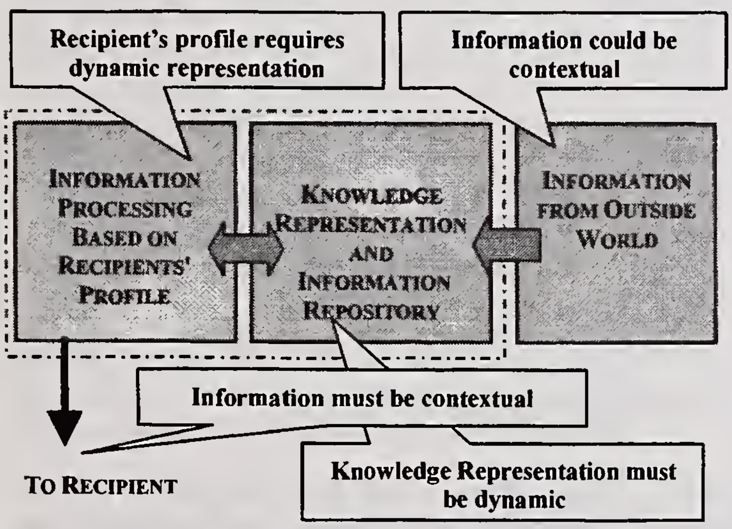


Figure 3. Simple Information Processing based on Knowledge Representation

The information profile contains attributes to answer questions such as **what echelon is the recipient, what duties does the recipient have at that instance in time, where is the recipient, what information must be sent,** and **what is the level of granularity.** To answer the question **what is the state of battlefield variables** requires an updated world model, a multilevel knowledge representation of the environment. The multilevel knowledge representation of the environment will provide the required inference to answer the question of **when to send the information.** The question **what does the recipient already know** can be answered by maintaining a repository of previous transactions local to the information source.

II. Intelligent Node Architecture

Intelligent agent architecture, defined in earlier work [Dawidowicz E, 1999], is also applicable to the intelligent node architecture, but requires modification and improvement to qualify as an intelligent node described here. The improvement is required specifically in the area of adaptation of the intelligent node to the changing battlespace environment. A likely candidate for such improvement is the application of an intelligent controller as described in semiotic modeling [Meystel A, 1995]. This model is applicable to both individual intelligent nodes As well as to a cluster or clusters of collaborating intelligent nodes. The analogy to intelligent automatic control is evident and emphasized.

The think-before-act or the actuation simulation loop is the foundation of the proposed architecture and is shown in Figure 4. The Elementary Loop of Functioning is a goal driven process. Before selecting a possible response for a specific goal it generates, using the World Model, several potential actions (this is not a complete sentence). The best- actions are selected and used to stimulate the simulated world (or environment). The simulated sensory response is collected, processed and fed back into the world model. This constitutes the contemplation of think-before-leap process and is analogous to imagination.

A. Knowledge Representation Repository

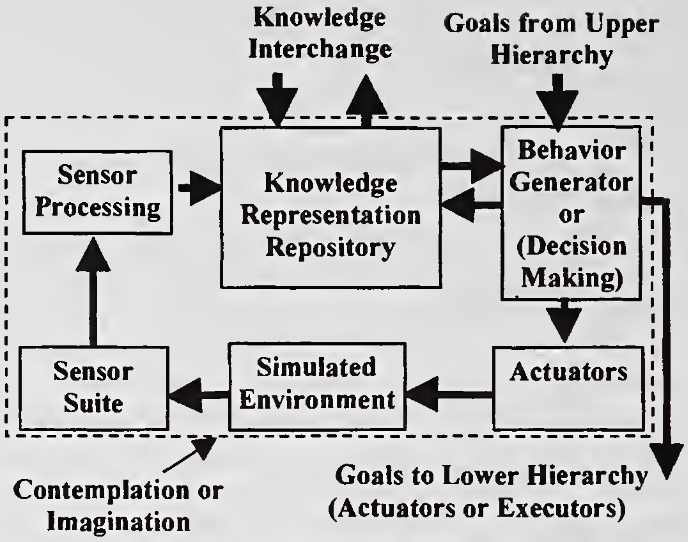


Figure 4. The Elementary Loop of Functioning (ELF)

The Knowledge Representation Repository (KRR) in general, is a description of the world. The KRR contains the model of the anticipated and learned environment or the battlespace. Specifically KRR[3] is a set consisting of, but not limited to models of:

a) Representations of terrain, in the sphere of interest, with elevation data and features,
b) Physical geographical data of the terrain such as soil properties, water levels, variations due to tide or precipitation,
c) Physical objects that are known to appear in that environment,
d) Object properties,
e) Objects which were detected in the environment,
f) Geo-spatial location of the physical objects,
g) Associative relationships between objects,
h) Rules and procedures associated with certain conditions of relevant battlespace,
i) Specific activities the objects which are in the modeled environment,
j) Meteorological data,
k) Profiles and information requirements of the users,
l) Ontology for textual discourse

The KRR is both, a process and a repository of information subject to a phenomenon called reflection [Meystel A. 1995, p68]. The KRR will contain knowledge extracted from doctrine, pollicies, operational requirements, mission plans, maps, map features, equipment capability, and situational awareness.

The KRR is updated by exchange of information between KRRs on the network. The rules of information exchange depend on the geographic proximity between the nodes and their functional interdependence. The rules within the KRR are also updated using the Elementary Loop of Functioning process discussed later and in [Meystel A. 1995, p67].

To be valuable within the KCW paradigm the KRR must contain the representations of the information interchange on at least three different levels; on its own level, on an equivalent level of functionally equal or functionally different, and on one level above and one level below. These levels are synonymous with echelons, while the functionality is derived from the service these echelons are expected to perform and are critical in heterogeneous KCW. For example this diversity in functional representation will be instrumental in determining the context of the message interchange, in close air support mission, between the Army and Marine warfighters on the ground and the Navy and Air Force pilots who provide the air support to them.

[3] The modeling properties reflect a specific KRR level of representation and hence employ a particular resolution or granularity appropriate to such level.

B. Decision Making

The Decision-making process (DM) is initiated by a goal, either given by a decision-maker from a level above or in response to critical changes detected within the KRR. The detected changes within the KR become critical when the DM can detect or anticipate possible deviations from the plan. The goal of the DM is to provide tasking to the external actuators to correct the deviation from the plan under execution.

The DM within the intelligent node compares a current situational picture to the picture anticipated based on a plan in execution. The DM also prioritizes, required to be performed tasks, based on a particular situation, or a particular set of states. The rules of KRR are used to determine the priority of a particular task. The prioritization can be illustrated in a scenario when a particular intelligent node is involved in a CAS mission and the planes are a few minutes from delivering their munitions on the enemy positions. The first priority of that particular node is to prevent a potential fratricide situation, by providing the pilots with the latest positions of the friendly forces in the proximity of the anticipated kill zone. The second priority is to notify the pilots of where the enemy is. However, when an enemy antiaircraft threat is detected, an intelligent node must make the threat notification to the pilots first and then provide CAS critical information.

C. Elementary Loop of Functioning

The DM is more complex than a typical follow-the-rules process. It can 'reason' by invoking the Elementary Loop of Functioning (ELF) [Messina E, Meystel A. 2000] Figure 5[4]. By using the information in KRR it forms a hypothesis as to what needs to be done. To test the hypothesis a

[4] Please note that Figure 5 is significantly different from Figure4. The significant different is in another ELF which runs from DM and another ELF within KRR. This architecture allows the intelligent nodes to "correct" its models on different levels of resolution based on knowledge representation shared and received.

command or a set of commands is sent to the Actuator block. The Actuator block is a set of simulated actuators or a set of processes expected to simulate task actuation.

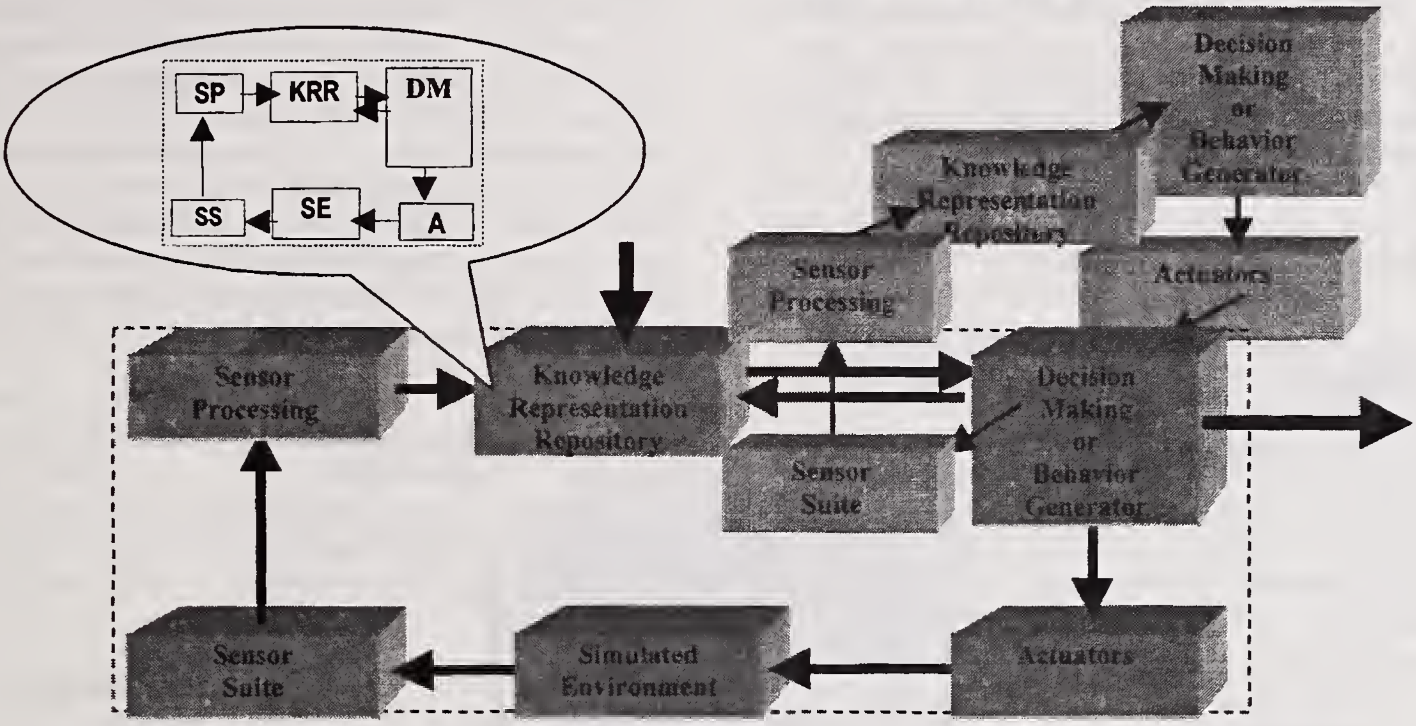


Figure 5. Elementary Loop of Functioning with multi-resolution ELFs

D. Simulated Environment

The simulated environment (SE) is a subset of KRR. Only the elements of KRR, pertinent to the immediate domain within which the simulation is to occur, are incorporated in the simulated environment. The simulated actuators are activated within the SE. The Sensors Suite (SS) detects the resulting changes, from actuation, within the environment caused by the simulators.

E. Sensory Processing

Sensory Processing (SP) processes the changes in the SE, detected by the SS. The SP block fuses and correlates information as it would to in the real environment. The processed sensory information is sent to the KRR.

F. Completing Contemplation Loop

The results of the simulation are compared to expected values. When the simulated results are acceptable the DM will perform a required action by sending an appropriate message to the outside world, or to another node on the network. Please note that during all processes within the large ELF, smaller ELF process run within the larger loop elements. The number of nested loops depends on the required level of granularity or resolution for a particular contemplation cycle. Usually one level above and one level below are sufficient, but rarely may require several levels down. The execution of different levels of ELFs, within each individual block, is dictated by a requirement for higher or lower granularity models. The DM, KRR, and SE blocks specifically require ***multi-resolution*** modeling.

III. Intelligent node as an Intelligent Controller

The intelligent node is an intelligent controller, which continually adapts itself to the environment. If allowed, it initiates situational awareness information exchange between other intelligent nodes based on established relations. The relations are determined by homogeneous or heterogeneous combat cells, which are formed into task/mission teams. Such teams can also be called habitats. The habitats are not bound to a single geography, they may be globally distributed, and can consisting of humans, intelligent agents and robots.

The purpose of the intelligent node, in the KCW intent, is to contextually process and disseminate information. To achieve the KC aspect, the intelligent node should have the knowledge representation of the receiving node. This does not mean that that it must contain all of the KRR of the receiving node, but the knowledge representation must be sufficient to formulate a contextual message. The contextual message must be formulated, prioritized and timely sent to the receiver containing only the information required.

The formulation of messages and informational content is based on the need to know and the security level of the receiver. Both the need to know and the security levels are based on doctrine, policies and plans.

The ELF modeling of the intelligent node is not limited to KC information exchange. Such modeling is an invaluable tool for mission planning, mission execution, and replanning. The intelligent nodes also serve as a useful asset in filling the Critical Commander's Information Requirements (CCIR) and Priority Intelligence Requirements (PIR).

A. Intelligent Node in Two Echelons

The ELF model supports the information flow pattern of a military organization. Figure 6 represents instances of a battalion and three subordinate companies or brigade and three subordinate battalions and depicts the purpose of the individual components.

consideration of both individual components and a system of such components.

The performance evaluation of individual intelligent nodes must reflect the echelon levels they are modeled to represent. Since events evolve faster at the lower echelons, the intelligent nodes must evaluate information proportionately faster. This is reasonable since lower echelons are near term planners and are concerned with the more immediate future. In general, the granularity of information is finer at the lower levels, but requires shorter term planning. The criteria for performance evaluation therefore cannot be applied equally to a node, but must reflect the echelon and functional purpose such an intelligent node serves in the KC network.

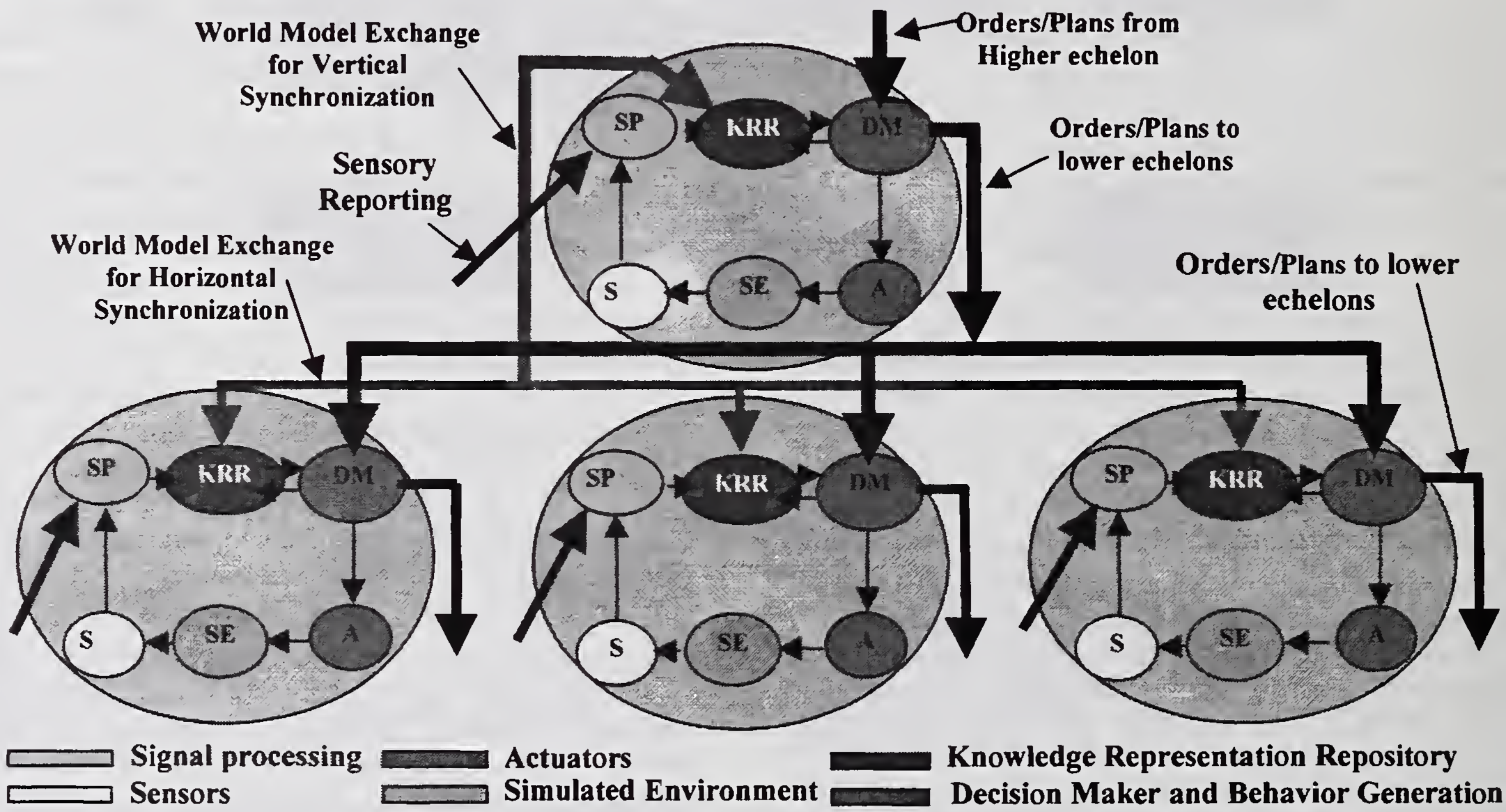


Figure 6. Information exchange between command and three subordinate units

IV. Performance Evaluation

Before discussing performance evaluation, Measures of Effectiveness (MOE) and Measures of Performance (MOP) must be point out. The MOE and MOP are important abstractions used for system evaluation [Noel Sproles, 2001]. The MOE provides the formulation of purpose or need, while the MOP refers to the performance of a particular entity developed to fill that need. The system of Intelligent Nodes responds to the MOE: 'Ability to provide task pertinent and concise information to the user'. The definition of MOP is more complex and requires

The Intelligent Nodes are but elements in a system where the value of the system is greater then the sum of its parts. The evaluation criteria are therefore not scalable from individual components to the system. The architectural framework together with the performance requirements provides the basis for evaluation. Below are listed some architectural and performance requirements.

A. Architectural Requirements of Intelligent Nodes

1) Completeness of the Knowledge Representation of the battlespace reflecting a specific level of granularity. The Knowledge Representation model must reflect specific echelon and functional levels
2) Ability to adapt the Knowledge Representation model to changing and evolving battlespace

3) Develop Decision Generator/Behavior Generator capable
   a) of dealing with incomplete and uncertain world representation models,
   b) developing hypothesis or a set of assumptions to resolve uncertainty,
   c) to simulate the hypothesis/action,
   d) to evaluate the results of simulation,
   e) and finally to select the "best" result as a decision/action.
   f) to enrich the Knowledge Representation Repository with a new "rule" if a particular hypothesis yields a better solution.
4) Develop a process, identifying the important elements to process
5) Ability to dynamically prioritize tasks to reflect the current situation
6) Natural language or controlled natural language understanding.
7) Ability to express reasoning using natural language
8) Ability to share knowledge representation among other Intelligent Nodes

B. Performance requirements

1. The Intelligent Nodes must be evaluated based on their specific echelon and functional levels.
2. The lower the echelon, the greater the requirement for faster processing.
3. The speed of processing must be examined against the methodology used in information processing.
   a. Number of possible permutations / hypothesis resulting from evaluating the environment and the actions/goals of the entities involved.
   b. Optimal selection of the best permutations
   c. Formulation of hypothesis and ability to evaluate them for optimum results.
4. Number of granularity levels of Knowledge Representation used in the hypothesis evaluation process

Discussion and Conclusion

The performance evaluation of Intelligent System is a difficult process. It is especially difficult since the definition of intelligence remains largely elusive. Perhaps the issue is not what intelligence is, but rather how it must assist in resolving an unspecified problem. Digital computers have their limitation " Might it be that the symbol grounding problem is created by the digital computer rather than solved by it? Perhaps the idea of abstract information or symbols is a computer-based fiction?" [Hoffmeyer J, 1997]. The purpose of an Intelligent Node based system is not to model intelligence in its pure sense, but to produce a pragmatic tool to assist in dealing with the information explosion.

The tale of a few blind men and their encounter with an elephant comes to mind. They were allowed to touch the animal to learn what it was. After examination they shared their findings and learned that the animal is a huge barrel standing on four pillars with a large hose in the front and a dust sweeper or fly swatter in the rear.

A system with a single layer of resolution may just produce the same view of the world as that of the elephant perceived by the proverbial blind men. If the blind men could go beyond the single resolution in their verbal description and were able to share among themselves their tactile findings in several levels of resolution, then their perception of the animal would appear closer to the truth.

The Intelligent Nodes described here are analogous to our proverbial blind men, but only in the ability to share information that they sense. When modeling described here is implemented, the discourse among the Intelligent Nodes will be much richer, for they will be able to share information with a sufficient complexity, however not in bulk, but in context. By sharing contextual information they as a system will arrive at a better understanding of their world.

References

1. [Alberts, D et. al. 1999], Alberts, David S., John J. Garstka and Frederick P. Stein. "Network Centric Warfare" - 2nd Edition. September 1999. PDF version http://www.dodccrp.org
2. [Dawidowicz E, 1999], Dawidowicz E, " Intelligent Agent Technology in Command and Control Environment", Proceedings of 1999 Command and Control Research and Technology Symposium, Newport, Rhode Island, June 29 -July 1 1999.
3. [Harris D.B., 1996], David B. Harris, "Creating a Knowledge Centric Information Technology Environment", Seattle, WA, September 15, 1996 http://www.dbharris.com/ckc.htm
4. [Hoffmeyer J, 1997] Jesper Hoffmeyer, 'Semiotic Emergence', Revue de la penseé d'aujourd'hui, vol. 25-7, No 6, pp.105-117 (in Japanese) 1997. English version: http://www.molbio.ku.dk/MolBioPages/abk/PersonalPages/Jesper/SemioEmergence.html
5. [Messina E, Meystel A. 2000]. E. Messina, A. Meystel, “The Challenge of Intelligent Control,” Proceedings of the IEEE Int'l Symposium on Intelligent Control, Patras, Greece, 2000
6. [Meystel A, 1995], "Semiotic Modeling and Situational Analysis an Introduction", AdRem Inc. 1995.
7. [Sproles N, 2001], "A System Approach to Establish Effectiveness for Command and Control", the 6-th ICCRTS, June 19, 2001, http://www.dodccrp.org/6thICCRTS/Cd/Tracks/T6.htm

PART III
RESEARCH PAPERS

2. Testing for Performance Evaluation

2.1 Designing Metrics for Comparing the Performance of Robotic Systems in Robot Competitons
H. Yanco, USA

2.2 Experiences in Deploying Test Arenas for Autonomous Mobile Robots
A. Jacoff, E. Messina, J. Evans, NIST, USA

2.3 Hierarchical Architecture for Coordinating Ground Vehicles in Unstructured Environments
A. Lacaze, NIST, USA

2.4 Evaluating Knowledge and Representation for Intelligent Control
E. Messina, J. Evans, J. Albus, NIST, USA

2.5 Evaluating the Performance of E-coli with Genetic Learning from Simulated Testing
A. Meystel, Drexel, Univ., USA

Designing Metrics for Comparing the Performance of Robotic Systems in Robot Competitions

Holly A. Yanco, *Member, IEEE*

Abstract – Many robotics competitions have been held over the past decade. These competitions often have the stated or unstated goal of comparing different robotic systems and their research approaches. When designing the rules for a competition, there are several ways to compare the performance of robots: objectively, subjectively, or a mix of the two. This paper discusses several robot competitions that have been held and how the metrics for judging performance were designed.

I. Introduction

Robot competitions bring together a group of people interested in a particular problem to demonstrate and discuss ways to accomplish a given task. Competitions often influence the direction of research in robotics, which can be used to great advantage. Indoor navigation is considered by many to be a solved task now, and this accomplishment was driven by several years of office navigation competitions in the AAAI Robot Competition and Exhibition. The latest additions to the AAAI contest are Robot Challenge and Robot Rescue, both of which include many hard research problems. Despite these good examples, when designing a robot competition that will compare research institutions, it is important to consider that a particular competition could drive research for several years.

Rules for robot competitions can take one of three forms: a ranked competition with subjective scoring, a ranked competition with "objective"[1] scoring, and a non-ranked competition with technical awards. A subjectively ranked competition should have clearly stated areas that will be judged and suggest guidelines for the judging. An objectively scored competition should have easily quantifiable metrics (e.g., number of objects found or amount of time taken to accomplish the goal). A non-ranked competition allows for more flexibility in the design of rules, since the lack of rankings will prevent any contentions that might arise in a ranked competition.

Competition metrics can be useful to compare research approaches. However, it is often very difficult to directly compare different solutions to the same problem. For example, at the Robot Rescue competition in 2001, one entry had treads and was teleoperated, while another had wheels and AI control software. In this case, task completion is used as a metric, rather than judging the methods used to accomplish the goal.

Competitions may be head-to-head or have each competitor run separately in the competition arena. The advantage of a head-to-head competition is that it is much more exciting for spectators, as they can root for one team over another. However, individual runs can be much easier for judges to watch and score, especially when the task is not one that easily lends itself to head-to-head competition.

II. History of the AAAI and RoboCup Competitions

In 1992, the first annual AAAI Robot Competition and Exhibition was held in San Jose, California. The introduction of this event marked the first AI robot competition and brought together many of the major robotics research laboratories and universities. This inaugural year introduced a competition involving navigation and identification of locations marked with encoded poles. Navigation continued to be a major component of the competition for several years, with office navigation as the primary focus. At the time of these early competitions, indoor navigation for mobile robots benefited greatly from the intense work in the area; the competition drove research forward.

Computer Science Department, University of Massachusetts Lowell, One Univeristy Avenue, Olsen Hall, Lowell, MA 01854.

[1] Many "objective" scoring methods involve some amount of decision that must be made by the judges, which introduces some subjectivity.

The AAAI Robot Competition has evolved over its ten years to include several other contests, each with different research aspects. *Find the Remote* was an event at AAAI-97 where a vision system was necessary in order to locate specified objects. *Life on Mars* was another competition that encouraged the use of computer vision; competitors needed to find colored "aliens" in a field of black boulders, then put the "aliens" into a "lander" with a colored door. The *Hors d'Oeurvres Anyone?* competition, introduced in 1997, encouraged the development of systems with good human-robot interaction, by creating robot servers that would both bring food to people while trying to entertain or interact with people. The *Robot Challenge* was first held at AAAI-99; the goal of this event is to have a robot register for the conference and give a talk about itself at an appointed time, after being dropped off at the entrance to the conference hall. In 2001, the *Robot Rescue* event was added, bringing an urban search and rescue scenario to the AAAI Competition.

Another robot competition, RoboCup, started in 1997. The goal of RoboCup is to have robots playing soccer with humans by the year 2050. The first five years have encouraged research in this direction by having several robot leagues, each of which encourage the development of different aspects of the research problem. In the small league, a camera placed above the arena allows for off-board vision processing. Larger robots have on-board cameras. The Sony dog league encourages research in legged locomotion for soccer, and the humanoid league is promoting the development of human-like robots, although there have not been any humanoid league soccer games at this early date. In 2001, RoboCup added a Robot Rescue league, held in conjunction with AAAI-2002. RoboCup also has simulation leagues for both soccer and rescue.

III. Designing Competitions and Metrics for Judging Performance

When designing any competition, the organizers must carefully consider the rules and scoring. The rules and scoring are often points of contention, so care must be taken to avoid skewing the algorithm towards any single research approach or robot base. Additionally, it is desirable to create a set of rules that are broad enough to encourage many different approaches, as this is likely to advance the state of the art more quickly.

Competitions fall into three categories:

1. Ranked competitions using subjective scoring based upon pre-specified criteria. The AAAI *Hors d'Oeuvres Anyone?* event is an example of this scoring method.
2. Ranked competitions using objective scoring using carefully spelled out criteria. The AAAI/RoboCup *Robot Rescue* event is an example of this scoring method.
3. Non-ranked competitions with technical awards. The AAAI *Robot Challenge* is an example of this type of competition.

A. The AAAI Hors d'Oeuvres Anyone? Event

The AAAI *Hors d'Oeuvres Anyone?* event was first held at AAAI-97 and has been an event in all of the subsequent AAAI Robot Competitions. The task of the *Hors d'Oeuvres Anyone?* competition is to serve hors d'oeuvres to people in a crowded reception. Robot servers should cover the entire space, in a attempt to serve as many people as possible. Entries may consist of a single robot or a team of robots.

The competition encourages human-robot interaction beyond driving food on a tray to people. In the first competition in 1997, one robot showed movie clips while serving food. Another team included a performance with their trio of servers, acting out a "Robotic Love Triangle." Almost all of the teams outfit their robots for the event, from masks to signs to butler uniforms. Some robots tell jokes when serving, while others try to greet people by name, using computer vision to locate a conference badge, extract the name region, perform character recognition, and then speak the result. Some of the years have provided bonus points for robots that could recognize VIPs by the color of the ribbons hanging from their conference badges.

Robots are also rewarded for recognizing that they need to reload their tray, either by counting the number of people served, by measuring the weight of the tray, or by using a computer vision system to judge when the tray is empty. Once the robot has determined that it needs more food (or a human attendant has made that decision for a robot unable to make its own determination), it should be able to guide itself back to a food reloading station. At this station, a human attendant reloads the food. While it would be desirable to have a robot reload its own food, there will need to be additional research into manipulators for mobile platforms.

When designing rules for competitions, it is important to consider the different robotic bases that researchers have in their labs. In this particular competition, the floors are flat and regular, allowing the majority of labs with wheeled bases to compete. The problem with many of the robot bases currently in use is that they are too short to interact effectively with people. To solve this problem, teams build structures on top of their robots to increase the robot's height to a person's waist height. Speech is also an important ability for robots in this competition; fortunately, relatively inexpensive systems are available to generate speech from text.

The robots are ranked using subjective scoring. In the 2001 competition, event judges awarded a subjective score of 1 to 10 in the following categories: ability to serve food, interaction with humans, interaction with other contestants, manipulation and sensing modes. To produce the final rankings for the event, the rankings determined by the event judges are combined with a popular vote. During the event, each attendee is given a token which is to be placed in the box of his/her favorite server. After the conclusion of the serving period, the votes are tallied and combined with the judges' scores to produce the rankings for the competition.

The metrics for determining the winner of this competition thus may have two disparate results: the crowd pleaser may not be the best technical entry. When designing a competition with metrics for technical judging and for popular voting, one should consider whether the two parts should have equal weight or if the technical aspects should outweigh the votes of non-roboticists. In the case of robotic servers, effective interaction with its audience is very important; a very technically-advanced entry that acts like a rude waiter may not be the best entry.

This competition is intended to serve as an entry level competition at AAAI. Undergraduate teams can be as successful as teams consisting of more advance robotics researchers. Additionally, the robot platforms can vary without too much of an effect on a team's competitiveness.

B. *The AAAI/RoboCup Robot Rescue Event*

In the *Robot Rescue* competition, the goal is to find victims in a collapsed building, which is represented by the Rescue Arena designed and built by the National Institute of Standards and Technology (NIST). The robots must report the location of victims to operators outside the arena. Entries may consist of a single robot or a multi-robot team.

The NIST designed rescue course has three areas: yellow, orange and red. In the yellow area, there are even floors, allowing wheeled bases to be used in the competition. The orange area has ramps and stairs with some rubble on the floor. The red area is the most difficult, with narrow collapsed areas and large amounts of rubble.

The differences in hardware and research approaches are more pronounced in this competition than in the *Hors d'Oeuvres Anyone?* competition, since two of the arena's areas are impassable to wheeled robots. In the 2001 competition, one team's entry was a custom built tracked robot that was teleoperated (future plans include the inclusion of AI software). Another entry used commercially available wheeled bases with custom AI software to navigate and locate victims. The wheels on the second team's entry precluded them from entering the orange or red areas. Since more points are earned for victims found in the more difficult areas, it is more difficult for a wheeled team to rank above an all-terrain team.

The Robot Rescue event debuted at AAAI in 2000. In 2001, the competition was held jointly at the co-located IJCAI-2001 and RoboCup-2001 conferences. At AAAI-2000, teleoperation was not allowed, as the focus of the AAAI competitions is the development of the algorithms. However, the inclusion of the RoboCup community, which includes many roboticists on the mechanical engineering side, warranted a change to this rule. The focus shifted from judging how the robot performed its task to how well it performed its task. A joint rules committee consisting of AAAI and RoboCup representatives designed the rules for the 2001 competition.

The rules of the competition focused on the desired outcome in a real search and rescue situation. It is important to be able to find all of the victims quickly and to report their locations to people outside the building. The reported locations should be accurate, and it is best if the robots are able to generate a map that would allow human rescuers to find the victims quickly. In a real rescue situation, it is better to have fewer human operators required for a robot, since there are restrictions on who can enter the "warm zone" around a disaster area.

The joint rules committee identified several variables to be used in judging the competition. All were spelled out carefully, resulting in an objective scoring algorithm.

The variables for the scoring algorithm are as follows:

- N is a weighted sum of the number of victims found in each region divided by the number of actual victims in each region.

- C_i is a weighting factor to account for the difficulty level of each section of the arena: $C_{yellow} = .5$, $C_{orange} = .75$, and $C_{red} = 1.0$.
- N_r is number of robots that find unique victims.
- N_o is the number of operators.
- A is an accuracy measurement for the location of each victim: A = F/V. F is equal to 1 if the victim is in the reported volume, and 0 otherwise. V is the volume in which the reported victim is located, given by the operator in the warm zone to the judge. The average accuracy is used in the scoring algorithm.

Each team ran for twenty five minutes; the best two scores from four runs were used to determine the final score. The algorithm for determining the score of a round is as follows:

$$Score = N * \frac{N_r}{(1+N_o)^3} * \overline{A},$$

where

$$N = \sum_{i=\{yellow,orange,red\}} \frac{C_i * N_{victimsDetectedS_i}}{N_{actualVictimsS_i}}$$

In order to receive a ranking in the competition, the competitors needed to meet a minimum score requirement, which was equivalent to finding all of the victims in the yellow zone. No competitor earned the minimum score in 2001, although two teams were close. Instead of rankings, two technical awards were presented by the judges, one which rewarded the development of mobility for rescue and the other which rewarded the development of AI algorithms for rescue.

C. *The AAAI Robot Challenge*

The task of the AAAI *Robot Challenge* is to have a robot attend the National Conference on Artificial Intelligence. The event is started when a robot is dropped off at the entrance to the conference center. The robot needs to find the registration desk for the conference, which it may do by asking people for directions and assistance. After registering, the robot needs to find a specified conference room and give a talk about itself at a specified time.

The event is very challenging for the robotics field and includes many open research problems. The intent of the event is to encourage senior robotics researchers and graduate students to bring their work to AAAI. Since there are many areas of research involved in this problem, it would be difficult to rank the competition entrants. Instead of rankings, judges may give technical awards. Examples of possible awards are innovation in localization and navigation, innovation in robot vision or sensor technology, innovation in human-robot interaction, innovation in real-time planning, innovation in manipulation, and excellence in collaboration and integration. The advantage of a non-ranked competition is also that people may be more willing to demonstrate work in progress, resulting in additional communication between researchers.

IV. Conclusions

When designing performance metrics for competition, a rules committee must decide what is important. Task completion may be the most important goal, as it is in the *Robot Rescue* competition; it may not be important how a victim is found, as long as the person can be rescued. Other competitions may choose to allow partial completion of the specified task, judging instead a demonstration of good research and/or intelligence. Some of the aspects of the *Hors d'Oeuvres Anyone?* rules include this approach. The initial stages of the *Robot Challenge* also reward partial completion, although the ultimate goal is task completion.

A competition must also decide whether it aims to showcase new research or systems that are ready for deployment. In the case of the *Robot Rescue* event, wheeled robots may be used to demonstrate new algorithmic capabilities, but can not score as highly as a tracked robot in the more difficult areas. In contrast, the *Robot Challenge* allows new research to be showcased and eliminates most of the performance pressure with the removal of rankings.

All of these approaches have valid purposes. When designing a new competition and set of rules, determining the desired outcomes of the event should be the first task. This step will help to determine whether the scoring should be objective or subjective. The next step should be designing rules that can include multiple robot bases and research approaches. Whatever the design, the rules should be clearly spelled out and available as far in advance of the competition as possible.

Experiences in Deploying Test Arenas for Autonomous Mobile Robots

Adam Jacoff, Elena Messina, John Evans
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899

ABSTRACT

The National Institute of Standards and Technology has created a set of reference test arenas for evaluating the performance of mobile autonomous robots performing urban search and rescue tasks. The arenas are intended to help accelerate the robotic research community's advancement of mobile robot capabilities. The arenas have been deployed in two competitions thus far and are also being used by researchers to test their systems' capabilities. We describe the arenas, their use in competitions and our near-term and long-term plans for the arenas.

1. INTRODUCTION

The National Institute of Standards and Technology (NIST) has been collaborating with other government agencies and university researchers to develop methods of evaluating and measuring the performance of robotic and other intelligent systems. The community agrees that it would benefit from having uniform, reproducible means of measuring capabilities of their systems to evaluate which approaches are superior under which circumstances, and to help communicate results. One of the efforts in the performance metrics program at NIST is the creation of reference test arenas for autonomous mobile robots. The first set of arenas was modeled after the Urban Search and Rescue (USAR) application and was designed to represent, at varying degrees of verisimilitude, challenges associated with collapsed structures. This is a domain that is very dangerous for rescue personnel and in which robots will likely be able to provide increasing levels of assistance in searching for survivors. [1] The arenas were first deployed at the American Association for Artificial Intelligence (AAAI) Rescue Robot Competition in 2000. In 2001, the arenas were used at the International Joint Conference on Artificial Intelligence (IJCAI). They will again be used at AAAI-2002. Additionally for 2002 and henceforth, the RoboCup Federation [3] will use the arenas to host their newly formed RoboCupRescue league competitions. A discussion of the details of these competitions is contained in Section 3 of this paper.

There are three sets of customers for the arenas. The first are researchers, who need testing opportunities. The repeatable obstacles (sensory and physical) that are focussed towards mobile robotic perception and intelligent behavior provide them with challenges for their robots. The second are the sponsors of research. They can use the arenas for validation exercises to objectively evaluate robots in structured, repeatable, representative environments. The arenas can be used to validate robotic purchases, identify strengths and weaknesses in systems, and compare the cost effectiveness of different approaches. Finally, the end users of the robots can benefit from the resulting performance metrics. The eventual goal is to develop standard performance metrics from the arenas that can be used by purchasers to evaluate mobile robot capabilities.

There were several motivating factors for building the arenas. The first was the desire to be able to compare "apples to apples" in a technological sense. When researchers publish results, they typically describe the performance of their systems in their laboratory or demonstration environments, making it difficult to compare and contrast with others researchers' results. Isolating tests for sensing, behaviors, and other robotic capabilities – and making these tests reproducible – allows the research community to make meaningful comparisons of algorithms, sensors, platforms, and other independent items. A standardization of these challenges, through use of the arenas, enables a direct comparison of approaches.

A second desire was being able to "teach to the test." The arenas provide an objective set of measures for evaluating different robotic implementations. The arenas are not idealized "blocks world" tests. They provide some fairly realistic challenges that mobile robots must be able to address to be considered capable in this domain. We hasten to add that the USAR domain is extremely challenging. Although the arenas do provide some elements of what may be encountered in a collapsed building, they are not representative of the reality of a disaster scene. Rather, they provide a step-wise abstraction of such challenges in an attempt to isolate and repeatably test specific robot capabilities.

Another concern of research sponsors and of researchers themselves is the slowing of progress due to re-invention of the wheel. When building a robot, numerous hardware and software subsystems are required and it is not possible (or very difficult) to reuse any work done by other organizations. By highlighting successful approaches negotiating well-known obstacles, it is hoped that others will better understand and adopt these approaches, and expedite their progress into other areas of research.

Finally, practice makes perfect: arenas that are available to researchers year-round should enable them to repeat experiments and therefore debug and improve their systems. The arenas are set up near the NIST campus in Gaithersburg, Maryland, and can be used by researchers year-round. Since robustness comes through repetition and testing outside perceived limits, the three arenas provide increasing levels of difficulty, so that researchers can move on to new challenges once they master the simpler sections.

2. DESIGN CONSIDERATIONS

2.1. Elements of Robotic Capabilities

The primary goal of the test arenas is to provide reproducible measurements and tests of autonomous mobile robots. There are several elements that come together to create a fully autonomous mobile robot. Recognizing that there are going to be different levels of autonomy implemented in mobile robots, the arenas are designed to isolate the different capabilities that may be available on any particular robot. They are shown schematically in Fig. 1. For a more in-depth discussion of the design considerations for the arenas, see [2].
At the lowest level is the locomotion capability of the robot's physical platform. Although two of the three arenas provide some challenges for locomotion and require general agility of the robots, our emphasis (and that of the AAAI competitions) is on algorithms. So the arenas attempt to isolate and test the higher elements of robot autonomy and do not address locomotion directly.

The element just above the hardware implementation of locomotion and sensors is sensory perception. The robot has to sense what is in its environment in order to navigate, detect hazards, and identify goals (simulated victims and their locations). Sensor fusion is an important capability, as no single sensor will be able to identify or classify all aspects of the arenas. The simulated victims in the arenas are represented by a collection of different sensory signatures. They have shape and color characteristics that look like human figures and clothing. They have heat signatures representing body heat, along with motion and sound. The

Figure 1: Constituent Elements of an Autonomous Mobile Robot

arenas are also designed to pose challenges to typical robot navigation sensors. For example, acoustic-absorbing materials confuse sonar sensors. Laser sensors have difficulty with shallow angles of incidence, smooth surfaces, and reflective materials. Highly regular striped wallpaper and other types of materials pose challenges to stereo vision algorithms. Compliant objects that may visually look like rigid obstacles require the robots to apply tactile sensors or other means of verifying that they can

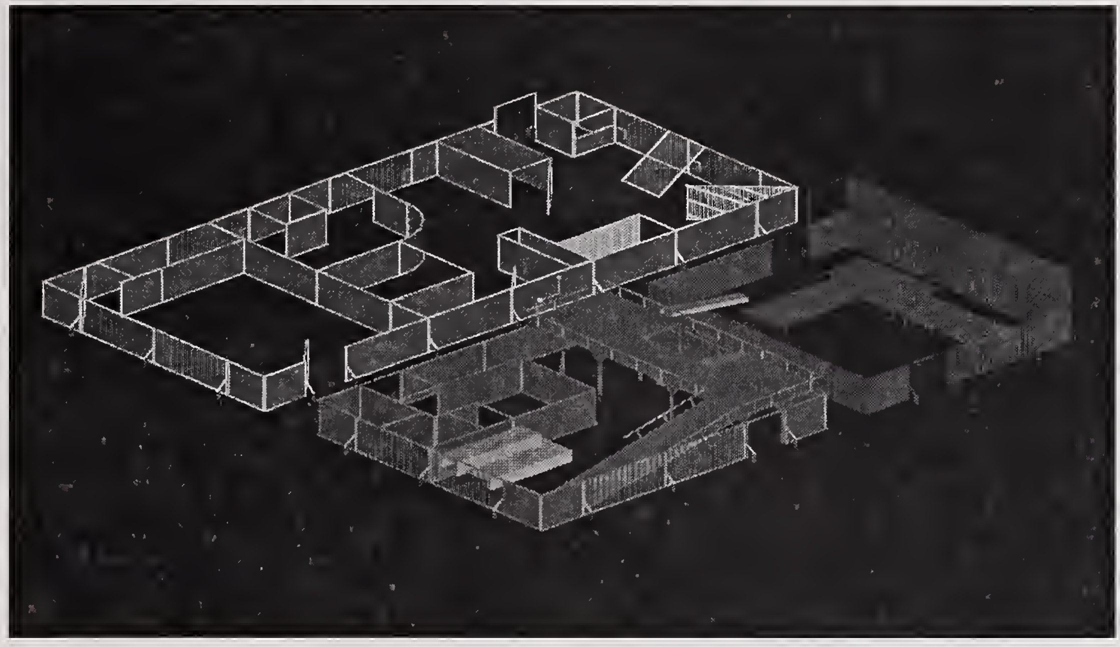
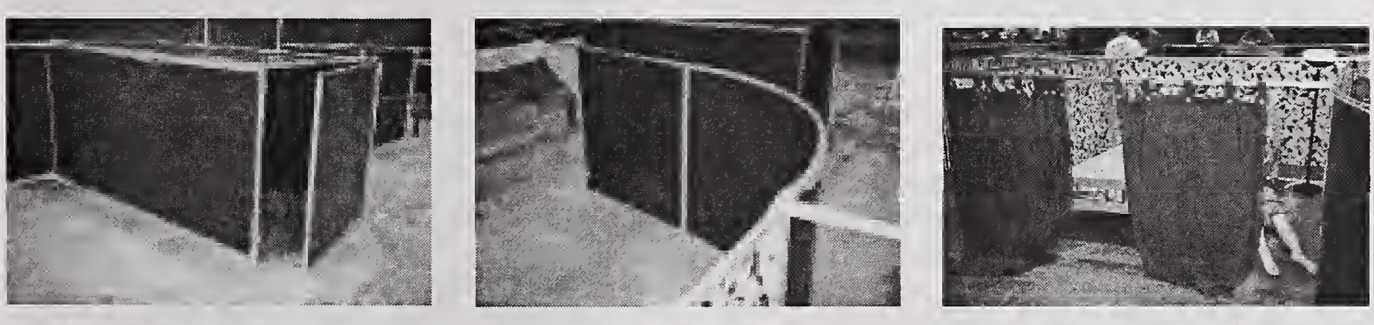


Figure 2: Model of the Reference Test Arenas for Autonomous Mobile Robots

indeed push them aside (e.g., open doors or curtains). Manipulation of rigid obstacles, such as closed doors or debris, provide more advanced challenges. Robot localization is another essential capability derived from sensing. Different flooring materials affect localization schemes based on wheel encoders. Additional cues from the environment need to be employed to help localize the robot in an effort to generate and maintain correct maps. Since the arenas represent collapsed structures and buildings, GPS is not considered to be available.

Knowledge representation is the next element. It encompasses the robot's ability to model the world, using both a priori information (such as might be needed to recognize certain objects in an environment) and newly acquired information (obtained through sensing the environment as it explores). In the mobile robot competitions for AAAI and RoboCupRescue, the robots are expected to communicate to humans the location of victims and hazards. Ideally, they would provide humans a map of the environment they have explored, with the victims' and hazards' locations marked. The environment that the robots operate in is three-dimensional, hence they should reason, and be able to map, in three dimensions. The arenas may change dynamically during a competition (as a building might further collapse while rescuers are searching for victims). Therefore the ability to create and use maps to find alternate routes is important.

The planning or behavior generation components of the robots build on the knowledge representation and the sensing components. The robots must be able to navigate around obstacles, make progress in their mission (that is to explore as much as possible of the arenas and find simulated victims), take into account time as a limited resource, and make time critical decisions and tradeoffs. The planner should make use of an internal map generated by the



a.) Darkened chamber with door **b.) Curved wall** **c.) Soft materials, victim under bed**

Figure 3: Features from the Yellow arena

robot and find alternate routes to exit the arenas that may be quicker or avoid areas that have become no longer traversible.
The overall autonomy of the robot is the next element to be evaluated. The robots must be designed to operate with humans. However, the level of interaction may vary significantly, depending on the robot's design and capabilities or on the circumstances. The intent is to allow for "mixed initiative" modes to limit human interaction, maximizing the effectiveness and efficiency of the collaboration between robot and humans. Robots may communicate back to humans to request decisions, but should provide the human with meaningful communication of the situation. Pure teleoperation is not a desirable mode for the robot's operation. The human should provide the robot with high level commands, such as "go to the room on the left" rather than joystick the robot in that direction.

The final element to be evaluated in the robot's overall capabilities is collaboration among teams of robots. One very rich area of research is in cooperative and collaborative robotics. Multiple robots, either heterogeneous or homogenous in design and capabilities should be able to more quickly explore the arenas and find the victims. The issues to be examined are how effectively they maximize coverage given multiple robots, whether redundancy is an advantage, and whether or how they communicate amongst themselves to assign responsibilities. Humans may make the decisions about assignments for each robot a priori, but that would not be as desirable as seeing the robots jointly decide how to attack the problem.

2.2. A CONTINUUM OF CHALLENGES

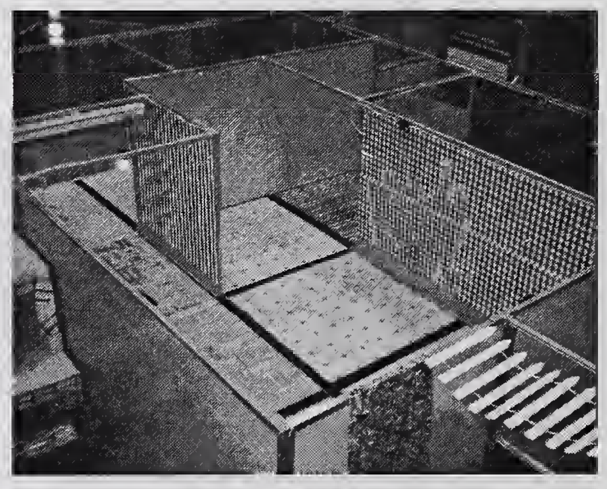
There are three separate Reference Test Arenas for Autonomous Mobile Robots, each labeled by a color denoting increasing difficulty. A schematic of all three arenas assembled together is shown in Figure 2.

The Yellow arena is the easiest in terms of traversability. Researchers who may not have very agile robot platforms, yet want to test their sensing, mapping, or planning algorithms, can use the Yellow arena only. The arena consists of a planar maze. There are isolated sensor tests, based on obstacles or simulated victims. The arena is reconfigurable in real time, with doors that can be closed and blinds that can be raised or lowered. The reconfigurability provides challenges to the mapping and planning algorithms of the robots. A series of photographs of the Yellow arena features are shown in Fig. 3.

The Orange arena provides traversability challenges. Different types of flooring materials are present and there is a second story, reachable via ramp, stairs, and ladders. Holes in the second story floors requiring the perception, mapping, and planning capabilities of the robot be able to consider a three-dimensional world. The Orange arena is also reconfigurable in real time. Fig. 4 shows some features from the Orange arena.

The Red arena provides the least structure and the most challenges. It essentially represents a rubble pile (but is transportable). It is very difficult to traverse, with debris of various sorts throughout the arena. The debris is problematic for most robot locomotion mechanisms and includes rebar, gravel, plastic bags, and thin pipes. Simulated rubble resembling cinder blocks is strewn throughout. There are simulated pancaked floors (floors collapsed onto lower

a.) Ramp and other routes to 2nd story



b.) Different flooring materials and
Figure 4: Features of the Orange arena

Figure 5: Red Arena

floors) and leaning collapsed walls which can be triggered to cause secondary collapses. For example, the flooring in certain sections is unstable and will collapse if a robot attempts to surmount it. These features encourage robots toward a safer, more tactile approach toward negotiating the environment. A view of the Red arena is shown in Fig. 5.

3. THE 2001 COMPETITIONS

The NIST arenas made their debut at the AAAI-2000 Rescue Robot Competition [4][5]. Their second deployment was at the International Joint Conference on Artificial Intelligence (IJCAI) in 2001, where the RoboCupRescue and AAAI Robot Rescue competitions were jointly held.

In preparation for the second competition, a great deal of attention was paid to the development of scoring rules. The competition rules were designed to produce a final scoring distribution that defines clear winners. The focus of the competition is on intelligence; hence the scoring system favors solutions that demonstrate on-board autonomy, intelligent perception, world modeling, and planning. Fig. 6 shows the scoring formula.

Scoring is biased towards high quality interactions with humans, meaning that there is low-bandwidth, high content, infrequent communications to and from humans. The robots are expected to present human-understandable maps of their findings, highlighting the location of simulated victims. The scoring formula heavily favors multiple robots managed by a single operator. Improving the 1:1 ratio of operator to robot (teleoperation) is a key focus for these events. Simple teleoperative implementations, remotely using human perception for navigation and target acquisition, are not rewarded well in the scoring formula. The intent of these competitions is to push the state of the art toward autonomous solutions, while encouraging effective mixed-initiative modes of operation along the way.

Some disincentives were built into the scoring to discourage undesirable traits in the robots. For example, using simple redundancy of robots, while demonstrating no clear collaboration among the robots, implying the team could simply afford more robots, was discouraged. If the team could not demonstrate a cost-benefit advantage to having more robots (homogeneous or heterogeneous), their scoring suffered. In general, teams deploying multiple robots were penalized when their human-robot interface could not facilitate control of multiple robots by a single operator.

Other considerations in the design of the scoring were reflective of the course's design. "Gaming" of the arenas, that is, learning the course and its characteristics in order to "tune" the robots to perform well was obviously undesirable. Human level maps gained from operators closely scrutinizing the arena layout and simulated victim locations, and then teleoperating based on that knowledge, clearly undermines the intent of the competitions. But deterring that in the scoring was difficult. Since there were some fairly easy simulated victims to find, a minimum score was required to qualify for one of the place awards. The scoring formula also was designed to reflect the increasing difficulty of navigating and searching each progressively more challenging arena.

Six teams registered for the competition, but only four actually competed. No team scored enough points to qualify for either first, second, or third place awards. The two most successful teams earned "qualitative" awards for demonstrating very different capabilities.

```
RobotRescueScore = (VictimsFound (NumberOfRobots / (1+ NumberOfOperators)^3)
AverageAccuracy

        VictimsFound  =(VictimsFoundInYellow / VictimsPlacedInYellow) (YellowVictimWeighting) +
                        (VictimsFoundInOrange / VictimsPlacedInOrange) (OrangeVictimWeighting) +
                        (VictimsFoundInRed / VictimsPlacedInRed) (RedVictimWeighting)

                                  [ YellowVictimWeighting  =  0.50 ]
                                  [ OrangeVictimWeighting  =  0.75 ]
                                  [ RedVictimWeighting     =  1.00 ]

        NumberOfRobots      =  Number of robots that find a unique victim
        NumberOfOperators   =  Number of operators having touched the robot or are in the hot zone
        AverageAccuracy     =  Average of the positional accuracy for each victim found
                               [ VictimAccuracy = (IsVictimInVolume)/(StatedPositionalVolume) ]
```

Figure 6: Scoring Formula at the 2001 RoboCup Rescue/AAAI Rescue Robot Competition

Swarthmore College (USA) demonstrated the most artificial intelligence capability, but only navigated within the easiest Yellow arena. The scoring formula required that the robots confined to the Yellow arena find all of the victims to earn the minimum score to qualify for a "place" award and be competitive with robots entering the other two more difficult arenas. They came close, finding all but one of the victims during one of their runs, falling just short of earning a "place" award. They received a "qualitative" for best artificial intelligence display.

Sharif University (Iran) demonstrated a more robust tracked robot, and even attempted to negotiate the Red arena. However, they had issues with their control strategy, bumping walls and obstacles frequently. They even triggered a secondary collapse of the pancaked flooring in the Red arena (an advanced obstacle). They resorted to identifying victims from outside the arena, but suffered from inherent inaccuracies in their approach. And they required too many human operators to manage their single robot, limiting their total score and keeping them from earning a "place" award. However, their effort was notable, and their robot mechanisms were well designed, so they earned a "qualitative" award for demonstrating the best hardware implementation. The experience will almost certainly allow them to improve their system for next year. Integration of more AI functionality should produce a very strong showing.

4. PROPOSED SCORING CHANGES

Given the experiences of two years of competitions within the Reference Test Arenas for Autonomous Mobile Robot, certain changes to the scoring seem reasonable. Note that these are the opinions of the authors and may or may not be reflected in the final rules for future mobile robot competitions.

The scoring formula should encourage robots to use a greater variety of sensors by awarding specific points for demonstrating superior sensory perception. This could be accomplished by awarding points for correctly identifying each sensor signature, or "sign of life," emitting from the simulated victims (form, heat, sound, motion). Since the simulated victims consist of various combinations of these sensor signatures, representing various states of consciousness and exposure, sensor fusion algorithms could deduce critical information regarding the state of the victim. This would allow more points to be scored per victim found, and would appropriately encourage the use of multiple sensors, along with sensory perception, sensory fusion, and error checking algorithms.

Some teams attempted to identify victims by looking through the clear windows on the perimeter of the arenas, thus avoiding the hazards within the harder arenas. The point values gained by identifying simulated victims from outside the course should be limited. The windows were placed to allow spectators visibility into the arenas, and to provide a realistic obstacle for the robots. However, since no agility is required when the robot is outside of the arenas, the robot should not receive full credit for victims found in the harder Orange and Red arenas. The point values in such cases should be equivalent to finding victims in the Yellow arena.

Several behaviors exhibited by robots in the competitions should be discouraged through point deductions. Foremost should be point deductions for crushing, or inappropriately contacting, victims. Finding a victim (scoring points) and then hurting that victim should produce limited net gain in terms of scoring.

Causing damage to the arenas or certain obstacles through purposeful, or inadvertent, contact with the environment should also be discouraged with point deductions. If a robot triggers a secondary collapse of debris, the results could be catastrophic leading to further injuries or worse. These robots need to learn to be as deft as rescue personnel in their interactions with the environment, and should be penalized when they fail. There are a few typical voids in the arenas that can be destabilized and collapsed. Triggering these collapses should cause severe point deductions. Some lesser deduction should be tied to routine bumping of walls and other obstacles, demonstrating perception, planning, or control issues.

Also, teams which deploy more than one robot but sequentially teleoperate each one should be more effectively recognized in the scoring formula as maintaining a 1:1, operator:robot ratio, and not be lavishly rewarded as are multiple robot teams.

Lastly, maneuvering a robot based on human knowledge of the arena layouts or simulated victim placements essentially thwarts the spirit of the competition and should be discouraged. This is, of course, harder to implement in the scoring formula. However, focusing a larger percentage of the scoring potential toward autonomous activities (perception, control, planning, mapping, collaboration), while allowing some points for teleoperative techniques (identifying simulated human forms via remote video), the incentives would at least be in line with the goals of the competition.

5. FUTURE ACTIVITIES

NIST's Reference Test Arenas for Autonomous Mobile Robots will continue to be used to host the AAAI Rescue Robot Competitions in 2002. After two years of competitions, no robot team has demonstrated the minimum capabilities required to earn a "place" award. So it appears the research community has been challenged effectively. The RoboCupRescue competition has adopted these same arenas to host their competitions, and will use the same scoring formula developed for AAAI. Replicas of the arenas will be built for each RoboCupRescue event and left in the host country. This will result in the dissemination of the arenas worldwide, raise awareness of the needs and challenges for search and rescue robots, promote the competitions, and enable researchers to practice in the actual arenas throughout the year.

In order to further disseminate the arena's challenges and encourage progress in mobile robotics, NIST is developing virtual versions of the arenas. The effort is two-fold. Initially, sensor datasets obtained from within the arenas will be made available for download from the internet. This will permit researchers to process the data captured from sensors directly in the arenas and develop their algorithms without the need for problematic robot hardware. Data from a range-imaging sensor and from a color camera will be the first datasets available. A second, more ambitious, effort involves creating a simulated environment representing the arenas into which teams can plug their algorithms, receive simulated sensor data, and send actuation commands to navigate simulated robots. Further interaction with the research community is needed to design and develop this environment.

6. CONCLUSIONS

Tangible, realistic challenge problems can provide robot researchers with direction and help focus their efforts and collaborations. Reproducible, and widely known, challenges can help evolving fields by providing reference problems with measures of performance. Therefore, competitions, such as the AAAI Rescue Robot, RoboCupRescue, and others, can be valuable in spurring advancements in robotic capabilities. Thus far, the Reference Test Arenas for Autonomous Mobile Robots have been very well received by the research community, and promise to provide a common set of reference challenges for the constituent elements of autonomous mobile robots. Their visibility in hosting competitions at AAAI, IJCAI, and other such events raises researcher's awareness of the types of challenges they must confront to be successful in the search and rescue domain. But the larger goal is to accelerate the advancement of mobile robotic capabilities through objective evaluation, collaboration, and the development of pertinent performance metrics, so that the capabilities that do emerge can be effectively applied to many other domains.

REFERENCES

[1] Blitch JG, " Artificial intelligence technologies for robot assisted urban search and rescue," EXPERT SYSTEMS WITH APPLICATIONS 11: (2) 109-124 1996.

[2] Jacoff, A., Messina, E., Evans, J., "A Standard Test Course for Urban Search and Rescue Robots," Proceedings of the 2000 Performance Metrics for Intelligent Systems Workshop, August 2000, Gaithersburg, MD.

[3] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., Matsubara, H., "RoboCup: A Challenge Problem for AI," AI Magazine 18(1): Spring 1997, 87-101.

[4] Murphy, R., Casper, J., Micire, M., Hyams, J., "Assessment of the NIST Standard Test Bed for Urban Search and Rescue Competitions," Proceedings of the 2000 Performance Metrics for Intelligent Systems Workshop, August 2000, Gaithersburg, MD.

[5] Schultz, A. (2001). "The 2000 AAAI Mobile Robot Competition and Exhibition," AI Magazine, 22(1), Spring 2001, AAAI, 67-72.

Hierarchical Architecture for Coordinating Ground Vehicles in Unstructured Environments

Alberto Lacaze
Intelligent Systems Division
National Institute of Standards and Technology
lacaze@cme.nist.gov

Abstract—**This article presents a hierarchy of planners that can be used to coordinate multiple autonomous vehicles for different applications. The particular architecture reduces complexity and creates a constrained representation that in turn generates a wide variety of complex behaviors. This article will concentrate on the upper levels of the hierarchy assuming that the autonomous mobility tasks can be executed by the lower levels of the hierarchy. A particular set of examples for the US Army's Demo III project will be presented.**

Keywords—**Planning, emerging behaviors, complexity, multiresolutional hierarchical control, Real-time Control Systems (RCS).**

I. Introduction

THE problem of coordinating multiple autonomous platforms has been thoroughly studied in the literature from operations research to artificial intelligence.

The manufacturing and operations research literature shows a long history of coordinating multiple manufacturing cells to optimize factory production [1]. Most of these methods are manufacturing domain dependent and do not always easily transfer to mobile vehicles in unstructured environments.

Another field of research that has historically created coordination of mobile vehicles can be found for aerial platforms. Methods for closed coupled formations of aerial vehicles to minimize drag have been studied using linear dynamic models [2], [3], and using non-linear models [4]. These approaches rely heavily on the dynamics of the airplanes to create classical control feedback techniques that theoretically warrant the stability of the formations.

Some attempts at controlling multiple vehicles have been in structured environments using behavior-based approaches.

The predecessor to the current system employs a behavior based approach was implemented for the Demo II project [5]. Since then, the behavior based approach has been abandoned and a hierarchical architecture based on Real-time Control Systems (RCS) is currently in use for the Demo III program. The main objective of the Demo III program is to create an autonomous scouting vehicle capable of traversing an unstructured off-road environment. Behavior-based systems like [5], [6], [7], [8] share many common components with hierarchical architectures with some important differences. They integrate several goal-oriented behaviors simultaneously. In most cases several behaviors are generated, and an arbiter or decision maker weighs these behaviors to create an "intermediate" behavior that better matches the cost criteria. The advantages of these systems is that they create interesting group behaviors in simple environments where the coordination can be done relying on local criteria and therefore require simple world representations. Some examples of flocking and schooling behaviors are presented in [9]. The downfall of these architectures is that, when applied to complex environments, the implementation of each of the many possible behaviors becomes cumbersome and situation dependent; and the arbiter rapidly increases in complexity.

On the other hand, hierarchical systems create a more explicit world representation. The cost criteria are used to evaluate a model of the system traversing the predicted world representation. In most cases only one behavior is generated at each level. First, a very coarse behavior is generated, and then this same behavior is refined at each level of the hierarchy. Proponents of hierarchical architectures argue that applying cost evaluation criteria is much easier to resolve using a complete representation as opposed to dealing with multiple, sometimes contradicting, sets of behaviors. However, complex world representation and the complexity of testing plan combinations make the implementation of hierarchical systems challenging. Both architectures contain reactive and deliberative (planning) components. Hierarchical architectures tend to lean towards planning solutions because they have a representation that allows the prediction components

A. Lacaze is with the Intelligent Systems Division of the National Institute for Standards and Technology. This work was partly funded under Cooperative Agreement No. 70NANB7H0020 with the University of Maryland, College Park

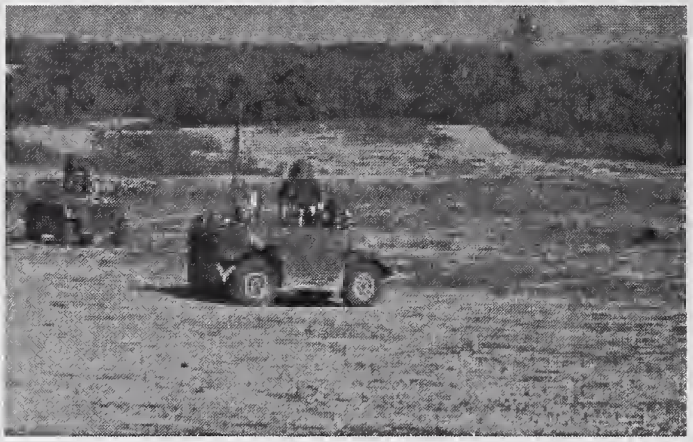


Fig. 1. Two Demo III vehicles performing an autonomous mission

Fig. 2. Autonomous vehicle followed by manned safety HMMWV

necessary for planning. Behavioral approaches tend to be more reactive in nature, which is sufficient in simple environments.

Other approaches taken in the literature include using classical control and stability techniques. However, since most mobile vehicles are non-holonomic, they cannot be asymptotically stabilized by smooth static-state feedback control laws. Some approaches have been taken using Lyapunov's second method [10] and smooth time-varying feedback control laws [11].

Many approaches in the literature concentrate in "tight" formations for ground vehicles where the exact location of each vehicle within the formation (parade like) is seldom used in military situations. The exact locations within the formation are very loosely followed in real scenarios and in general are not nearly as important as the overall sensor coverage, risk evaluation and distribution [12], [13], [14], [15].

Many applications assume that formations have a leader. Vehicles in the formation are then steered to a particular offset with respect to this leader [16], [17]. Other approaches allow for great freedom for the individual platforms, and the coordination is done for collision avoidance [18]. Some of these algorithms are based on the search of the configuration space [19]. [20] presents a comprehensive survey of robot coordination methods mainly concentrated for manipulators.

In this paper we will present a hierarchical system approach to controlling groups of vehicles. By imposing different constraints in the graph representation, complex militarily valuable behavior will emerge. Figure 1 shows the Demo III autonomous platforms being tested in Fort Knox in October 2000 running a hierarchical planning architecture. Figure 2 shows one autonomous platform traversing a challenging environment.

II. Hierarchical Architectures

Hierarchical architectures based on RCS [21] make use of multiple levels of coarseness or resolution to minimize complexity. First, very coarse plans are created that look far into the future and in space. In most realistic scenarios, plans that look further into the future can only be done coarsely, because our knowledge and ability to predict outcomes rapidly deteriorates the further out we try to predict. These coarse plans are then sent to other levels of resolution where a portion of them is refined. This portion is closer in space and time to the current state, and in general, more knowledge is available. This process is continued at each level until we reach a level where very detailed knowledge, and therefore, accurate predictions can be done. These higher levels of resolution plan very detailed plans which are short in scope. Lower levels of resolution create plans and representations that include large scopes. They are coarse and there are large amounts of time to plan (and re-plan) because the representation of the world tends to change more slowly at that resolution. At higher levels of resolution, the representation and plans are much more detailed. The re-planning cycles are comparatively faster, however, the scope is small. In general, the levels of the hierarchy are designed to create a similar level of complexity for different levels. Therefore, the number of levels depend on the complexity of the problem at hand [22], [23]. For simpler systems RCS degenerates into flatter architectures similar to [24] because only a few levels of resolution are necessary to deal with the combinatorial

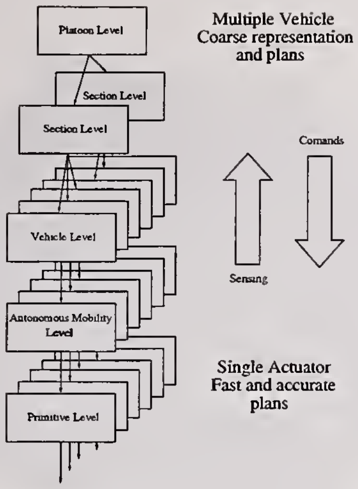


Fig. 3. RCS hierarchy for a scout platoon

complexity of the problem.

Hierarchical architectures are a very good match for military applications. Military personnel are very used to control hierarchies, and clearly understand their operation. Figure 3 shows a simple hierarchy for controlling a platoon. In the scenario presented, a platoon is composed of two sections [25]. Each section has several vehicles and each vehicle has its own vehicle level (similar to a vehicle commander), an autonomous mobility level (similar to a driver), and a primitive level which controls the vehicle. As in military structures, the commands flow from the top (platoon leader) to the bottom driver. In this paper we will concentrate on the upper two levels of this hierarchy and assume that the vehicle level and autonomous mobility levels have already been implemented in such a way that they can receive and carry out the commands passed by the upper levels.

III. Planning Algorithms

Most planning algorithms start from the following premises:
1. the universe of discourse can be subdivided into discrete states;
2. there is a starting (or current) state;
3. there are one or more goal states;
4. there is a cost associated with moving the systems from one state to another;
5. there is a cost associated with being at a state;
6. the planner must find one or more paths that will take the system from the starting state to a goal state, minimizing the cost along its motion.

Specifically, let $G = (V, E, s, f, \psi)$ be a digraph where V is a finite set of nodes, vertices or states. E has ordered pairs subsets of elements of V of called edges, that is, $E \subseteq V \times V$. $s, f \in V$, and represent a starting state and a finish state, respectively. $\psi(e)$ is a function where $e = [v_1, v_2] \in E$ and $v_1, v_2 \in V$ which computes the cost of traversing e. A planner is an algorithm $\phi(G)$ which returns a directed walk w through G (informally plan). $\phi(G) = w = (s, v_1, v_2, \ldots, v_n, f)$ where $v_1 \ldots v_n \in V$ minimizing $\sum_i \psi(e_i)$ where $e_0 = [s, v_1]$, $e_1 = [v_1, v_2]$, ..., $e_n = [v_n, f]$. $\phi(G) = \emptyset$, if there are no plans from s to f.

In most planning problems for a single ground vehicle:

- $\exists\, f : \Re \times \Re \to V$ where $\Re^2$ represents the location of the vehicle. A subsampled $\Re^2$ is used for computing the vertices of the planning graph.
- $\forall\, v_i, v_j \in V$, if $L(v_i, v_j) < thr, \exists\, e_l = [v_i, vj] \in E'$. $L(.)$ is a distance measure. In other words, vertices are connected within a vicinity.
- $E = \{e_k \in E' : Constrained(e_k) = False\}$ where $Constrained : E \to \{True, False\}$ is defined to represent the constraints that the vehicle may have (i.e. areas not allowed).

Once G is created, there are many optimal and sub-optimal $\phi(G)$ described in the literature. Specifically, Dijkstra's algorithm and A* are commonly used to find these paths optimally. Both algorithms are easily implemented for replanning so that even the complexity of the second cycle is lower in the average first plan.

IV. Planning Algorithms for multiple vehicles

For two vehicles it is possible to build

$$f_4 : \Re \times \Re \times \Re \times \Re \to V \qquad (1)$$

$\Re^4$ represents the position of the two vehicles. As expected, the number of elements in V and in E increases very rapidly. However, as we will see in the following examples, this is not a problem for formations. Formations create large amounts of constraints so that the number of elements of V becomes manageable.

For example, let $[x_a, y_a]$ and $[x_b, y_b]$ be two adjacent vehicle locations (i.e., $L([x_a, y_a], [x_b, y_b]) < thr$). Figure 4 shows the edges without any constraints associated with the 4D graph created for planning using the representation outlined by Definition 1.

At this stage, the number of vertices and edges that create a graph as defined could easily overwhelm the computing power, as well as the memory resources of any modern computing device. In the next few sections this paper will show how this graph is pruned by using constraints to create a graph that can be optimally searched in real time.

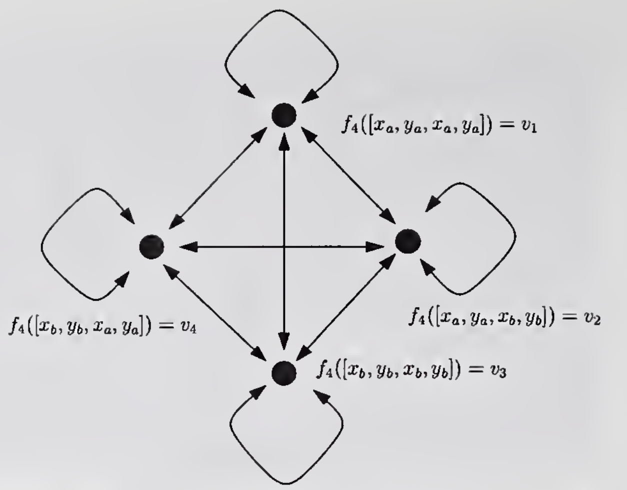


Fig. 4. Edges between two adjacent map locations for two vehicles

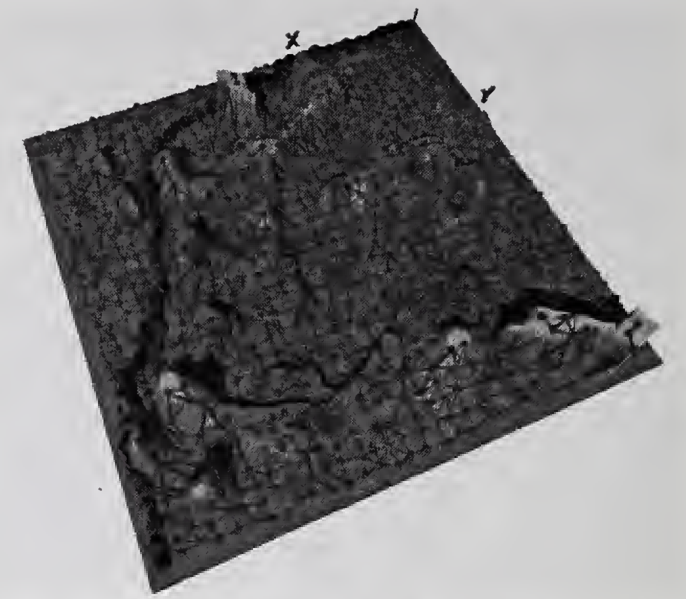


Fig. 5. Two vehicles traversing a terrain following tight distance constraints

V. Adding Constraints to Achieve Scouting Behavior

The constraints introduced in the following subsections are based on the doctrine taught to Army scouts, and it is based on [25].

It is assumed that only edges and vertices that fit the constraints are used in the creation of the graph, as opposed to creating the complete graph and then pruning it. Although similar conceptually, the amount of memory and computations required to do the former is generally orders of magnitude smaller than the later one.

A. Distance constraints

In most cases for scout maneuvering (and in most formations), there are distance constraints that must be maintained for it to be called a formation. In the case of scouting behavior, two often used constraints are as follows:

1. vehicles must not be more than l meters away from each other. A more complicated measure may actually force the vehicles in line of sight with each other. The reasons for this constraint from a military perspective are clear: cover each other, and if a vehicle gets shot, the second vehicle should find out where the shot originated.
2. vehicles must be more than m meters away from each other. This is done so that both vehicles will not be disabled by a single detonation.

Specifically, $\forall\, e_k \in E'; e_k = [[x_a, y_b, x_c, y_d], [x_e, y_f, x_g, y_h]]$:

$$\begin{aligned} &Constrained(e_k) = True \\ &iff\ L([x_a, y_b], [x_c, y_d]) > l \vee L([x_a, y_b], [x_c, y_d]) < m \vee \\ &\quad L([x_e, y_f], [x_g, y_h]) > l \vee L([x_e, y_f], [x_g, y_h]) < m \end{aligned} \tag{2}$$

Depending on the l and m chosen, this set of constraints reduces the space of search by a considerable amount. Figure 5 shows two vehicles traversing an artificially created terrain. The blue and red trails starting at the origin, represent the paths generated by the two vehicles. The underlying grid represents a two dimensional projection of the 4 dimensional graph stretch over the terrain. The map is 5 km in size, and the vehicles must be within 500 m of each other. It is possible to see from the figure that the vehicles travel mostly parallel to each other when the terrain permits, and they travel in a column when the terrain does not. There are no constraints or change in cost evaluations for the different behaviors. They travel this way because it is optimal with respect to the cost function.

Figure 6 and 7 show two vehicles traversing a artificially created maze-like and GPS generated elevation maps. The first picture shows the two vehicles with $l = 500m$ and with $l = 1500m$ (m was selected as to keep the number of nodes constant). Note that neither vehicle is following an optimal path. The paths followed by both vehicles are optimal overall. In this context, optimality refers to the fact that no other path that the two vehicles follow will give a lower cost within the given graph and constraints. This is different from

Fig. 6. Two vehicles traversing a terrain following tight distance constraints

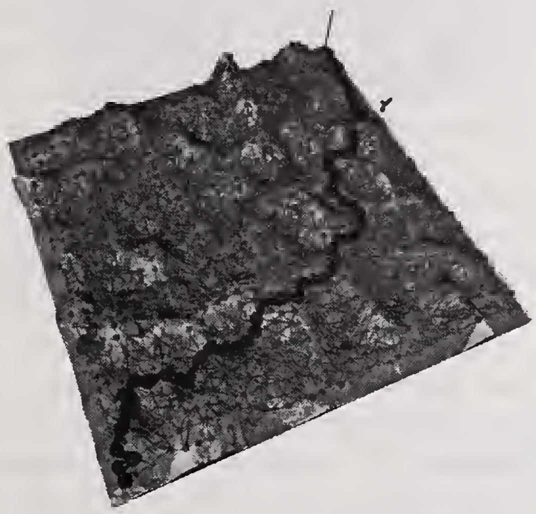


Fig. 7. Two vehicles traversing a terrain following relaxed distance constraints

the standard approach where an optimal path is found for one vehicle, and the other vehicles are constrained to the path found for the first one. In our case, the paths of both vehicles are optimized simultaneously. There are no heuristics that being used by the system (other than the constraints) that change the behavior of the system by optimizing the cost function. Very different behaviors automatically emerge depending on the terrain.

B. No Stopping Allowed

In some cases it may be necessary to only allow vehicles to stop or slow down in particular areas, and continue their moving the rest of the time. These constraints can be implemented as follows, $\forall e_k \in E'; e_k = [[x_a, y_b, x_c, y_d], [x_e, y_f, x_g, y_h]]$:

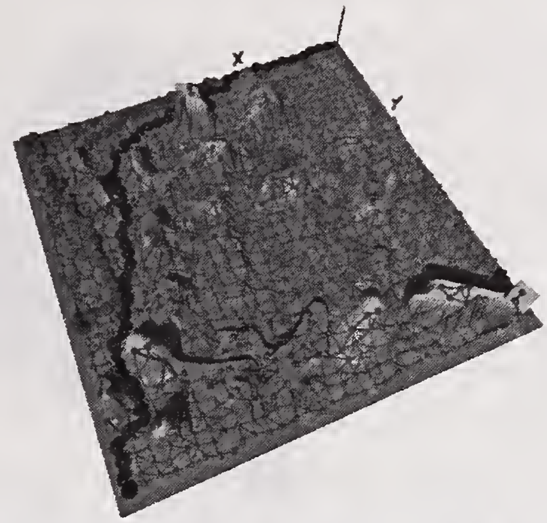


Fig. 8. Two vehicles traversing a terrain following relaxed distance constraints and a same path distance constraint

$$
\begin{aligned}
&Constrained(e_k) = True \\
&iff\ (x_a = x_e \wedge y_b = y_f) \vee (x_c = x_g \wedge y_d = y_f) \vee \\
&\quad L([x_a, y_b], [x_c, y_d]) > l \vee L([x_a, y_b], [x_c, y_d]) < m \vee \\
&\quad L([x_e, y_f], [x_g, y_h]) > l \vee L([x_e, y_f], [x_g, y_h]) < m
\end{aligned} \tag{3}
$$

Figure 8 shows the results of applying these constraints. The starting points for one of the vehicles was modified to meet the 500m minimum distance constraint. By comparing Figure 5 to Figure 8, it is possible to see that one of the vehicles is following an optimal path while the second one is moving out of the way of the first vehicle to meet the distance constraints.

C. Leap Frog

A commonly used strategy for scouting vehicles is a "leap frog" traversal, referred to as boundary overwatch or traveling overwatch. In these cases, only one vehicle moves at a time, while the other takes an observation position over the first vehicle. If one of the vehicles is shot, the other vehicle will be paying close attention to identify the direction of the fire and other details of the encounter.

These constraints can be implemented as follows, $\forall e_k \in E'; e_k = [[x_a, y_b, x_c, y_d], [x_e, y_f, x_g, y_h]]$:

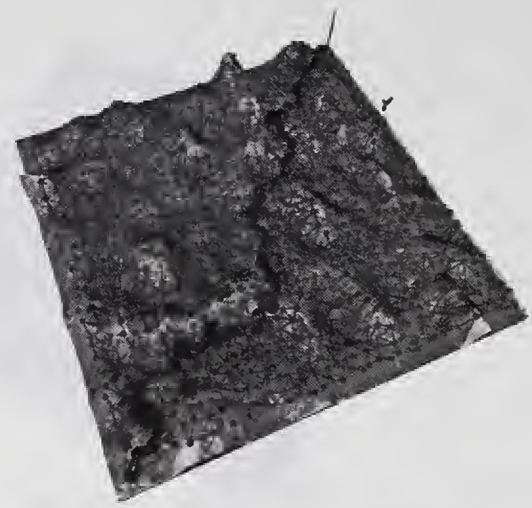


Fig. 9. Two vehicles traversing a terrain following tight distance constraints and leap frog constraints

$$
\begin{aligned}
&Constrained(e_k) = True \\
&iff\ (x_a! = x_e \wedge y_b! = y_f) \vee (x_c! = x_g \wedge y_d = y_f) \vee \\
&\quad !LOS((x_a, y_b), (x_c, y_d)) \vee !LOS((x_e, y_f), (x_g, y_h)) \\
&\quad L([x_a, y_b], [x_c, y_d]) > l \vee L([x_a, y_b], [x_c, y_d]) < m \vee \\
&\quad L([x_e, y_f], [x_g, y_h]) > l \vee L([x_e, y_f], [x_g, y_h]) < m
\end{aligned} \tag{4}
$$

where $LOS((x_a, y_b), (x_c, y_d)) = true$ if and only if (x_c, y_d) can be viewed (or cleared) from (x_a, y_b). LOS stands for line of sight. Figure 9 shows the results of applying these constraints. In this example, only one vehicle is allowed to move at the same time. If the cost for stopping is increased assuming that the vehicles have to take cover, the vehicles perform longer leaps. They generally stop at locations that give good visibility so that the other vehicle can perform long leaps and still be in the line of sight of the other vehicle. This explains the number eight pattern that can be seen in the path by the vehicles in the Figure 9.

VI. Coordinating Larger Numbers of Vehicles

In order to coordinate large numbers of vehicles, the dimensionality of the proposed approach becomes large. Although the number of constraints grows, this may not be enough to create small enough graphs for real time usage. In order to coordinate larger number of vehicles, following the examples given by the military organizations, we make use of hierarchical structures. Figure 10 is a schematic of the approach. At the top of this hierarchy, the platoon level creates a very coarse plan for all sections. Representation methodology, and planning strategy are the same at this level. The main differences between the levels are the coarseness of the representation as well as features of interest, cost evaluations and constraints.

In the example shown, the platoon level not only has distance constraints, but other sets of constraints do not allow sections to overlap with each other (following military doctrine). The graph is 6 dimensional where each pair of dimensions represents a rough location of each section. For the figure, the enemy is assumed to be to the left of the image, therefore, the leftmost section carries out a "leap frog" movement, while the two rightmost sections organize into more relaxed formations.

Following the lessons evolved in military doctrine, if a larger number of entities need to be coordinated, more levels would be added to the hierarchy. It is possible to describe hierarchies as sets of rules that constrain the space of search and therefore reduce complexity. This example shows that hierarchical tools designed for human entities can easily translate to artificial systems. In this example, if the paths for all six vehicles would have been searched in one level, the number of nodes and edges required to create a similar path would have overwhelmed the memory as well as the computational capabilities of the system. In general the results would not deviate from the optimally found in the 12D space.

Opponents of hierarchical systems often mention that hierarchies have a "bottleneck". In most cases these problems are caused by poor system design. Complexity of planning and representation determine the number of levels to be used for any particular system. If a level carries too much burden, then, more levels can be created to alleviate its complexity. On the down side, hierarchies create "bureaucratic" costs of communicating representations and commands between levels. In general, these added costs are negligible compared to the savings [22].

VII. Conclusions

Autonomous vehicles have been a central point of attention in recent years. The ever increasing number crunching capabilities of modern computers, as well as the recent advancement in sensor technology are paving the way for the implementation and deployment of groups of autonomous vehicles. Therefore, the need for robust formation control will become an important factor in future military applications. This paper presented a viable solution for the planning of formations of vehicles that closely resembles military organizations. It presents a departure from the behavioral approach commonly found in the literature, with

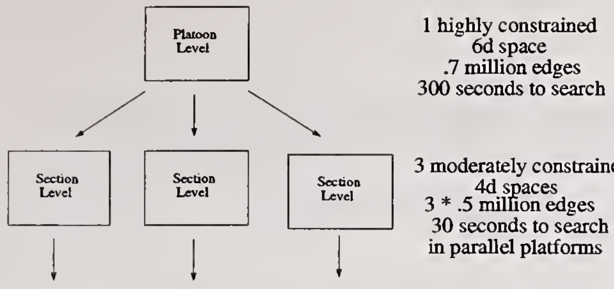


Fig. 10. Platoon Level and section levels.

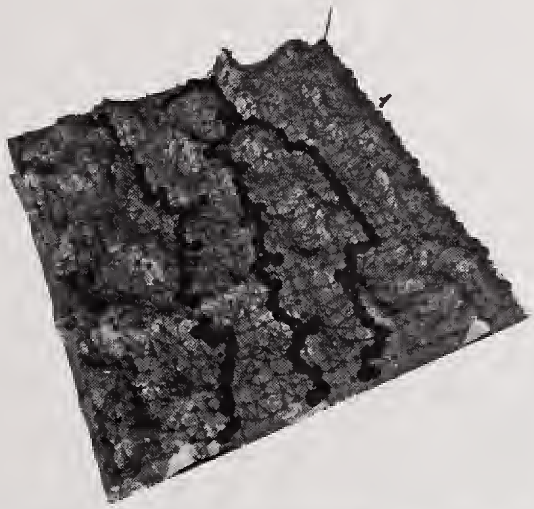


Fig. 11. A platoon formed of 3 sections with 2 vehicles each performing different section behaviors

some specific advantages:

- The system performs formation planning for multiple vehicles at the same time, as opposed to planning for one and having the others attached by control laws. In the paths created by these graph search techniques are not susceptible to the local minimums that can easily be found in ad hoc heuristics (bridges and multiple obstacles) because of their larger scope of temporal and spatial representation.
- The performance of the system is optimal within the graph representation and the constraints allocated.
- All levels shown in the examples can be easily implemented in desktop computers and allow for real-time operations at the shown resolutions. The shown examples create about 5×10^5 edges, seconds to create, plan and re-plan the graphs.
- The representation allows facilitates the generation of constraints to generate complex behavior that can result into into tactically correct behaviors.

References

[1] D. Brown, J. Marin, and Scherer W., "A survey of intelligent scheduling systems," *Intelligent Scheduling Systems*, 1995.

[2] A. Proud, M. Patcher, and J.J. D'Azzo, "Close formation control," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Portland, OR, 1999, pp. 1231–1246.

[3] J.D. Wolfe, D.F. Chichka, and J.L. Speyer, "Decentralized controllers for unmanned aerial vehicle formation flight," in *AIAA Guidance, Navigation, and Control Conference*, San Diego, CA, 1996, pp. 29–31.

[4] C.J. Schumacher and R. Kumar, "Adaptive control of uavs in close-coupled formation flight," in *Proceedings of the American Control Conference*, Chicago, Illinois, June 2000.

[5] T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 926–939, 1998.

[6] J. Rosenblatt, "DAMN: A distributed architecture for mobile navigation," in *AAAI 1995 Spring Symposium*, 1995.

[7] M.J. Mataric, "Reinforcement learning in the multi-robot domain," *Autonomous Robots*, vol. 4, pp. 73–83, 1996.

[8] R. A. Brooks, *Intelligence without Reason*, MIT Press, 1991.

[9] C. Reynolds, "Flocks, herds and schools," *Computer Graphics*, vol. 21, pp. 25–34, 1987.

[10] Pomet92a, "Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift," *System Control*, vol. 18, pp. 147–58, Feb. 1992.

[11] H. Yamaguchi and JW. Burdick, "Time-varying feedback control for non honolomic mobile robots forming group formations," in *37th IEEE Conference on Decision and Control*, Tampa, FL., Dec. 1998, pp. 4156–4163.

[12] U.S. Army, *Field Manual No 7-7j*, Department of the ARMY, May 1993.

[13] K. Singh and K. Fujimura, "Map making by cooperating mobile robots," in *IEEE International Conference on Robotics and Automation*, 1993.

[14] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Second International Conference on Autonomous Agents*, 1998, pp. 47–53.

[15] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *IEEE International Conference on Robotics and Automation*, Apr. 2000, pp. 476–481.

[16] Y. Moskovitz and N. DeClaris, "Basic concepts and methods for keeping autonomous ground vehicle formations," in *International Conference on Intelligent Systems*, NIST, 1998.

[17] S. Sheikholeslam and C.A. Desoer, "Control of interconnected nonlinear dynamic systems: the platoon problem," *IEEE Transactions on Automatic Control*, pp. 806–810, 1992.

[18] T.Y. Li and J.C. Latombe, "On-line manipulator planning for two robot arms in a dynamic environment," *Journal of Robotic research*, vol. 16, pp. 144–167, 1997.

[19] T. Perez-Lozano, "Spatial planning: A configuration space approach," *IEEE Trans. Computers*, vol. 32, 1983.

[20] E. Todt, G. Raush, and R. Suarez, "Analysis and classification of multiple robot coordination methods," in *IEEE International Conference on Robotics and Automation*, San Francisco, Apr. 2000, pp. 3158–3163.

[21] J. Albus, "Outline for a theory of intelligence," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, pp. 473–509, 1991.

[22] Y. Maximov and A. Meystel, "Optimum design of multiresolutional hierarchical control systems," in *Proceedings of IEEE Int'l Symposium on Intelligent Control*, Glasgow, U.K., 1992, pp. 514–520.

[23] J. Albus, A. Meystel, and A. Lacaze, "Multiresolutional planning with minimum complexity," *Intelligent System and Semiotics*, 97.

[24] R. Arkin, *Behavior Based Robotics*, MIT Press, Cambridge, MA, 1998.

[25] U.S. Army, *Field Manual No 17-19 - Scout Platoon*, Department of the ARMY, May 1993.

Evaluating Knowledge and Representation for Intelligent Control

Elena R. Messina, John M. Evans, James S. Albus
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230

ABSTRACT

Knowledge and the way it is represented have a tremendous impact on the capabilities and performance of intelligent systems. There is evidence from studies of human cognitive functions that experts use multiple representations in problem solving tasks and know when to switch between representations. In this paper, we discuss the issues pertaining to what types of knowledge are required for an intelligent system, how to evaluate the knowledge and representations, and provide examples of how representation affects and even enables functionality of a system. We describe an example of an intelligent system architecture that is built upon multiple knowledge types and representations and has been applied to a variety of real-time intelligent systems.

1. INTRODUCTION

Various definitions of intelligence, whether pertaining to artificial or biological, make reference to knowledge. The American Heritage Dictionary defines intelligence as "the capacity to acquire and apply knowledge." Newell and Simon stated that "a physical symbol system has the necessary and sufficient means for general intelligent action." [20] Despite this, there is a paucity of literature that provides guidance to developers in terms of what is the needed knowledge within an intelligent system and how to decide on appropriate representations. This is especially true when it comes to building real-time intelligent systems, such as those for controlling autonomous mobile robots and advanced manufacturing equipment.

2. STATUS OF KNOWLEDGE AND REPRESENTATION

In 1989, Wah stated that "despite a great deal of effort devoted to research in knowledge representation, very little scientific theory is available to either guide the selection of an appropriate representation scheme for a given application or transform one representation into a more efficient one." [24] There is little evidence to repudiate this statement in 2001, particularly for real-time control.

The most basic aspect of representation design is based on pairing it to the algorithms that use it. It is well known in computer science that there is a relationship between the representation of data and the algorithms that operate on it. Efficiency of algorithms is highly dependendent on the organization of the data, therefore a starting point for design and evaluation of knowledge representation should be based on broader computer science tenets, such as those described in [16].

Davis et al. argue for a broader understanding of what knowledge representation entails [7]. Certainly representation, in any form, is a surrogate for things that exist in the real world. The issue of required fidelity of representation therefore arises. They also see knowledge representation as a set of ontological commitments, meaning that the representation choice serves as a "strong pair of glasses that determine what we can see, bringing some part of the world into sharp focus, at the expense of blurring other parts." The focussing/blurring effect is crucial because of "the complexity of the natural world is overwhelming." They conclude that knowledge representation researchers ought to characterize the nature of the glasses they are supplying, thus making the ontological commitments explicit, and that the field ought to develop principles for matching representations to tasks.

In general, most of the literature describes the use of a single representation for all the knowledge within a given system. In mobile robotics, one sees three main approaches. The first is geometry-based, where sensors or probabilistic models are used to build maps. The second is feature-based, where the topology of the environment and high-level objects of significance are stored. The third is a symbolic approach, where first-order logic or rule-based systems are used. Examples of geometry-based approaches include occupancy grids [18] and sensor-based map building [23]. Feature-based systems include [14] and [25]. Symbolic systems include STRIPS [9] and GOLOG [15]. Exceptions to this "monomodeling" design do exist, such as the hybrid intelligent systems of Devedzic [8], the multimodeling system of Chittaro [6], and the qualitative and quantitative representations of Kuiper's semantic spatial hierarchy [13]. In most cases, these multirepresentational approaches have not been applied to functioning real-time controllers.

Evidence from the cognitive science field indicates that human problem solving capabilities rely heavily on the ability to switch between representations as required [5]. Chittaro et al. [6] note that systems that reason about physical systems require

- representation adequacy
- problem solving power
- problem solving economy
- multiple uses of knowledge (for multiple problem-solving tasks)
- cognitive coupling
- efficiency

They also claim that "efficiency cannot be achieved, in general, using only one model: an appropriate problem decomposition and the cooperation of a variety of knowledge sources organized at different levels of aggregation and accessible under appropriate views is possibly the only way of adequately coping with complexity issues."

3. MULTI-REPRESENTATION EXAMPLE

One example of a multi-representational approach to real-time intelligent systems is the Real Time Control System (RCS) and its mobile autonomous vehicle version, 4D/RCS [1][2]. A general framework for the RCS model-based control system is shown schematically in Fig. 1. This framework shows a hierarchical control structure with a world model hierarchy explicitly interspersed between the sensor processing hierarchy and the behavior generation or task decomposition hierarchy. Example labels for three of the levels (subsystem, primitive, and servo), per [1] are shown. Note that the subsystem level for locomotion is referred to as "Autonomous Mobility" in 4D/RCS implementations.

Within RCS, there are three distinctly different types of knowledge: system parameters at the servo level, maps, images and object models at the next levels, and symbolic data at the highest levels. We briefly describe each of these.

3.1 System Parameters

The lowest level for RCS, as for any control system, is the servo level. At the servo level, position, velocity, and/or torque are controlled by voltage values applied to motors or valves. Knowledge of the value of system parameters is needed to control these values. Control knowledge, such as gains and filter coefficients, is typical of the type of parametric knowledge common at this level. These are commonly represented as scalars.

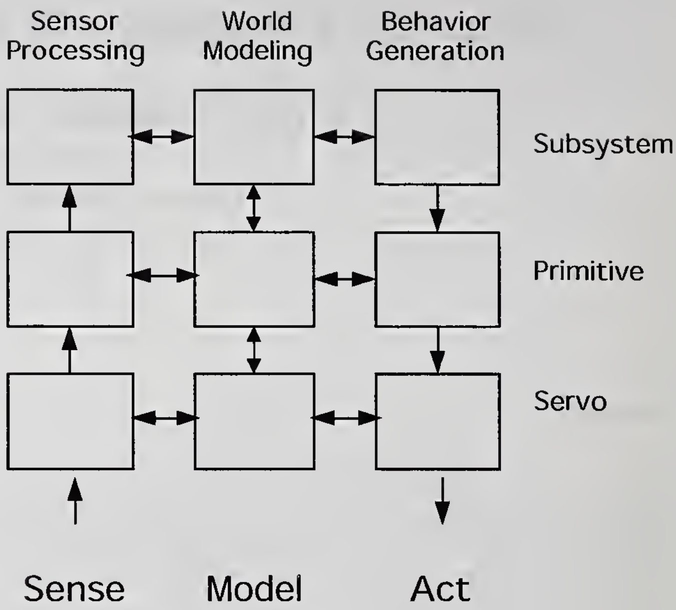


Figure 1: General Framework for Intelligent Control

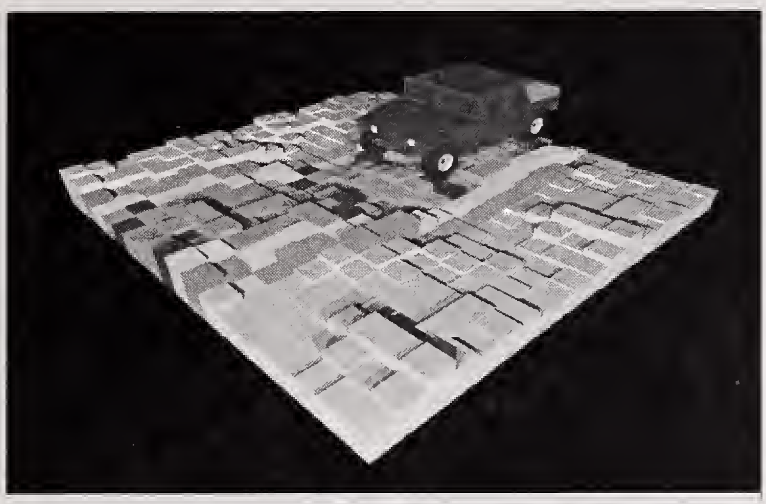
Any errors that deal with a single degree of freedom, such as ball screw lead errors, contact instabilities, and stiction and friction are best compensated for at this level.

3.2 Iconic Knowledge

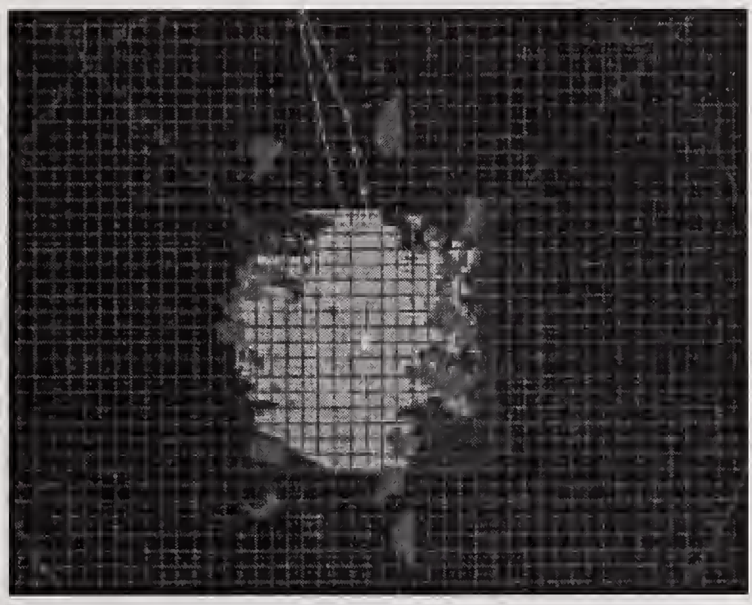
Multiple individual servo loops are coordinated at the next higher level. Interaction between axes comes into play, requiring knowledge of spatial dimensions, which we refer to as geometric or iconic knowledge. Iconic knowledge typically represents Euclidean space and includes maps, images, part models, and other geometric information. The relationship of the entities in time and space is captured through maps, images, and trajectories. Motion control for machine tool axes is computed at this level.

For mobile autonomous robots, maps are a natural representation for the environment in which the robot must function. Maps are defined as any two (or higher) dimensional grid with attributes referenced to the grid. A simple occupancy grid may indicate whether a cell is free or not (or passable or impassable by the robot) and the path planning algorithms will use shortest distance between start and goal cells, while avoiding impassable cells. A more sophisticated world model for an outdoor mobile robot may include a variety of feature layers, such as road networks, hydrology, elevation, intervisibility, and vegetation. The various features must be taken into account when planning movement and combined according to a weighting scheme based on the mission of the robot and current situation.

Maps used by an implementation of an outdoor mobile autonomous robot based on 4D/RCS are shown in Fig. 2.



a) Primitive Level



b) Autonomous Mobility Level

c) Vehicle Level

Figure 2: Maps at 3 Levels of 4D/RCS

[3] [19] Each level of the hierarchy concerns itself with a different spatial and temporal extent and resolution. The values listed below are representative examples for an implementation and may vary based on the computing configuration, sensors, and features supported. The features that the map contains at each level are also different, based on the area of focus for that level's planning.

Fig. 2a shows the map at the Primitive level, where planning for the robot's motion takes into account the kinematics and dynamics of the vehicle. The Primitive level of the hierarchy plans at roughly 10 Hz frequency and within a space of 5 m surrounding the vehicle (which is centered in the map) and a resolution of 20 cm. or less. This level of the hierarchy simulates the movement of the vehicle along potential obstacle-free paths and evaluates the position of the 4 wheels as they are placed along the trajectory to find the most traversable path. Terrain elevation is evaluated from range data provided by the Laser sensor, enabling computation of how stable and how rough a given path would be.

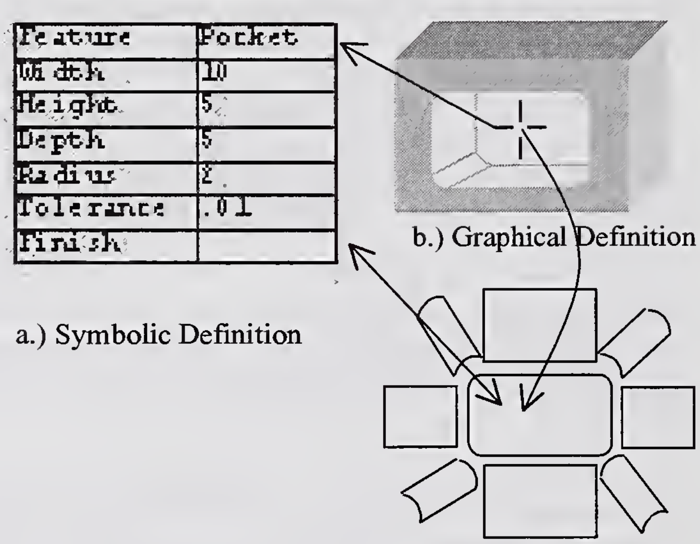
The next level up, referred to as Autonomous Mobility, plans at a frequency of 4 Hz within the 50 m surrounding the vehicle (which is again, centered in the map), with a resolution of 40 cm. Generally, this level of the hierarchy is concerned with avoiding obstacles and hazards to the navigation of the vehicle. The features that are contained in the map at this level include obstacles, cover, and roads obtained from sensory processing. Fig. 2b shows a combined Primitive level and Autonomous Mobility level map. The central square shows elevation (gray), unseen areas (blue) and obstacles (in red) detected by processing input from the vehicle's laser scanner sensor. The obstacles propagate to the Autonomous Mobility map (outside the blue and gray square). Not shown in the Fig. 2b are the precomputed feasible trajectories for the vehicle, given a starting wheel angle and velocity. The feasible trajectories that are blocked by obstacles are eliminated from consideration. Computing them offline enables the system to efficiently produce kinematically and dynamically stable steering commands.

Fig. 2c shows an example of the highest level currently implemented, the Vehicle level. This level plans within a map that is 500 m square, at a 4 m resolution, once a second. Planning at this level is concerned with generating a path between the current location of the vehicle and its goal point(s) (the operator may have specified certain waypoints or just an end location) while taking into account mission requirements. The paths generated for a mission that is stealthy versus one that gives highest priority to speed are completely different, yet the world model and the planner utilized are identical. Only the cost functions that are applied to evaluating candidate paths change. The features represented at the Vehicle level include road networks, water, vegetation, elevation, risk (for each grid in the map, which locations can see that grid) and visibility (for each grid, which other locations can be seen). Features are typically obtained from *a priori* digital terrain maps.

3.3 Symbolic Knowledge

At the highest levels of control, knowledge will be symbolic, whether dealing with actions or objects. A large body of work exists in knowledge engineering for domains other than control, such as formal logic systems or rule based expert systems.

At the present time, symbolic knowledge has not yet been implemented in the vehicle application of RCS, but it has in manufacturing ones [17][12]. An example of a symbolic description of a solid model of a block is shown in Fig. 3. The description notation is the International Standards Organization Standards for the Exchange of Product Model Data (STEP) Part 21 [10]. Symbolic representations such as this have been used to automatically generate manufacturing process plans from part models [12]. Reasoning about a pocket feature is appropriate at higher levels of process planning. This is in contrast to having to jump directly to the geometric representation and try to derive appropriate machining sequences based solely from the surfaces of the final part geometry.



c.) Geometric Definition

Figure 3: Pocket Feature.

Linguistic representations provide ways of expressing knowledge, expressing relationships, manipulating knowledge, and of extracting new knowledge based on knowledge already expressed, including the ability to address objects by property. Behaviors can be efficiently captured through symbolic representations. For example, in an autonomous vehicle system, entities such as "cars," "pedestrians," and "bicycles" each have certain properties and anticipated possible behaviors that affect the autonomous vehicle's planning *vis a vis* these other entities. A car can be expected to travel only on roadways (in normal circumstances) and to generally stay in a lane, whereas pedestrians may be expected to traverse roadways. Bicycles may squeeze between cars and straddle two lanes. The symbolic representation for each of these can be used in an intelligent system to derive potential behaviors in the near future and in the proximity of the autonomous vehicle. The symbolic entities may therefore be used to populate a map layer, such as the ones described in Section 3.2, based on current state information and expected potential behaviors. Higher level symbolic knowledge drives map-based (iconic) world model representations.

3.4 Other Dimensions in Knowledge

Another distinction within RCS is whether knowledge has been programmed into the system, is accessed from longer-term stores (*a priori* knowledge) or if it has been acquired or learned by the system recently during its operation (*in situ* knowledge) [17]. This distinction provides a framework for considering learning and adaptive control.

A final differentiation is in terms of whether knowledge pertains to things (nouns) or actions, task, or behaviors (verbs). This is akin to the distinction that the ancient Greeks made regarding "knowing that" versus "knowing what." System designers can make use of this distinction when matching sensor processing and world model specifications to the control task specification. This becomes very useful at higher levels in considering the interaction of autonomous machines with complex environments, where appropriate behaviors depend upon the nature of the objects encountered in the environment [2]. Generative process planning for machining or inspection [12] makes use of this distinction. Representations of actions will require a temporal element, unlike representation of things. An event has a time associated with it such as start, end, or duration.

4. EVALUATING KNOWLEDGE AND REPRESENTATION

Several obvious challenges exist in evaluating the knowledge that a system contains. It is difficult to isolate the world model from the sensing functions that populate and update it. The content and quality of the world model is dependent on the sensors and processes that are external to it. It is similarly difficult to separate the contribution of the world model independently from the planning subsystems that use it. There may be a very complete and efficient world model, yet the planning algorithms may be mismatched with it, poorly implemented, or inefficient.

Although it will be challenging, quantitative measures of the efficiency, completeness, and effectiveness of the representation must be developed.

Some may argue that, if a system works correctly, the particulars about the implementation are of no consequence. This is a shortsighted view of the science and engineering of intelligent systems. In order for the field to progress, successful and not so successful experiences must be shared. In this way, the capabilities of a system can be known and the best approaches can be leveraged by others in order to "raise all boats."

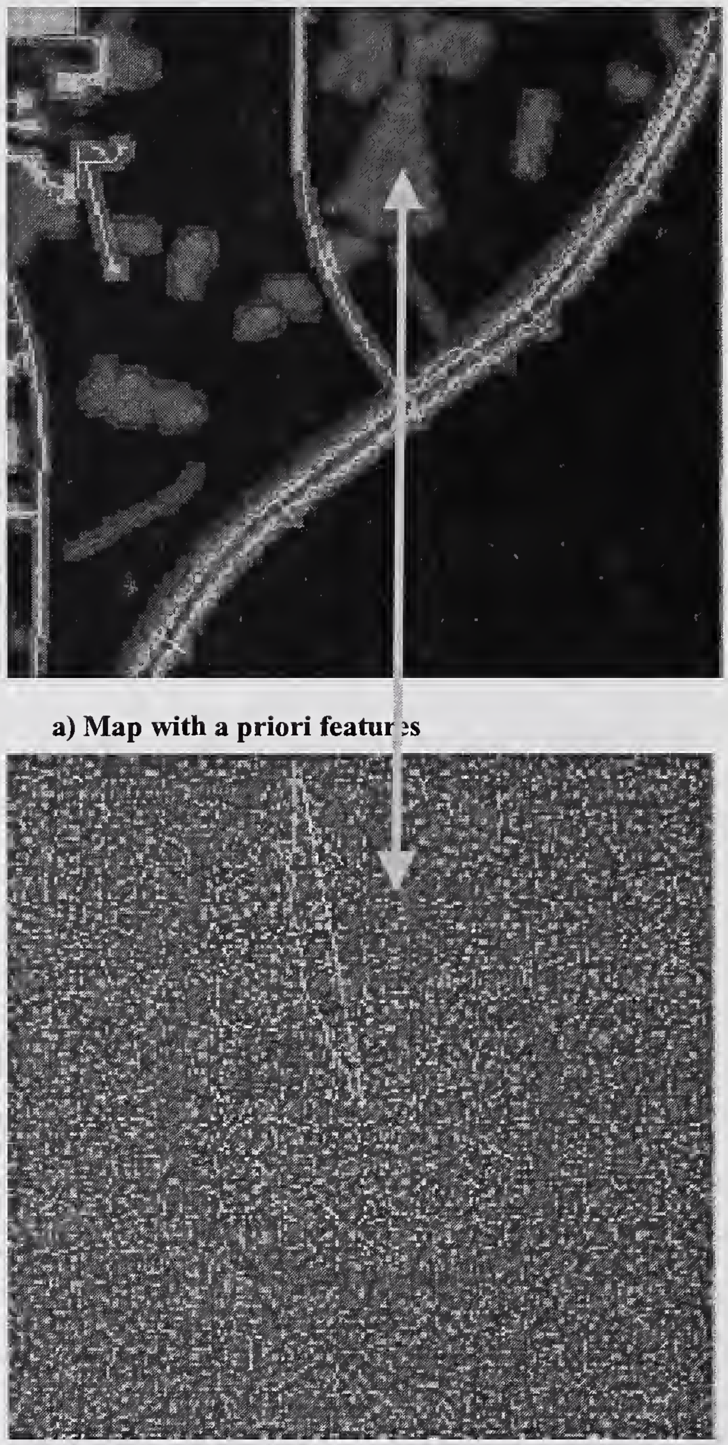
There are several aspects of knowledge content and representation that can be evaluated in an intelligent system, for which the community should strive to develop quantitative measures. We briefly present a few examples of evaluations without claiming this list to be exhaustive.

- The systems's ability to use *a priori* knowledge, and update it with newly-acquired knowledge. It is vital for most applications that the system start performing its tasks with given knowledge. That may take the form of maps of the area where an autonomous vehicle is expected to drive, a catalog of available cutter tools for machining, or an ontology to facilitate natural language interaction. When operating in the world, the intelligent system will have to sense changes in its environment and update its internal models. The new knowledge has to be placed in context of existing knowledge. Obstacles encountered during movement have to correctly update *a priori* maps. Tools that are no longer available must be deleted from the local copy of the tool catalog. Idioms or new terminology must be integrated into the language ontology.
- Mapping the environment in order to accomplish the given task. For a system that operates in the physical world, a current representation of its surroundings is crucial. Therefore, the system must be evaluated for its ability to understand and interact with a dynamic environment, including moving objects.
- Understanding general as well as specific concepts. Humans can accommodate thinking about the abstract and the concrete. Intelligent systems need to know about general classes of entities, such as "elevator" in addition to specific instances of elevators that they have to interact with. All elevators can be used to travel between floors, but the user interfaces for specific instances vary considerably. Another example is the concept of window, which may be important to a military scout robot. The general concept is important as it plans to look for windows during its mission. When it recognizes objects that fit that category, it must then plan its actions with respect to the specific instances. Windows may or not be see-through. They may be used to enter a building, but the robot needs to realize that windows at higher floors may not be useful for entering a building (unless the robot can scale the walls).
- Dealing with incomplete and imperfect knowledge. The system must accommodate and reason about partial and incorrect information about its environment. If not, it will rapidly be unable to cope.
- The correctness of the knowledge that a system holds. The system should be able to store *a priori* (given) knowledge correctly and be able to acquire correct knowledge. Correctness measures may be based on validation against ground truth or they may be evaluated based on confidence values based on multiple or redundant sensing.
- The efficiency of the knowledge representation. There are always many alternatives when implementing a system. The general representation approach (e.g., symbolic versus iconic) for a particular category of knowledge is one coarse aspect that can be examined. It may only be necessary for a system to store a structure that defines an entity as a tank and includes high level definitions such as min-max dimensions, make, model, friendly/foe, rather than an occupancy grid in three dimensional space or a solid model of the tank's geometry.

Once the dimensions of knowledge and representation that are to be evaluated are identified, the actual evaluation process is still a challenge. In this emerging new technology of intelligent systems, there are few examples of evaluation procedures that specifically target the knowledge itself, as opposed to the overall system performance. One of the key aspects of evaluations is that they be accurate and reproducible. We will describe some possible approaches to address these requirements.

Test arenas and scenarios are already being used to test robotic system capabilities. Examples include RoboCup [11][22] and the American Association for Artificial Intelligence Competitions, such as the Urban Search and Rescue Robots and Hors d'Oeuvre Anyone [21]. In the urban search and rescue competition, robots enter arenas that represent a collapsed building and search for targets that represent victims and hazards. The robots are supposed to communicate to human supervisors the locations of each victim and hazard. This requires at minimum the ability to map the environment and localize objects within the maps. The competition arenas have second stories, hence a good representation scheme would accommodate a third dimension. An excellent competitor would produce a map of every area explored, not just coordinates of the targets.

Virtual test environments and simulators can also be used to glean the knowledge representation aspects of intelligent systems. A virtual environment is one in which an organization can "plug in" their software and have the intelligent system, such as a mobile robot, receive simulated inputs from the environment and compute outputs to the virtual actuators. The level of interfaces from and to the virtual environment may be high level or, for high fidelity systems, could be equivalent to the interfaces to the actual sensors and servos. Isolating the world modeling databases and processes becomes feasible with the right simulation or virtual environment.



a) Map with a priori features

b) Node Status after Planning Cycle

Figure 4: Correspondence between planning space and physical features

Test harnesses that can be hooked up to knowledge bases can be used to evaluate its contents. A knowledge base that has been functioning and updating as an intelligent system performs its tasks can be isolated, either after the tasks are completed, or at certain points during operation. The harness can be used to query the contents of the knowledge base. For instance, it can check what entities have been detected in the environment and where they were estimated to be located. A harness would require defining or making known interfaces to the knowledge base.

5. KNOWLEDGE REPRESENTATION MATTERS

In this section we very briefly present examples of how the type of representation chosen for knowledge can affect the capabilities and effectiveness of a system. The examination of these examples is cursory and is meant to stimulate thought.

The first example is a classic taken from [20]. As an introductory exercise, a checkerboard, eight by eight squares, is to be covered by rectangular tiles. Each tile covers exactly two of the squares in the checkerboard. How many tiles are needed to completely cover the board? The solution is obvious (64/2=32) and can be easily found by a computer algorithm that searches through a grid-based representation of the checkerboard. Now, take away 2 of the squares, one from the top left corner and one from the bottom right. 62 squares remain, so one might naively assume that 31 tiles should be able to cover the remaining squares. The computer program that performs a search will have to expend a lot of compute cycles and may not be equipped to confront the fact that with this geometric configuration, there is no solution that fully covers the board with tiles. A different representation is better suited to quickly reach the correct conclusion. If the board is viewed as 2-tuples of black and red squares, since two same color squares can never be adjacent, then a tile covers each tuple of exactly one red and one black square. The missing corners took away 2 squares of the same color, hence there are more squares of one color than the other. Given this perspective, it is impossible to cover the board completely with tiles.

A second example is taken from [4]. In Balakirsky's system, a graph representation is used to solve planning problems. The LAyered World Modeling and Planning System (LAWMPS) has been applied to path planning for autonomous military vehicles. The world model in LAWMPS consists of a set of layers, organized in a grid representation. Each layer is dedicated to a particular feature, such as roads, vegetations, buildings, and sensed obstacles. The cost map is built by computing the contribution of each layer to the cost of having the vehicle traverse that location. The cost weights, which control the contribution of each feature are variable and determined by user preferences, modes, and objectives. A subset of the grid locations is used to generate the nodes and arcs for the planning graph. The planning process proceeds on the resulting graph, where each node represents a location, and the arcs have costs associated with moving between two specific locations.

Having the graph connect nodes that align with a vehicle-centered map grid and applying a Dykstra search algorithm can lead to discovery of knowledge that is useful to a mobile robot. "Problem" areas in the graph (where the search essentially stalls) as the search progresses can be correlated with map features and used to extract rules about traversability or other aspects of the problem state. In Figure 4a, an *a priori* map is shown with trees and fences (red), buildings (blue), and roads and parking lots (green). Figure 4b shows the node states after a cycle of planning. Green ones have never been visited, blue ones are closed (all their children have been visited), and red ones are still open). Due to the spatial relationship between the planning space and the *a priori* maps, the correspondences are clear: one area that appears problematic in the graph space is shown to correspond to a fenced or treed area, which would be impassable by the vehicle. Balakirsky uses this correspondence to allow the system to learn rules about planning.

6. CONCLUSIONS

Knowledge content and representation are critical aspects of an intelligent system. In constructing intelligent systems, there is a need for more science and engineering in the area of what should be represented and how it should be represented. Work in the area of knowledge representation has not, for the most part, addressed the area of real-time intelligent control. We argue that there are several categories of knowledge and types of representations that are necessary within a system that demonstrates advanced capabilities. Much work still needs to be done in understanding how to capture, use, and build knowledge within these systems. It is imperative to capture quantitative data about systems that demonstrate intelligence so that the field can benefit and move forward.

REFERENCES

[1] Albus, J.S., Lumia, R., Fiala, J., Wavering, A., "NASREM -- The NASA/NBS Standard Reference Model for Telerobot Control System Architecture"," Proceedings of the 20th International Symposium on Industrial Robots, Tokyo, Japan, October 4-6, 1989.

[2] Albus, J., "4-D/RCS: A Reference Model Architecture for Demo III," NISTIR 5994, Gaithersburg, MD March 1997.

[3] Balakirsky, S. and Lacaze, A., "World Modeling and Behavior Generation for Autonomous Ground Vehicles," Proc. Of the IEEE International Conference on Robotics and Automation," April 2000.

[4] Balakirsky, S., "A Layered World Model for Intelligent Planning," journal article in review.

[5] Bauer, M. I. and Reiser, B., "Incremental Envisioning: The Flexible Use of Multiple Representations in Complex Problem Solving," Proceedings of the 12th Conference of the Cognitive Science Society, Cambridge, MA,. 1990, pp. 317-324.

[6] Chittaro, L., Guida, G., Tasso, C., Toppano, E., "Functional and Teleological Knowledge in the Multimodeling Approach for Reasoning about Physical Systems: A Case Study in Diagnosis," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 6, November/December 1993.

[7] Davis, R., H. Shrobe, et al. 1993. "What is in a Knowledge Representation?" AI Magazine, Spring 1993.

[8] Devedzic, V., "A Survey of Modern Knowledge Modeling Techniques," Expert Systems with Applications 17 (1999) 275-294.

[9] Fikes, R. and Nilsson, N., STRIPS: a new approach to the application of theorem proving to problem solving," ARTIFICIAL INTELLIGENCE, 2(3-4): 189-208, 1971.

[10] ISO 10303-21: Industrial Autonmation Systems And Integration—Product Data Representation And Exchange—Part 21: Clear text encoding of the exchange structure. Geneva, Switzerland: ISO 1994:1994.

[11] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., Matsubara, H., "Robocup A Challenge Problem for AI," AI Magazine 18(1): Spring 1997, 87-101.

[12] Kramer, T., Huang, H., Messina, E., Proctor, F., Scott, H., "A Feature-Based Inspection and Machining System," Computer-Aided Design; Vol. 33; August 2001; pp. 653-659.

[13] Kuipers, B., "The Spatial Semantic Hierarchy," Artificial Intelligence 119 (2000) 191-233.

[14] Kunz, C., Willeke, T., Nourbakhsh, I., "Automatic Mapping of Dynamic Office Environments," Autonomous Robots Journal. 7(2) 1999.

[15] Levesque, H., Reiter, R., Lesperance, Y., Lin, F., Sherl, R., "Golog: A Logic Programming Language For Dynamic Domains," Journal of Logic Programming, 31:59-84, 1997.

[16] Lewis, H. and Denenberg, L., Data Structures and Their Algorithms, Addison Wesley, 1991.

[17] Messina, E., Horst, J., Kramer, T., Huang, H.M., Tsai, T.M., Amatucci, E., "A Knowledge-Based Inspection Workstation," Proceedings of IEEE Intn'l Conf. on Intelligence, Information & Systems, Bethesda, MD, Oct 31-Nov 3, 1999.

[18] Moravec, H., and Elfes, A., "High Resolution Maps from Wide Angle Sonar," Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, 1985.
[19] Murphy, K., Abrams, M., Balakirsky, S., Coombs, D., Hong, T.H., Legowik, S., Intelligent Control for Unmanned Vehicles, Proceedings of the World Automation Congress Conference (WAC 2000), Maui, HI, June 11-16, 2000.
[20] Newell, A., and Simon, H., "Computer Science as Empirical Inquiry: Symbols and Searc," Communications of the ACM, Volume 19, No. 3, pp. 113-126, March, 1976.
[21] Schultz, A., "The 2000 AAAI Mobile Robot Competition and Exhibition," AI Magazine 22(1), Spring 2001, 67-72.
[22] Stone, P., ed., "RoboCup 2000: The Fourth Robotic Soccer World Championships," AI magazine 22(1), Spring 2001, 11-38.
[23] Thrun, S., Fox, D., Burgard, W., "Probabilistic Mapping of an Environment by a Mobile Robot," Proceedings of the 1998 IEEE International Conference on Robotics & Automation.
[24] Wah, B., Lowrie, M., Li, G., "Computers for Symbolic Processing," Proceedings of the IEEE, Vol. 77, No. 4, April 1989, 509-540.
[25] Yeap, W. K., Jefferies, M. E., "Computing a Representation of the Local Environment," Artificial Intelligence 107 (1999) 265-301.

Evaluating the Performance of *E-coli* with Genetic Learning From Simulated Testing

A. Meystel, J. Andrusenko
Drexel University, Philadelphia, PA 19104

Abstract

This paper addresses the problem of finding the techniques of performance evaluation for elementary agents. From an evolutionary standpoint, the robust navigational algorithms were used by even the simplest of biological systems because the systems were able to learn how to evaluate their performance. The objective of this paper is to study one of the simplest biological, yet intelligent systems, an *E. coli* cell, and see how this could be of benefit to the design of control strategies for the single-agent intelligent systems. The robot is equipped with sensors and actuators, has a rudimentary knowledge representation system and is capable of conducting search, i.e. is equipped by the means of decision making. The robot itself is looked upon from a two-dimensional perspective and is analyzed in a computer-simulated environment. We present a design of the **Variable Structure Controller (VSC)** that combines the properties of any two structures or strategies from the ten initially available to our robot. **VSC** equipped robot should be able to come up with its own strategies of motion, without human intervention. The system under consideration supports the rudimentary learning subsystems that could be envisioned. The idea of using **Genetic Programming (GP)** is not introduced here for the sake of finding the best controller but rather for the purpose of demonstrating that improved functionality can be achieved *via on-line or simulated learning.*

Keywords: Escherichia coli; evolutionary computation; genetic algorithms; genetic programming; intelligent agents, mobile robots; motion planning; navigation, natural search

1. Genetic Programming as a Combination Mechanism in VSC

We introduce a **VSC** that combines properties of any two strategies using the principles of **Genetic Programming (GP)** [1]. The idea of using **GP** is not introduced here for the sake of finding the best controller, but rather for demonstrating that the improvement of functioning can be achieved without making a thorough investigation, and, even ON-LINE, while moving towards the goal. By thorough investigation we mean the investigation of ALL possible meaningful combinations of strategies' properties, which could be a very time consuming task. Our robot has 10 different strategies to choose from (Appendix 1). It knows how well each strategy performs in the environment it is in right now. It also knows which of the five performance criteria it wants to either minimize or maximize (Appendix 2). Lets assume that we want to maximize the efficiency (ε = **[D_{EUC} / D_{total}] * 100 %**). It is our desire for the robot to reach the goal while traveling along the most prferable trajectory. Under the first scenario conditions, **Experiment 1**, simply choosing Strategy 5a as the most efficient one will not lead to the efficiency optimization. Hence, we must allow our robot to somehow let its controller to evolve in order to maximize (minimize) a desired criterion.

Genetic Programming (**GP**) originated from Genetic Algorithms (**GAs**). The main difference between **GP** and **GAs** is in the way the solution to the problem is represented. **GP** creates new computer programs as the solution whereas **GAs** generate a string of numbers or some quantity that represent the solution. **GP** is a lot more powerful than **GAs**. In essence, **GP** is the key in creation of intelligent systems that program themselves.

GP can be useful in the problems where there is no ideal solution, (for example, a program that drives a car or operates a tank) [2]. Moreover, **GP** is very useful in finding solutions where the variables are constantly changing (for instance, a robot's positioning). Generally, the program will find one solution for one type of environment, while it will find an entirely different solution for another one.

Step 1 - Initial (Virtual) Population

First, an initial population of random computer programs is generated. In our case we will assume that our 10 strategies comprise the initial population. All of the computations and changes take place within a single robot's "mind".

Step 2 - Reproduction Mechanism

Then, each program (strategy) in the population is executed and assigned a fitness value according to how well it solves the problem. Our *E. coli* robot already knows how well each strategy performs in the environment it is currently in. If a strategy performs **above** or **below** the average **depending on the performance criterion** chosen, it is considered to be "fit", and, hence, will be allowed to participate in the reproduction process. For example, if our fitness function is based upon the **efficiency criterion**, a strategy with the **efficiency criterion** above the average is considered to be "fit". However, if the fitness function is based on the **energy criterion**, a strategy with the **energy criterion** above the average is considered to be "unfit". The pseudocode for the reproduction mechanism is shown in Table 1.

Table 1: Pseudocode for the Reproduction Mechanism

CHOSE the Performance Criterion for optimization, PCO

FIND AVE = average ($PCO_{Strategy\ 1}$ $PCO_{Strategy\ 1a}$ ... $PCO_{Strategy\ 5a}$)

***for** i=1:10*

if PCO of a particular strategy > (<) AVE, name this strategy FIT

***else**, name it UNFIT*

end

end

Step 3 – Formation of a New Population

After that, a new population of computer programs (strategies) is created. "Parents" are chosen randomly, in pairs, based upon their fitness. Two parents produce two children. The population size usually remains fixed for the duration of the search [3, 4].

The following sub-steps take place:

a) The best existing strategies are copied into a new population.

b) *Crossover*

New computer programs (strategies) are formed as a result of a ***crossover*** (sexual reproduction). In our case, during ***crossover***, the chosen "parenting" strategies swap the bottom halves of their programs (second parts) to produce two children. This process is represented below graphically:

The probability of *crossover* was chosen to be 0.6. If a randomly generated number in [0,1] interval is less than a *crossover* probability, the chosen pairs of strategies will go for crossover [5]. If *crossover* doesn't occur, the exact copies of parents are placed into the new population. The pseudocode for the process of crossover is represented in Table 2.

c) *Mutation*

New strategies are formed as a result of *mutation*. Here, we will somewhat deviate from the traditional definition of the mutation

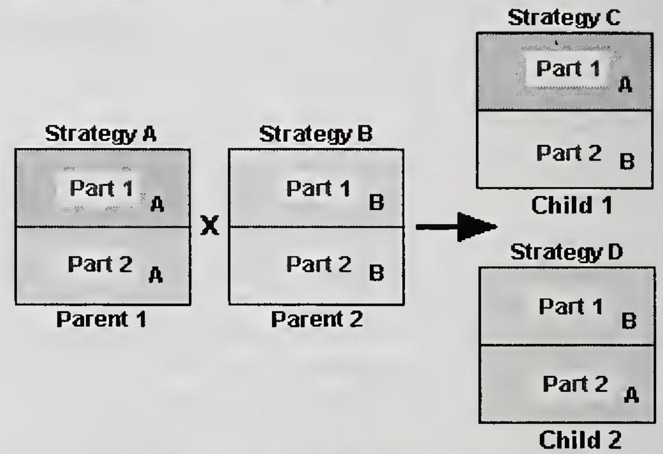


Figure 1: The Process of Crossover

mechanism to suit the design purposes of our robot's controller. First of all, in our design, it was a desire to have a *mutation* probability of 1 (usually it is preferred to have a very low mutation rate[5]). Then, we define the *mutation* operator as a change of some Control Variable Parameter's value to a randomly generated number. For instance, the average length of the robot's jump μ_J can undergo mutation when specified, i.e. μ_J will be changed to some random value. The pseudocode of the mutation process is shown below:

Table 2: Pseudocode for the Process of Crossover

***while** the formation of a new population is NOT completed*

{

Randomly choose two parents out of FIT strategies

Generate a random number P in the [0, 1] interval

***if** P < **0.6**, CROSSOVER and place two children into a new population*

***else**, place the exact copies of parents into a new population*

end

}

CALCULATE Performance Criteria of a new population

Table 3: Pseudocode for the Mutation Process

while the formation of a new population is NOT completed

{

MUTATE a particular strategy by randomly changing a specified Control Variable Parameter

}

CALCULATE Performance Criteria of a new population

Step 4 - The Best-So-Far Solution

The best strategy that appeared in any generation, "the best-so-far solution", is designated as the result of **GP** [6].

Benefits of GP Implementation in VSC

In previous chapter, we roughly estimated the most plausible ranges of operation for the **Control Parameters**. However, for the particular scenario, we never found a specific value of each **Control Parameter** under which a specific strategy would perform the best. We have 6 **Control** Parameters, 6 sets of values per **Control Parameter**, and 10 control strategies. Under assumption that there are at least 20 values per set, we would have to perform 1200 computations! Instead of performing all 1200 computations we could simply allow our strategies to ***mutate***, let's say for 5 generations. In other words, now, we would do the same type of calculations but with 5 randomly chosen values from each set of 20. The number of computations reduces to 300. However, we are not guaranteed that these 300 computations would contain the *best solutions* (but we are hoping). Most likely, we are able to determine just improved solutions.

In a summary, what are the possible benefits of **GP** implementation into our controller? First of all, as it was mentioned earlier, we believe that it is possible to find the ***improved*** (not necessarily the best of all) solution without making a thorough investigation of all meaningful combinations of control rules. Second, with this type of controller, our robot could improve its operability while still moving towards the goal, i.e. being ON-LINE!

VSC should be able to:

- Reduce the computational complexity via **GP**, by finding better solution (best-so-far and not necessarily the best of all) faster
- Create new strategies otherwise unimaginable to humans
- Improve robot's behavior while it is still in motion towards the goal, i.e. stay ON-LINE
- Reduce the cost factor
    - All of the calculations and iterations happen inside a **single robot's "mind"** (as opposed to multiple intercommunicating agents)

When we refer to our robot being ON-LINE, we envision the following scenario: While being in ON-LINE mode, i.e. while being on its way to the goal, our robot could *locally* evaluate the **Performance Criteria** of the strategy it's currently using, every ***n*** units of time. Then, it would decide on whether to change its strategy of motion or not in accordance with the results.

Experimentations with Genetic Operators: Mutation and Crossover

1st Set of Tests: Using the Reproduction and Mutation Mechanisms only (Scenario 1)

Below is the general schema used for this particular set of tests:

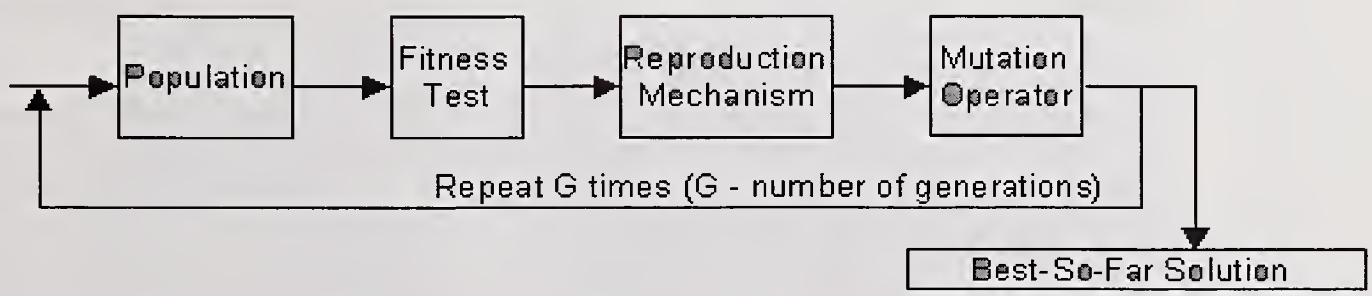


Figure 2: General Schema for the 1st Set of Tests

Table 4 describes the algorithm used for this set of tests. We chose the *efficiency* criterion to be our *Performance Criterion for optimization (PCO)*, i.e. the Fitness function in the ***Reproduction*** mechanism is based upon *efficiency*. ***Mutation*** was done by the change of the average value of the random jump μ_J to some random value. The reason why we chose to optimize (maximize) the efficiency via μ_J is because there is a dependency of the efficiency criterion on μ_J. For example, if we wanted to optimize (minimize) the *energy* criterion we would have to ***mutate*** either μ_J, ρ or **R**. The main idea is, to make sure that there is correlation between the chosen *performance criterion* and the *control parameter* to be mutated.

Table 4: Algorithm of Actions for the 1st Set of Tests

```
Given: Initial (virtual) population – 10 control strategies
Known: Their five Performance Criteria
for j=1:G (number of generations)
CHOSE the Performance Criterion for optimization, PCO
Reproduction Mechanism (for all strategies):
FIND AVE = average (PCO_Strategy 1 PCO_Strategy 1a ... PCO_Strategy 5a)
for i=1:10
            if PCO of a particular strategy > (<) AVE, name this strategy FIT
            else, name it UNFIT
            end
      end
      COPY the best existing strategy into a new population
      Mutation:
      while the formation of a new population is NOT completed
      {
      MUTATE a particular strategy by randomly changing a specified Control
      Variable Parameter
      }
CALCULATE Performance Criteria of a new population
  end
CHOSE the best-performed strategy from the current generation
```

Results:

For Scenario 1, from the initial population we can see that Strategy 5a is the most efficient one. The number of generations **G** was set to 5. Eventually, original 10 strategies were all replaced by Strategy 5a. In the 5th generation, the algorithm found the value of μ_J with which the efficiency of Strategy 5a increased. In the initial population, the *efficiency* criterion (mean value of 10 runs) of Strategy 5a was found to be **63.84 %** (see Table 3.13) with μ_J = **50**. However, in the 5th generation, with the mutated μ_J = **40.76**, the *efficiency* of Strategy 5a increased to almost **65 %** .

2nd Set of Tests: Using the Reproduction and Mutation Mechanisms only (Scenario 2)

The only difference between this set of tests and the 1st set of tests is in the initial setup (Scenario 2). The general schema and the algorithm of actions are identical to those of the 1st set.

Results:

For Scenario 2, from the initial population we can see that Strategy 4a is the most efficient one. The number of generations **G** was again set to 5. Eventually, original 10 strategies were all replaced by Strategy 4a. In the 5th generation, the algorithm found the value of μ_J with which the efficiency of Strategy 4a increased. In the initial population, the *efficiency* criterion (mean value of 10 runs) of Strategy 4a was found to be **57.94 %** with μ_J = **100**. However, in the 5th generation, with the mutated μ_J = **50.89** the *efficiency* of Strategy 4a increased to **62.09 %** .

Conclusion for the 1st and 2nd Sets of Tests:

From the results of 1st and 2nd sets of tests we conclude that through the sole use of the reproduction and mutation mechanisms we may find the value of the chosen control parameter under which the best-so-far strategy may perform even better.

3rd Set of Tests: Using the Reproduction and Crossover Mechanisms only (Efficiency Fitness Function, Scenario 1)

Below is the general schema for the 3rd set of tests:

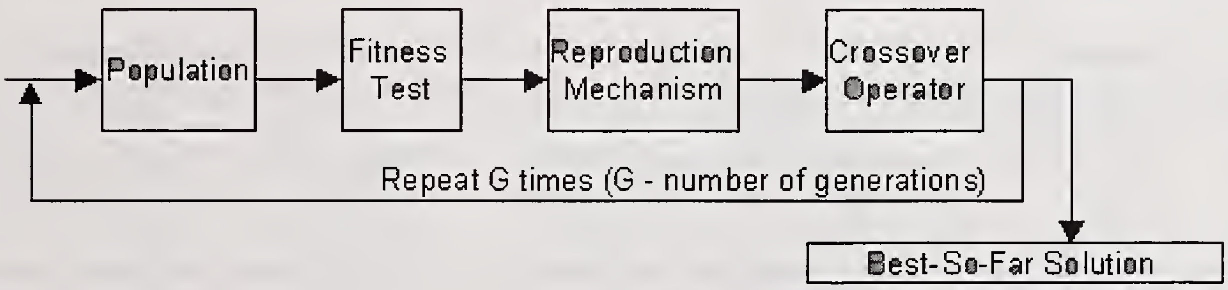


Figure 3: General Schema for the 3rd Set of Tests

This schema is described algorithmically in Table 5. Once again, we chose the *efficiency* criterion to be our ***PCO***, i.e. the Fitness function in the ***Reproduction*** mechanism is based upon *efficiency*. Since we are not changing (*mutating*) any of the control variable parameters, there should be nothing that would affect the **PCO**. The point of performing a crossover is in the fact that when we are pairing FIT parents (e.g. with efficiency above the average), we'll have a higher probability of getting an offspring with better **PCO**. However, by attempting to improve one performance criterion we might inadvertently improve others as well.

Table 5: Algorithm of Actions for the 3rd Set of Tests

Given: *Initial (virtual) population – 10 control strategies*

Known: *Their five Performance Criteria*

for j=1:G (number of generations)

  CHOSE the Performance Criterion for optimization, PCO

  Reproduction Mechanism (for all strategies):

  FIND AVE = average ($PCO_{Strategy\ 1}$ $PCO_{Strategy\ 1a}$ ... $PCO_{Strategy\ 5a}$)

  for *i=1:10*

    if *PCO of a particular strategy > (<) AVE, name this strategy FIT*

    else*, name it UNFIT*

    end

  end

  COPY the best existing strategy into a new population

```
Crossover:
while the formation of a new population is NOT completed
{
Randomly choose two parents out of FIT strategies
Generate a random number P in the [0, 1] interval
if P < 0.6, CROSSOVER and place two children into a new population
else, place the exact copies of parents into a new population
end
}
CALCULATE Performance Criteria of a new population
end
CHOSE the best-performed strategy from the current generation
```

Results:

In Scenario 1, from the initial population we know that Strategy 5a is the most efficient one. The number of generations **G** was set to 2. During the process of crossover Strategies 3a and 4a were chosen for mating. One of their children turned out be highly **efficient**, since it was the **efficiency** that we tried to maximize. The results of this crossover are tabulated below. Table 6 also demonstrates from which parent the child inherited this or that property. Table 7 compares the performance criteria of parents, Strategies 3a and 4a, to those of their offspring, Children 1 and 2.

From these tables one can see that, **efficiency** wise, Child 1 performed extremely well. None of the original 10 strategies, in the same scenario, could ever achieve the efficiency of **73 %** ! However, Child 2 performed quite poor in terms of **efficiency**. Nevertheless, in all of the other aspects, it performed slightly better than one of its parents, Strategy 3a. Thus, we conclude that when optimizing one performance criterion we may also inadvertently improve other criteria as well.

Table 6: 3rd Set of Tests - Results of the Crossover

Strategy	Part 1	Part 2	Control Rules Used	Supplemental Rules Used	Utilized Sensors
3a (1st move is always a jump)	If ΔC_s & $\Delta C_t < 0$, **rotate**	If ΔC_s & $\Delta C_t > 0$, **jump_decrease**, else, **rotate**	1, 2, 3, 4	1, 2	head, tail, belly
4a	**rotate** *n* times and measure all *n* C's; find max C out of *n* C's; **rotate**; find Cnew; while Cnew < max C, **rotate**	**jump_decrease**	3, 4	1, 2	belly

Child 1 of 3a & 4a	**rotate** *n* times and measure all *n* C's; find max C out of *n* C's; **rotate**; find Cnew; while Cnew < max C, **rotate**	If ΔC_s & $\Delta C_t > 0$, **jump_decrease**, else, **rotate**	2, 3, 4	1, 2	head, tail, belly
Child 2 of 3a & 4a (1st move is always a jump – inherited from 3a)	If ΔC_s & $\Delta C_t < 0$, **rotate**	**jump_decrease**	1, 3	1, 2	head, tail, belly

Table 7: 3rd Set of Tests - Parents' Performance vs. Children's With Efficiency Fitness Function

Strategy	**Ave Time of 10 runs / Std Dev**		**Ave Velocity of 10 runs / Std Dev**		**Ave Efficiency of 10 runs / Std Dev**		**Ave Energy of 10 runs / Std Dev**		**Ave Error of 10 runs / Std Dev**	
Parent 1 (3a)	46.05	21.45	13.63	4.52	45.31	14.30	33	13.67	10.32	0.74
Parent 2 (4a)	151.63	40.62	2.97	0.65	55.69	10.08	148.10	39.73	11.20	0.41
Child 1	177.91	67.192	2.02	0.7314	73.35	7.435	165.1	64.578	10.65	0.26
Child 2	41.27	12.49	14.28	2.88	42.91	11.33	26.10	9.87	9.71	0.87

In Figure 4 we compare Strategies 3a and 4a trajectories of motion to those of their "children". It is apparent that Child 1 has the highest efficiency (the thickness of the "tube" is smaller than that of others).

4th Set of Tests: Using the Reproduction and Crossover Mechanisms only (Energy Fitness Function, Scenario 1)

The general schema and the algorithm of actions are the same as in 3rd Set of Tests. For this particular set of tests we chose the *energy* criterion to be our ***PCO***, i.e. the Fitness function in the ***Reproduction*** mechanism is based upon *energy*.

Results:

For Scenario 1, from the initial population (Table 13) we know that Strategy 5a is the most efficient one. The number of generations **G** was set to 2. During the process of crossover Strategies 1a and 2a were chosen for mating. Their children turned out to be more *energy* efficient than one of their parents (remember it was the *energy* performance criterion that we tried to minimize).
The results of this crossover are tabulated below:

The comparison of performance criteria of parents, Strategies 1a and 2a, to those of their offspring, Children 1 and 2 are collected in the table:
From another table one can see that, *energy* wise, both children performed better than Parent 2 (Strategy 2a). Also, the *efficiency* criterion for both children is a lot better than that of Strategy 2a. Thus, we come to the same conclusion (see results for the 3rd Set of tests) again that when optimizing one performance criterion we can also unconsciously improve other criteria as well.

Figure 5 compares the parents' trajectories of motion to those of their offspring. Visually, it is difficult to make any sort of conclusion about strategies' performances. Even though the "tube" of trajectories for Child 2 seems to be narrower, numerically, Strategy 1a has the highest efficiency.

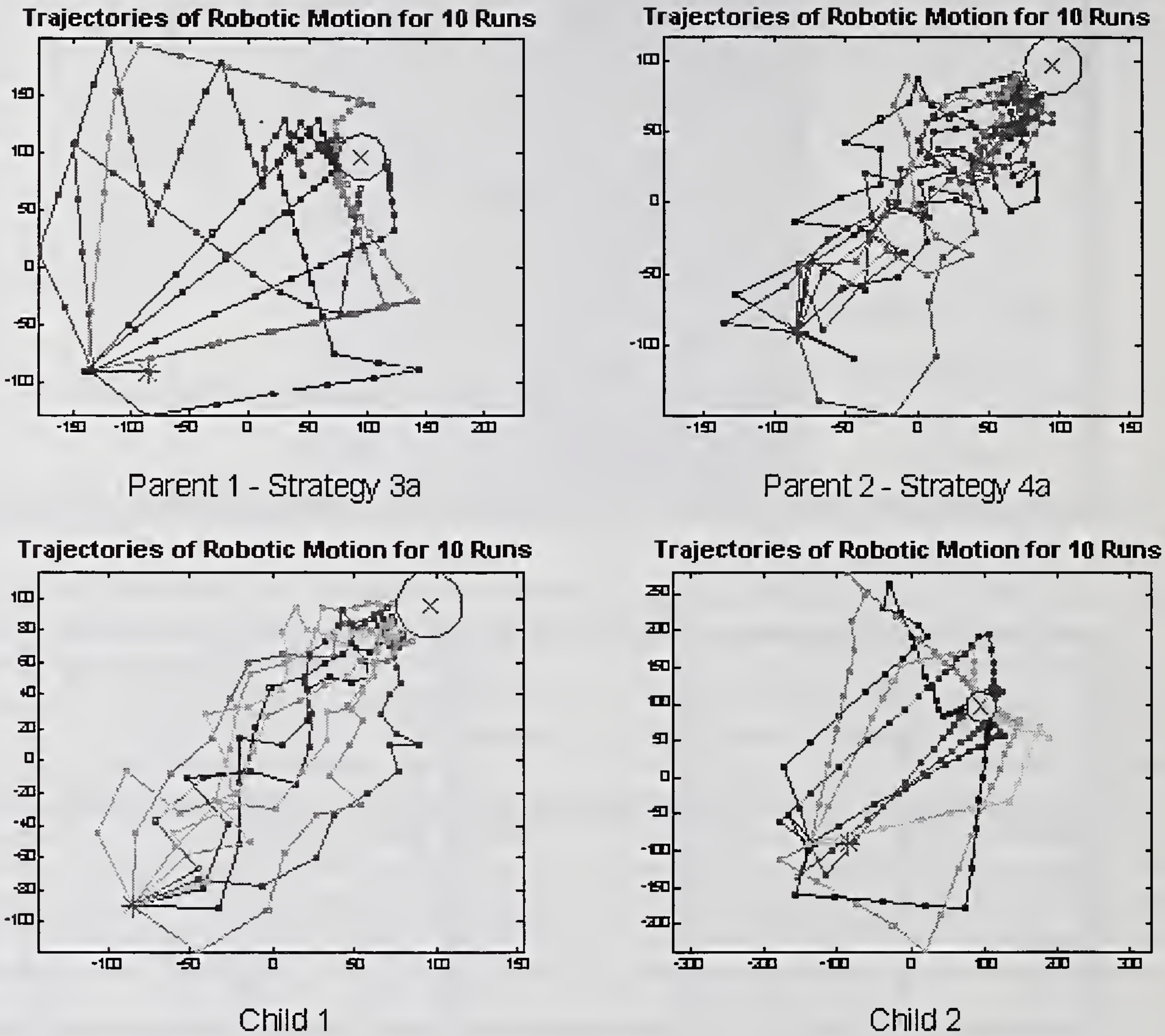


Figure 4: 3rd Set of Tests - Trajectories of Robotic Motion for Strategies 3a, 4a, and their Children

Table 8: 4th Set of Tests - Results of the Crossover

Strategy	Part 1	Part 2	Control Rules Used	Supplemental Rules Used	Utilized Sensors
1a	If $\Delta C_s < 0$, rotate	If $\Delta C_s > 0$, jump_decrease	1, 2	1, 2	head, tail
2a (1st move is always a jump)	If $\Delta C_t < 0$, rotate	If $\Delta C_t > 0$, jump_decrease	3, 4	1, 2	belly

Child 1 of 1a & 2a (1st move is always a jump)	If $\Delta C_s < 0$, **rotate**	If $\Delta C_t > 0$, **jump_decrease** (else, **rotate** – if neither of conditions is met—an additional rule we had to introduce)	1, 4	1, 2	head, tail, belly
Child 2 of 1a & 2a (1st move is always a jump)	If $\Delta C_t < 0$, **rotate**	If $\Delta C_s > 0$, **jump_decrease** (else, **rotate** – if neither of conditions is met—an additional rule we had to introduce)	2, 3	1, 2	head, tail, belly

Table 9: 4th Set of Tests - Parents' Performance vs. Children's With Energy Fitness Function

Strategy	**Ave Time of 10 runs / Std Dev**		**Ave Velocity of 10 runs / Std Dev**		**Ave Efficiency of 10 runs / Std Dev**		**Ave Energy of 10 runs / Std Dev**		**Ave Error of 10 runs / Std Dev**	
Parent 1 (1a)	32.73	9.30	12.64	1.36	60.8	17.74	31.9	9.10	10.58	0.74
Parent 2 (2a)	36.1	15.18	21.88	4.29	33.96	10.64	35.2	14.78	10.09	0.52
Child 1	35.97	16.67	18.58	4.68	42.93	16.58	34.60	16.19	10.54	0.45
Child 2	34.73	9.92	14.33	3.40	50.95	11.69	33.40	9.56	10.50	1.05

Conclusion for the 3rd and 4th Sets of Tests:

From the results of 3rd and 4th sets of tests we conclude that through the sole use of the reproduction and crossover mechanisms we may find new strategies that perform better than their parents or at least one of the parents.

Operation of the Genetically Programmed VSC

Combining results from the four sets of tests analyzed above, we came up with the following design of our **Variable Structure Controller**:

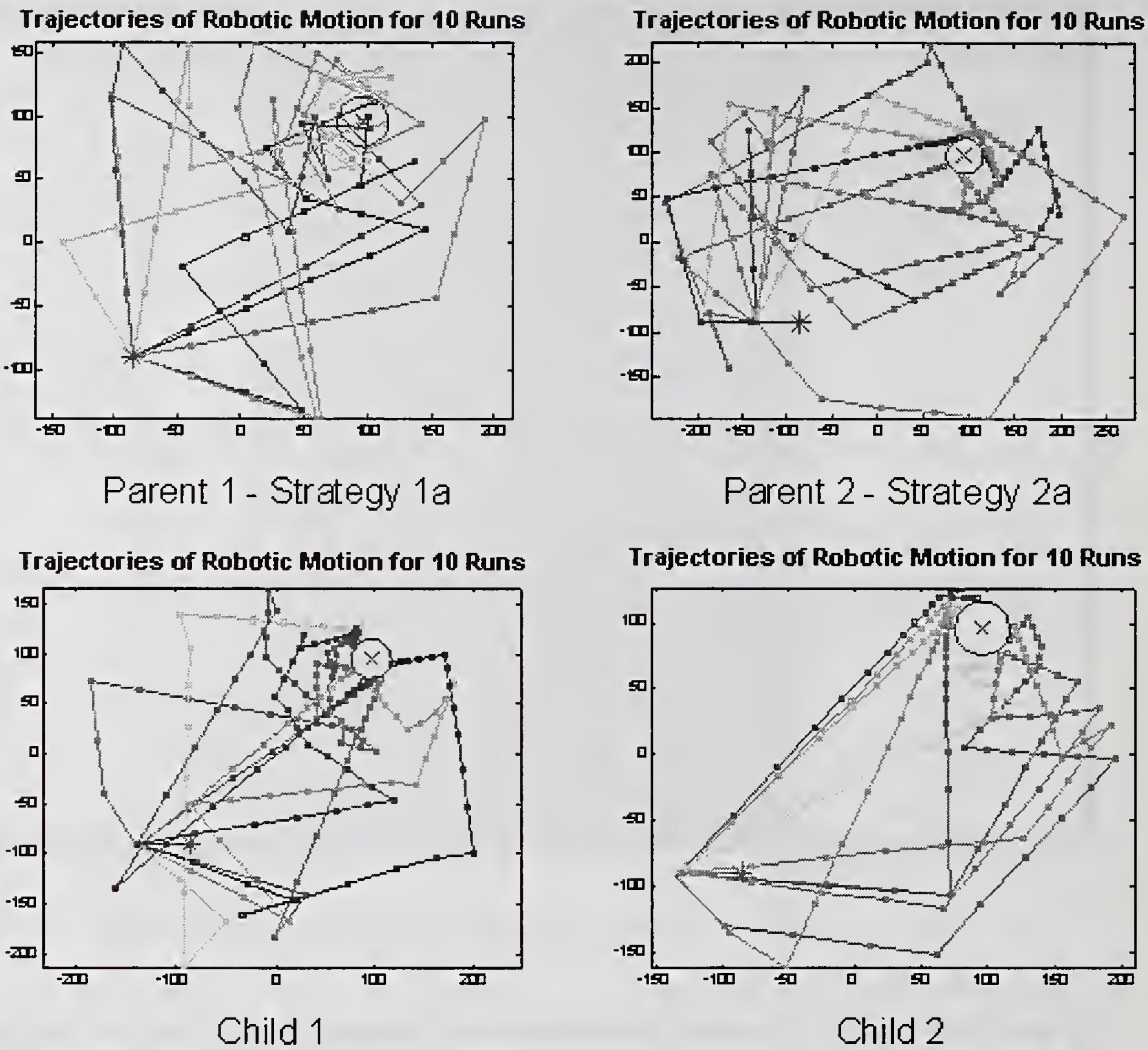


Figure 5: 4th Set of Tests - Trajectories of Robotic Motion for Strategies 1a, 2a, and their Children

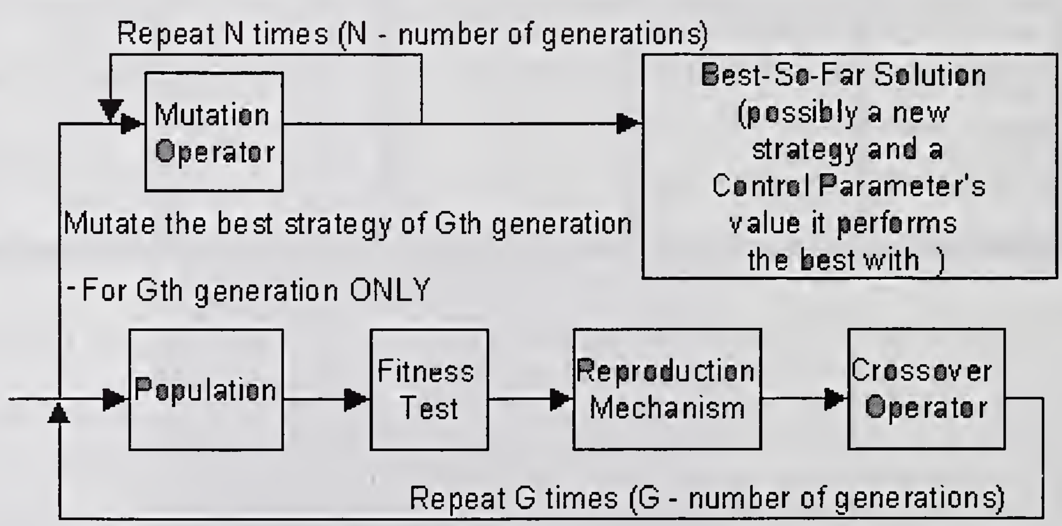


Figure 6: Variable Structure Controller's General Schema

Generally, **VSC** does the following:

- Uses the Reproduction and Crossover mechanisms for a **G** number of generations.
- It may create a new strategy that performs better than its parents or at least one of its parents. If a new strategy is created, it's placed into a new population.
- In $\mathbf{G}^{th}$ generation it chooses the best performed strategy and mutates it **N** number of times by changing some specified Control Variable Parameter to a random value.
- Outputs an IMPROVED solution in terms of the best-performed strategy and the Control Variable Parameter's value it performs the best with.

Also, we believe that if we let our controller vary the fitness function from generation to generation, it might be able to come up with a strategy that will have an improvement along more than one *performance criterion*. Below, we will describe the operation of our **VSC** algorithmically:

Table 10: Pseudocode of the VSC's Operation

Given: *Initial (virtual) population – 10 control strategies*
Known: *Their five Performance Criteria*
for *j=1:G (number of generations)*
CHOSE the Performance Criterion for optimization, PCO
Reproduction Mechanism (for all strategies):
FIND AVE = average $(PCO_{Strategy\ 1}\ PCO_{Strategy\ 1a} \ldots PCO_{Strategy\ 5a})$
for *i=1:10*
if *PCO of a particular strategy > (<) AVE, name this strategy FIT*
else, name it UNFIT
end
end
COPY the best existing strategy into a new population
Crossover:
while *the formation of a new population is NOT completed*
{
Randomly choose two parents out of FIT strategies
Generate a random number P in the [0, 1] interval
if *P < 0.6, CROSSOVER and place two children into a new population*
else, place the exact copies of parents into a new population
end
}
CALCULATE Performance Criteria of a new population
end
CHOSE the best-performed strategy from the current generation
Mutation:
for *k=1:N* (number of generations)
MUTATE the best-performed strategy by randomly changing a specified Control Variable Parameter
CALCULATE Performance Criteria of a mutated strategy

end

RETAIN the value of a mutated Control Variable Parameter under which the best-performed strategy performs even better

In essence, our **VSC** not only can create new strategies, it can also determine under which value of the specified **Control Variable Parameter** they perform the best.

Conclusions and Recommendations

In this paper, the following three major goals were pursued:

- To study a behavior of a real *E. coli* bacterium
- To synthesize robotic control strategies that are both efficient and robust based on the observations of *E. coli*'s behavior
- To design a robotic controller that would presume a creation of a very broad scope of logically compatible combinations of control rules comprising the earlier developed control strategies

It is worth mentioning that out of our 10 designed control strategies Strategy 2 emulates the behavior of a real *E. coli* bacterium the best, even though it is not the most robust strategy. In the figure below we compare the behavior of our robot implementing Strategy 2 to that of a real *E. coli* bacterium in a nearly isotropic homogenous medium:

The decision-making mechanism of an *E. coli* cell helped us design 10 robust control strategies. This led to the creation of a variable structure controller (**VSC**) that not only can create new strategies all on its own, but can also determine under which value of the specified **Control Variable Parameter** they perform the best.

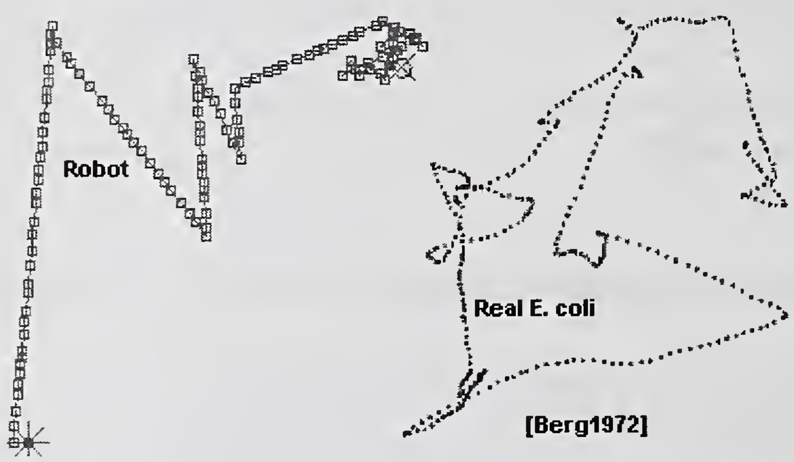


Figure 7: Robotic Trajectory of Motion (Strategy 2) vs. Real *E. coli* Bacterium's Trajectory of Motion

References

1. Andrusenko, Julia, *Biologically Inspired Variable Structure Controller (VSC) for an Autonomous Robot*, MS Thesis, Drexel University, 2001
2. Fernandez, Jaime, The GP Tutorial, last updated: June 03, 2000 http://www.geneticprogramming.com/Tutorial/index.html
3. Grefenstette, John J., *Learning Decision Strategies with Genetic Algorithms*, Naval Research Laboratory, Washington, DC, 2000
4. Aha, David W., *Tutorial on Machine Learning, 1995 AI & Statistics Workshop*, Ft. Lauderdale, FL, Jan. 1995
5. Pal, Sankar K., Wang, Paul P., *Genetic Algorithms for Pattern Recognition*, Boca Raton, FL: CRC Press, Inc., 1996
6. Koza, John R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge, MA: The MIT Press, 1992

Appendix 1

Control Strategies

Strategy	Part 1	Part 2	Control Rules Used	Supplemental Rules Used	Utilized Sensors
1	If $\Delta C_s < 0$, **rotate**	If $\Delta C_s > 0$, **jump**	1, 2	1	head, tail
1a	If $\Delta C_s < 0$, **rotate**	If $\Delta C_s > 0$, **jump_decrease**	1, 2	1, 2	head, tail
2 (1st move is always a jump)	If $\Delta C_t < 0$, **rotate**	If $\Delta C_t > 0$, **jump**	3, 4	1	belly
2a (1st move is always a jump)	If $\Delta C_t < 0$, **rotate**	If $\Delta C_t > 0$, **jump_decrease**	3, 4	1, 2	belly
3 (1st move is always a jump)	If ΔC_s & $\Delta C_t < 0$, **rotate**	If ΔC_s & $\Delta C_t > 0$, **jump**, else, **rotate**	1, 2, 3, 4	1	head, tail, belly
3a (1st move is always a jump)	If ΔC_s & $\Delta C_t < 0$, **rotate**	If ΔC_s & $\Delta C_t > 0$, **jump_decrease**, else, **rotate**	1, 2, 3, 4	1, 2	head, tail, belly
4	**rotate** ***n*** times and measure all ***n*** C's; find max C out of ***n*** C's; **rotate**; find Cnew; while Cnew < max C, **rotate**	**jump**	3, 4	1	belly
4a	**rotate** ***n*** times and measure all ***n*** C's; find max C out of ***n*** C's; **rotate**; find Cnew; while Cnew < max C, **rotate**	**jump_decrease**	3, 4	1, 2	belly
5	If $\Delta C_s < 0$, **rotate**	If $\Delta C_s > 0$, **jump**, **rotate**	1, 2	1	head, tail
5a	If $\Delta C_s < 0$, **rotate**	If $\Delta C_s > 0$, **jump_decrease**, **rotate**	1, 2	1, 2	head, tail

Appendix 2

Performance criteria

Introduction of Performance Criteria

The performance criteria (for a single run) of our 10 strategies are defined as follows:

Time, t (sec) – total time it takes to complete a single run

Velocity, V (units/sec) – overall velocity, defined as a total distance traveled, $\mathbf{D_{total}}$, over total time: $\mathbf{V = D_{total} / t}$

Efficiency, ε (%) – Euclidean (shortest) distance, $\mathbf{D_{EUC}}$, over total distance traveled: $\mathbf{\varepsilon = [\, D_{EUC} / D_{total}\,] * 100\ \%}$. $\mathbf{D_{EUC}}$ is the distance between initial position of our robot's tail and the sugar point. For instance, for the scenario that we chose (Table 3.2), $\mathbf{D_{EUC}}$ = 232.03 units of length. The reason why we are finding distance between the robot's tail and the sugar point instead of the one between the robot's belly and the sugar point is because of the fact that our

Energy, E (elementary moves) – energy in this thesis is defined as a total number of elementary moves (jumps and rotations). It is assumed that both **JUMP** and **ROTATION** have a unit of energy.

Error, Err (% from $\mathbf{D_{EUC}}$) – error of arrival to the goal. When $\mathbf{D_{EUC}}$ is calculated there is a need to compensate for the error of arrival to the goal. Due to the fact that it would be quite difficult for the *E. coli* robot to find a single (sugar) point, we introduced a **Stopping Rule** with its circle of radius **R** around the sugar point. Introduction of this so-called circular "sugar vicinity" also introduces an error of arrival to the goal. To compensate for that we do the following:

$$\mathbf{D_{EUC} - (h / 2 + R)},$$

where **h** is the height or length of our robot and **h /2 + R** quantity represents the maximum **Err** possible in units of length. To elaborate on what we mean by the maximum error possible we present the picture below:

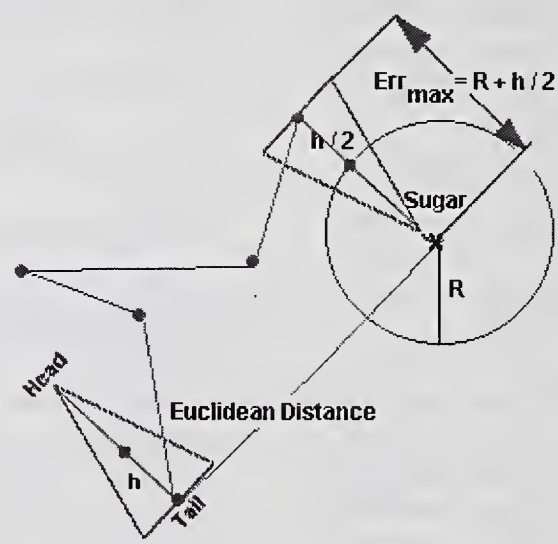


Figure A : Depiction of the Robot's Stop in the Sugar Vicinity when the Error (in units of length) of Arrival to the Goal is Maximum

Remember that the robot stops if the distance between its belly and sugar point is less or equal to **R**. Thus, the $\mathbf{Err_{max} = R + h / 2}$ since we are calculating distances from the robot's tail and not its belly.

PART III
RESEARCH PAPERS

3. Performance Evaluation in Non-numerical Domain

Measuring the Impact of Information on Complex Systems

Larry H. Reeker
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, Maryland
larry.reeker@nist.gov

Albert T. Jones
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
Gaithersburg, Maryland
albert.jones@nist.gov

Abstract

The application of power-driven machinery to manufacturing and other areas of human endeavor characterized the Industrial Revolution in the 18th and 19th centuries. Measurement contributed in many ways to the increasing economic influence of these machines. Using formal or informal physical principles, metrics and measurement techniques were found that allowed the comparison of machine performance (evaluation), the development of machines with the needed qualities (engineering), and the coordination of machines within factories (integration). The required physical dimensions were space, time, and mass, and the common physical quantities derived from these three; and, for these quantities, measurement techniques were established. In the Information Revolution begun in the 20th Century, measuring information is also vital to the continued influence of machines. Unfortunately, information is not as well understood, as are physical constructs. It seems to have an unlimited number of dimensions, and no generally accepted metrics or measurement procedures. So how do we measure the impact of information in the 21st Century? This paper sketches research directions that may help to answer this question and it stresses the importance of obtaining an answer.

I. Introduction

The machines or systems (machine and system will be used as synonyms) to which we will refer in this paper are ones in which information is vitally necessary and for which information affects the behavior. Our use of "behavior" is not limited to input and output, not a black box definition. There is information coming into, going out of, and residing within a system that is essential to both its internal and external behavior. So machines have a physical aspect, but it is the informational one that will be stressed here. Of particular interest is manufacturing, where systems have practical importance plus high complexity; but the same problems arise in all information domains.

Information must be conveyed in physical symbols like marks on paper, sounds, or electrical pulses. Nevertheless, information has an effect on a system that is not explainable by its physical properties alone. That effect is related to (1) the *organization* of the symbols, (2) the *meaning* ascribed to the symbols and their organization, and (3) the change in the system *state* that comes from understanding and acting on that meaning. Since the state varies with time, so will the effect of a particular item of information on the properties of a given system.

There are three practical reasons for measuring these properties. First, they are useful to evaluate systems successfully. Measurements are needed to compare one system to another, to show that they meet a particular need, to prove that they fulfill the specifications of a prior agreement, to demonstrate that they conform to standards, and so on. Second, they are necessary to engineer systems successfully. Measurements are needed to ensure that constraints in the building process are met and that the system will behave in the required way. As an extension of the process of building to the practical need for modularity, they are required to integrate systems successfully. Measurements are needed to verify that the information needed by one system can be supplied by others without error and on time. The importance of measurement can thus be based in the three roles for measurement: *evaluation*, *engineering,* and *integration*. There are other reasons, too, that might be cited, such as *understanding* the system; but they can be seen generally as overlapping the three practical reasons[1].

In the earliest applications of information that impacted the performance of systems, the physical carrier of the information was mechanical links in steam engine governors, punched holes in Jacquard looms, or electrical connectivity in thermostats. The impacts of the information could be measured for these applications by its physical properties, and its physical cause and effect (as heat causing expansion of a certain amount or current

[1] An appendix is attached that discusses some meta-level aspects of the measurement, with respect to science and engineering.

flow in a thermocouple), reaction times, and so on. Those impacts could be quantified, therefore, in terms of system performance. The *meaning* of the information, not its *representation*, was what influenced that performance. The punched holes in the loom cards were originally in stiff pasteboard and were read by needles. After their evolution into Hollerith's paper cards, they could be read by pins that conveyed electricity and later by light and electricity. Finally, when the cards went away entirely in favor of other information representations, the same information could be conveyed by different physical means. Thus, the performance of many physical systems is, in some sense, independent of the physical form of the information that drives them.

The complexity of systems has evolved considerably over the past twenty years. Among the more complex systems are what we often call "intelligent systems". In these systems,[2] the impact of the informational component, its representation and its meaning, is paramount. It is clear that testing for the amount of and the impact of information in any particular area is going to be difficult, and that even the terms "amount" and "impact" will be difficult to define. In short, a metric of the information abstracted from the physical parameters is not evident. The paper will argue that a great number of such systems exist, even outside the area that might be labeled "intelligent" or "knowledge based". It stresses several critical points to understanding and controlling such systems.

II. Impact of Information on Control of Complex Systems

Two of the simple systems mentioned above as examples – the thermostat and the governor – are ones in which the information is gathered by feedback, which is the collection of information, its representation in a physical medium, and its interpretation to control a system. The handling of information for control can be much less direct. Yet it is often the case that simple models can provide ideas that can be generalized to more complex ones, and maybe it can help in this case to understand the general problem of measuring information impact.

Many complex systems can be viewed as a collection of integrated and layered subsystems, which might at some "bottom" layer be cases of direct physical control. Typically, the layering occurs in both the temporal domain and the spatial domain. The bottom layer contains some combination of biological, chemical, and physical processes. In humans, the evolution of these processes is governed by an internal and natural intelligence, which we call the mind. While we do not know exactly how it works, we see its benefits every moment of our lives.

Man-made systems, on the other hand, are not endowed with a mind. The processes that make up these systems are subject to the second law of thermodynamics. Hence, without any external intelligence to guide their evolution, entropy will increase and they will go out of control over time. To keep this from happening, researchers have expended an enormous amount of time, energy, and money to develop models, algorithms, and heuristics that come under the general heading of control theory.

While it is conceptually simple, control theory can be complicated in practice. Conceptually, it consists of two steps. Step 1 is to set the desired goal and develop a plan to achieve that goal. Step 2 is to observe the execution of that plan and make adjustments as required. The first step usually involves the development of a model of the system, an optimization problem based on that model, and technique to solve that problem. Models, which can be continuous or discrete, and deterministic or stochastic, typically have temporal and spatial parameters. The optimization problem has at least one measurable, quantitative goal and constrains the parameters in the model. Sometimes these problems can be solved analytically, sometimes not. Regardless of how the solution is derived, it results in a plan to be executed by the system.

Consider a robot that that must move a part from point A to point B in the shortest possible time. To generate a path to accomplish this goal, the robot controller, which could be a human or a software procedure, needs models of the robot and its environment. These models are continuous time, continuous state, and deterministic. The controller formulates an optimization problem whose solution will specify the start coordinates, the end coordinates, the time, limits on models parameters (such as speed, joint angles, and so on), and possible obstacles to avoid. That solution yields the optimal plan that the robot should use. This plan is then sent to the robot, or more accurately the execution part of its controller, to be implemented. Once the robot begins to move, we must proceed to step 2. This means that we must somehow make sure that the robot does not exceed any of the limits and follows the predetermined path. We do this through the generation and analysis of feedback. Sensors on the robot create the feedback, which is analyzed by the controller. When a problem is detected, a new plan will be generated.

CRITICAL POINT 1: Both the plan and the feedback are information objects, which impact the performance and the behavior of the robot. Some of these objects are simple; some are not. The meaning of these objects must be conveyed to and understood by

[2] This paper used the term "complex" interchangeably with "intelligent", to avoid defining the latter term, the difference not being important for our purposes.

all hardware and software components or there is no hope of achieving the desired goals. These capabilities do not happen "naturally"; they must be built into the system.

As we move up the layers, we no longer deal directly with biological, chemical, and physical systems. Instead, we deal with decision-making and information systems that affect those bottom-layers, but on a longer-term basis. Nevertheless, the same two steps are involved. In this case, however, the models are discrete time and discrete state systems that often contain one or more stochastic parameters. There are several, often conflicting, quantitative performance measures and the techniques are implemented in a number of software applications such as linear programming, demand forecasting, and supply chain management. These applications also produce plans that are implemented in other, lower-layer software applications -- demands lead to production plans, which lead to schedules, which lead to sequences and so on. These plans are based on information that has a high degree of uncertainty. Some of this uncertainty arises because of the influence of the second law on the bottom-layer processes. Some of it arises because of the stochastic nature of predictions associated with demand projections, priority orders, and material arrivals, to name a few.

CRITICAL POINT 2: Optimizing high-level performance measures is critically dependent on the ability of the associated software applications to share complex information objects. Furthermore, without have a common understanding of the meaning of those objects, optimization is useless.

As we progress through the various layers of a complex system like a manufacturing enterprise, an evolution occurs from continuous time to discrete time and from continuous state to discrete state. Furthermore, an aggregation in information takes place as well - very detailed, relatively simple, deterministic information at the bottom; very little detail, more complex, highly stochastic information at the top. No one knows how this evolution or aggregation takes place. Moreover, at every layer, there is some influence of entropy from both the second law and information uncertainty. At the bottom, the second law dominates. At the top, information uncertainty dominates. We have a very good idea of how to measure and control the effects of the second law on physical system performance. We have almost no idea how to measure and control the effects of information on performance.

CRITICAL POINT 3: Information has a large impact on system performance. Integration, getting the right information from one software application to another, also has an impact. Consequently, ensuring that all software applications have the same understanding of that information is critical to system performance. Furthermore, and most importantly, our ability to measure how well they understand impacts directly our ability to measure the true performance of the system.

An important question then is how can we build software applications that are capable of understanding information. The simple answer is that we must make software, just as we must make equipment, more intelligent. More accurately, we must surround each software application with the "stuff" it needs to understand the information it receives from other applications. A partial list of some of that "stuff" includes:

Parsers to determine the structure of an encoding (the physical representation) according to a known structural description (for symbols, called a "grammar").

Ontologies to describe the internal model that the system can use to recognize inputs in terms of catalogues of entities and processes and their relationships.

Dictionaries to define the relationship of discrete elements of the encoding to objects and processes in the ontology.

Mappers from encodings to models or directly from one model to another.

Controller which makes decisions on a course of action (a sequence of behaviors), based on information in plans that have been preprogrammed or formulated and inputs (from users or sensors, including feedback), and operates actuators to cause the behavior sequence.

Actuators: Devices which behave physically to produce behavior.

Perceptors: Systems that convert the input of sensors into information for the system to process.

Equivalence, Similarity, and Difference Metrics: Ways of measuring how the information in one system or subsystem relates to another – whether it is equivalent or not (more on this below!)

CRITICAL POINT 4: Our ability to control the performance of the physical systems can depend directly on our ability to measure the similarities and differences between information objects.

III. Measuring Equivalence between Information Objects

Perhaps the first thing to consider in looking at measurement metrics is whether definitions of equivalence can be established. This is a tricky issue, because in some sense they cannot. Consider two ontologies, as defined above. They conventionally are represented by classes of entities and their attributes, linked into hierarchies (lattices are mathematically one representation) based on the IS-A relationship. IS-A relationships are based on the attributes of classes of entities, and those attributes are based two things: fundamental properties and the behaviors of entities in activities. Trying to compare behaviors of entities after a certain degree of complexity is reached leads to things like the halting problem. Thus, just as it may be formally undecidable if two programs are equivalent, it may be difficult to determine ontological equivalence formally. Perhaps we can still get measurements that will enable satisfactory performance within bounds, and undecidability will not be a problem. We still need to measure some concepts of equivalence, even with the blanket restriction of undecidability, which is a common restriction that must be sidestepped often in computing. One approach is to use approximations, which are often required by limited measurement precision anyway.

CRITICAL POINT 5: The equivalence of information objects may be undecidable, but we may be able to develop approximate measurements.

Developing an approximate equivalence metric puts us right in the middle of an ongoing controversy. That controversy revolves around the best way to represent uncertainty in information. There are two views: probabilistic and fuzzy. The probability proponents argue that there is only one consistent way to measure uncertainty and that is probability theory. They further argue that all probability is conditioned upon prior information and that the proper way to do inferencing must be based on a Bayesian framework. That framework says (1) create a prior distribution using the Principle of Maximum Entropy, (2) update that distribution using any new information and Bayes theorem, and, (3) use this new distribution for inferencing [Jaynes, 88].

The fuzzy proponents argue that information is not crisp enough to be measured using the quantitative laws of probability. To overcome this difficulty, the concept of a membership function is used. It has yet to be determined for many researchers if there is any essential difference between using fuzzy information and exact numbers with probabilistic error bounds. At this point, many people agree that fuzzy information can be a useful concept for engineering systems and simplifying the code that runs those systems. It may turn out that it is a mathematical difference analogous to that between matrix and wave mechanics in physics.

CRITICAL POINT 6: A full understanding of the relationship between various approximate ways of measuring information is needed.

Another important issue related to measuring uncertainty in information objects is the notion of entropy. That there is a relationship between information and entropy has been postulated for many years. A number of information measures have been proposed [Arndt, 01], including those by Shannon [Shannon and Weaver, 71] and Stonier [Stonier, 91]. Information is a measure of the decrease of uncertainty, and its representation requires an organized notation. Entropy is a measure of the increase of randomness. If one takes an organized body of information and randomizes it (adds noise) then there is less information and higher entropy.

The term "information" is itself used in different ways, however, because organization can mean many things. In thermodynamics, it is molecules behaving in an organized fashion. In Shannon's communication examples, it is strings of symbols sent from a sender arranged in a way that can potentially lessen uncertainty at a receiver that can decode the symbols. In other uses, however, information has to be relevant to some task being performed by a system. In computation, it is related to complexity considerations. The work of Solomonoff, Kolmogoroff and Chaitin [Chaitin, 92] links information conveyed by symbols in logic and information systems with complexity of computation, and relates them to Shannon's measures, as well.

The problem of information content is that it is "about *something*". How do we compare information about two different subjects? The answer may be that we just do not do so, at least if the subjects are independent. But how do we know if they are independent? We may not want to mix oranges and apples; but, if we are concerned about fruit, we can develop information about them because they are no longer independent. Consider the following simple experiment. Suppose we have nine pieces of fruit, five oranges and four apples, and someone puts three of them in a bag. If we find three apples, we know that there are no oranges. This becomes much more difficult when we get to questionable or fuzzy sets -- try repeating this experiment with five big apples and four small apples. This second experiment is typical of problem of measuring information content. It depends on the individual system and its ontology. Independence can be classified as being in different dimensions, analogous to dimensions in physics; but it seems there are too many dimensions to measure.

In the example above "fruitiness" might be considered an attribute and the question might be whether a tomato has some fruitiness, so a negotiation is needed to decide if it will count or not. The Garden of Eden "fruit" is generally considered to be an apple. Could it be an orange? Is an apple "fruitier" or more likely to be fruity than an orange? Reasoning like this would call for a lot of dimensions, since apples and oranges alone have plenty of attributes to be compared. The psychologist and communication scholar Charles E. Osgood developed work in the measurement of a type of meaning (which is information content in much the same way that work is energy; meaning changes information content).

Osgood was interested in connotative meaning – meaning that is related to an individual's personal ontology [Osgood, 57]. So, it is beyond the denotational meaning and only intended to be partial. In trying to define it, he postulated three dimension types or factors, within which pairs of adjectives would indicate denotations.

- Evaluative factor (example: good - bad)
- Potency factor (example: strong – weak)
- Activity factor (example: active - passive)

Osgood then measured each pair, for each factor, on a seven point Likert scale. In the apple example, perhaps "fruity" could be equated to one and "not fruity" could be equated to seven. He then constructed an n-dimensional space, n being the number of adjective pairs, for his "semantic differential".

Clearly, much more than the semantic differential is needed to do the evaluation that can lead to integration of several manufacturing systems or bioinformatics systems. However, Osgood's ideas fit into the idea of fuzzy frameworks, and it was an important step in trying to formalize the idea of how the vocabulary of humans may vary. Vocabulary, while not the same as ontology, is closely linked, and provides a way to get at human ontologies. With machines where we know the code, we have the advantage of being able to read the ontologies more directly. The problem then becomes one of developing mappings from one ontology to another.

How do we reconcile the measures mentioned above with a system of dimensions like those used to measure physical dimensions, and how many dimensions do we actually have in information?

CRITICAL POINT 7: Even a theory that defines information not just as to amount but also in terms of "vectors" of information does not so far seem adequate for computing information equivalence or providing a precise measure of information overlap, though it is an interesting approach.

There are other problems of terminology that will not be discussed. It is rare to hear people use "data", "information", and "knowledge" in consistent, well-defined ways. And what about "potential information" that has a statistical amount but is not used at all? All of these still need some standard definition and scientific theories to put them in a framework. The underlying theory is not adequate. On the other hand, perhaps a limited categorization of knowledge that would cover particular programs is possible, if the categorization can be agreed upon. Here we return to the notion of ontology. This is a claim that sums up some of the ideas herein:

CRITICAL POINT 8: What we need to measure defines the model of the world that a system expects to find and its ways of coping with that world. The set of its behaviors may be infinite and unknowable if the machine is complex, but it is defined indirectly if we can predict behavior through the model. To predict behavior accurately, the information needs to be characterized in an ontology and what is done to information based on that ontology.

This point would seem to suggest an arduous task for satisfactory measurement of the relevant information that flows through a system; but it also suggest a potential benefit. Measurement of the information, if it is adequately powerful, can potentially "give back" to us in understanding enough value to repay the effort we put into creating and applying the metrics and techniques.

It is clear that every system that deals with information, whether computational or biological, contains an internal model of the outside world – an ontology. In computer programs, the elements of the ontology are data objects and procedures for manipulating those data objects; both are used by the program. Like matter and energy in physics, data objects and procedures in an ontology are related. Consider the notion of an ordered list of customers. It is a data object; yet it can be defined by a random set of customers and a sorting procedure. Its inputs are the (unordered) list and a statement of the type of order desired; and its output is the ordered list. If we want to integrate two systems that need an ordered list of customers for some purpose, it is important that we be confident that they employ the same order; otherwise they will not correctly operate together. To do that, it is necessary to measure the ontologies of the two systems to see if that is true.

The order example is not very complex on its surface. In practice, it is not possible, in general, to find out whether two algorithms producing an arbitrary order (not just a simple linear one) are outputting the same information. This is a consequence of a variety of undecidability results. So it is necessary to consider how we can use standards and measurements to be relatively confident that there is going to be interoperability between the two systems.

IV. More on Measuring, Comparison, and Characterization of Ontologies

The point has been made above that a core ontology is needed to specify what information can be communicated to a given system by another system and what information the given system can send back. There is work going on in measuring these ontologies.

Every ontology has certain terms that may be grounded in physical parameters. In these cases, the physical information needs to be expressed in the appropriate dimensions. We are all familiar with the problem that arose in a probe of the planet Mars when the input and the expected physical parameters had different dimensions. That problem was not unique, and is even common in the building of software systems that do not have well-engineered descriptions of requirements. It is just easier to recognize when the information is closely related to physical parameters, as in the Mars probe. This also happens when the output from physical sensors is used as input to software applications. One tests the input requirements to see if the sensor outputs match exactly or in a way that is 'mappable' at the information level. Ontologies can help in both the matching and the mapping.

The development of explicit ontologies, therefore, is itself an important step because it clarifies which information items are directly grounded and which are indirectly grounded through computations. Comparing directly grounded objects is, in general, easier than comparing indeirectly grounded objects. Even if the frameworks for the ontologies are different, they can be compared if they use a consistent style. [Noy and Hafner, 97] characterized and compared a number of different ontologies. They concluded that if the ontologies can then be mapped into a similar format, they may be aligned, and maybe merged, with perhaps some human interaction. [Noy and Musen, 00] discusses this for a system called PROMPT; [McGuinness *et al*, 00] discusses an environment that provides tools for people who wish to merge ontologies.

Three efforts are underway to develop some standards for ontologies related to manufacturing: the Standard Upper Ontology, SUO, [http://suo.ieee.org/], the Process Specification Language, PSL, [http://www.mel.nist.gov/psl/], and, the Defense Agency Markup Language, DAML, [http://www.daml.org/] . These can be helpful in that they make the comparison of systems with different ontologies easier. How does each deviate from the core ontology? If the top (more general) ontological categories are the same, that saves a lot of work; and if they are not entirely the same, it may be easier to compare them to a single, core ontology and note their deviations. But there will still be systems with different ontologies in overlapping subject areas that need to be merged. A recent paper on how these may be compared is found in [Maedche and Staab, 01], who provide some explicit measures of similarity.

The goal of determining the properties of ontologies by analyzing the information in them and then comparing them to ontologies of other systems for interoperability purposes is still some distance away. Nevertheless, the interest in the area is growing and the results are promising.

V. Implications for Computer Science and Software Engineering

Each programming language must provide means of instructing a machine *how* to process information. This can be done implicitly, through logic or objects, or explicitly, through commands and procedure calls. Equally, a language must be able to convey knowledge of *what* information it is dealing with. This fact is encapsulated in the title of Nicklaus Wirth's book *Algorithms + Data Structures = Programs* [Wirth, 1976]. The history of programming languages shows that there is a tradeoff between describing the *how* and *what.* The tradeoff is illustrated by comparing object-oriented languages with procedural languages. In the SIMULA language, the first object-oriented language, is both procedural and object-oriented, but a glance at the programs in, say, *SIMULA Begin* [Birtwhistle *et al* 73] , illustrates the tradeoff.

The data structure is a fundamental part of a programming language. There has been descriptive work on data structures, and everybody has examples of "informationally equivalent" data structures. As a simple example, consider character strings. Before they were basic structures in some languages, they were handled as an array of characters and numbers with the number indicating the array address of the next character. Note that one of these has both single characters and integers, while the other one has only character strings. Despite the difference in structure, it is not difficult to show the informational equivalence of these two representations. In fact, the idea of *data abstraction* deals with using such equivalences to free the programmer from details of implementation. Though it is not the purpose of this paper to deal with programming *per se*, the idea of comparing ontologies is, in fact, similar to comparing two implementations that have different data structures. One first asks if the data structures are equivalent. If so, their *syntax*, the physical organization by which they are communicated by the programmer to the computer or stored in the computer is equivalent. The next question is "Do they mean the same thing?" Another way to ask this question is "Are they informationally the same?" This is usually far more difficult to ascertain.

The data structures used in a program are a part of the program's ontology, as are the procedures it uses. When we set out to integrate two existing programs, we want to be able to measure their ontologies because it is necessary that their data structures correspond in some way. That understanding allows us to couple them directly or through an interface. Two ways to improve software engineering are (1) to develop methods for creating and publishing these ontologies, and, (2) create processes for measuring informational objects and determining their role in the software programs.

One of the early issues in programming was modularity. As programs became more complex, and particularly as they began to be crafted by a team of people rather than a single individual, modularity became a design requirement. The possibility of reuse was another major impetus. Today, integration of software is probably more important than the creation of tailor-made programs. The challenge of integration is determining if the information-- data structures, knowledge bases and knowledge models, databases and their schemata, and the syntax and semantics -- are the same in each of the programs being integrated. Therefore, it is important to all systems that must share information – not merely the ones we might deem "intelligent" – that we have ways of measuring and comparing the information that each system uses.

VI. Conclusion

We do not know today how to measure equivalence of information nor its impact on a system. We need to be able to do so to evaluate existing systems, to engineer new systems, and, to integrate both. The benefits will be seen primarily in highly complex software systems that utilize a large amount of knowledge from either other programs with the system or devices in the real world. On the other hand, should the promise of "ubiquitous computing" come true, information will permeate physical systems as well. In this paper, we have argued that the measurement of the ontology of a system is a fundamental part of realizing these goals. We also indicated that some ideas are emerging in the areas of ontology standards and measurement.

References

Albus, J. S. [00] Features of Intelligence Required by Unmanned Ground Vehicles, in [Meystel and Messina, 00]

Arndt, C. [01] *Information Measures: Information and its Description in Science and Engineering*, Springer-Verlag, Berlin, Germany, 2001.

Birtwistle, G.M., Dahl, O-J., Myhrhaug, B., and Nygaard, K., *Simula BEGIN*,
Petrocelli/Charter, 1973.

Chaitin, G. J. [92] Information-Theoretic Incompleteness, World Scientific, 1992 (Reprinted 1998.

Chandrasekaran, B., J. R. Josephson, and V. R. Benjamins [98] Ontology of Tasks and Methods, 1998 Banff Knowledge Acquisition Workshop. [Revised Version is cited above. It appears as two papers "What are ontologies and why do we need them?," IEEE Intelligent Systems, Jan/Feb 1999, 14(1); pp. 20-26; "Ontology of Task and Methods," IEEE Intelligent Systems, May/June, 1999.]

Jaynes, E. 88, `The Relation of Bayesian and Maximum Entropy Methods (510Kb),' in *Maximum-Entropy and Bayesian Methods in Science and Engineering,* 1, G. J. Erickson and C. R. Smith (eds.), Kluwer, Dordrecht, 1988

Maedche, A. and S. Staab, Comparing Ontologies – Similarity Measures and a Comparison Study, Internal report 408, Institute AIFB, University of Karlsruhe, Germany, 2001. http://ontobroker.semanticweb.org/ontos/report-aifb-408.pdf

McGuiness, D., R. Fikes, J. Rice, and S. Wilder, an environment for merging and testing large ontologies, *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning*, Brekenridge, CO, 2000.

Meystel, A. and E. Messina, Eds., *Measuring Performance and Intelligence of Systems: Proceedings of the 2000 PerMIS Workshop*, National Institute of Standards and Technology, 2000. http://www.isd.mel.nist.gov/research_areas/research_engineering/PerMIS_Workshop

Noy, N. F., and C. Hafner, 1997, The State of The Art in Ontology Design: A Survey and Comparative Review, *AI Magazine*, 18(3), 53-74 (1997)

Noy, N. F., and M. Musen, Prompt: Algorithm and tool for automating ontology merging and alignment, in *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, Austin, TX, 2000

Osgood, C. E., G. Suci and P. H. Tannenbaum [57*], The Measurement of*
Meaning. Urbana: The University of Illinois Press.

Reeker, L. [00] Theoretical Constructs and Measurement of Performance and Intelligence in Intelligent Systems, in [Meystel and Messina, 00]

Simon, H. A. [69] *The Sciences of the Artificial.* Third edition, Cambridge, MA, MIT Press, 1996. [First edition published 1969].

Shannon, C. and Weaver W. [71], *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, IL, 1971.

Stonier, T. [91], *Information and the Internal Structure of the Universe*, Springer-Verlag, Berlin, Germany, 1991.

Wirth, N. [76], *Algorithms + Data Structures = Programs,* Englewood Cliffs, N.J : Prentice-Hall, 1976. Reeker, L., Theoretical constructs and measurement of performance and intelligence in intelligent systems, in [Meystel and Messina, 2000].

Walther, E., H. Eriksson, & M. A. Musen. [92] Plug-and-Play: Construction of Task-Specific Expert-System Shells Using Sharable Context Ontologies. AAAI Workshop on Knowledge Representation Aspects of Knowledge Acquisition, San Jose, CA, 191-198. AAAI, 1992.

Appendix: Technical and Scientific Progress Through Measurement

This appendix argues generally for the importance of measurement in technology and in science, of which the measurement discussed in the paper is an example. This importance is expressed succinctly in two statements of Lord Kelvin (William Thomson) in the 19th century. These statements are

"If you can not measure it, you can not improve it"
"To measure is to know."

The following sections describe in more detail what these statements have meant to engineering and science, thereby stressing the importance of paying attention of measurement considerations.

Let us assume that we are working on technology that is not underlain by an established scientific theory -- like robotics or AI. We should be aware of the relationship between technology and science, which is sometimes muddied when the public regards "information technology" and "computer science" as synonymous. Of course technology and science are linked, but they are separable, both logically and historically. As a rule, some technology in any given area has developed before the corresponding science. In a symbiotic relationship, technology has been stimulated by scientific interests and aided by scientific knowledge, and much scientific discovery has occurred in or been motivated by technology.

Science begins with *curiosity*, but technology starts with more mundane *needs*. A need is perceived and made more precise in what systems engineers call a set of *requirements.* Once that happens, any techniques available may be used to fill the prescription. For complex requirements a good deal of ingenuity is required, so we have come to call the people who transform the prescriptions into technology engineers. The ingenuity and experience needed for engineering has not required a developed science, but engineering has always been improved by the ability to measure. Comparison, matching, and duplication are engineering uses of measurement, and important for meeting requirements. Their usefulness was known by the time the Great Pyramids of Egypt were constructed (probably long before).

Today, we tend to link science and engineering because engineering is frequently able to call upon science to *predict* the outcomes of engineering processes that may be breaking new ground (not just ones that require matching parts or duplicating previous artifacts). The ability to predict is a key aspect of the understanding that scientific theories provide, and is a clear transfer from the ability to measure. But the use of substantial amounts of scientific understanding to improve engineering is relatively new because the development of scientific understanding has been slower than necessity-driven technology, requiring a similar but different kind of creativity.

From the standpoint of computational systems or physical systems, or their combination in robotics, we use all sorts of measurements in the construction process, but also in evaluating. We measure the performance of artifacts for engineering purposes, either to test the performance limits of a single artifact, to test its conformance to requirements, or to compare multiple artifacts. If it is for conformance, it may be done by matching quantitative behavior to requirements. If the requirements are qualitative, the number of requirements met and/or the degree to which they are met is interesting. Measurement can determine the success or failure of a portion of a technology project or of the entire project (or device, if that is the outcome of the project). But success is rarely absolute, and requirements met lead to ideas for better or stricter requirements. As Lord Kelvin pointed out, measurements provide a way of meeting these new requirements and thus of improving the product.

If project requirements are not easily translated into a behavioral outcome, then models of various suggested approaches can be developed and their behaviors may stimulate the development process by people who tacitly know the needs but may not have been able to articulate a satisfactory set of requirements. The point is, however, that thinking about tests and measurements that might indicate the success or provide data for comparison can both prove and improve the outcome. This may be seen as the beginning of science, since scientific theories are models that meet certain requirements beyond those of a particular project.

Engineering is the process of creating artifacts, and "engineering sciences" developed by studying the process, are themselves sciences of the artificial. But for most engineering sciences, there are also underlying physical sciences. Because of that and because physical science provides the leading paradigm of science as it has developed over the ages, it is useful to consider examples of physical theoretical constructs. (It is well to keep in mind that informational theoretical constructs are going to be primary in computer science and artificial intelligence and a major consideration in robotics.)

Consider the construct *gravity*, which has a theoretical basis traceable to Galileo and Newton, refined more recently by Einstein. The gravitational constant is a part of that theory that has major technological ramifications, such as great predictive value in ballistic calculations. If one is building a catapult to bring down the walls of a fortified city, it could greatly help in making the right design decisions before actually building one; though such catapults were engineered well before Newton, or even Galileo.

Similarly, Newton's laws of motion are very useful, and *mass* is a fundamental theoretical construct used in both gravity and motion. When we create theoretical constructs like gravity and mass and can measure them, they increase our understanding of the physical world and our ability to predict how artifacts will perform in all situations: "To measure is to know."

Measurable theoretical constructs from physical theory influence design decisions and increase the likelihood of meeting technology prescriptions, with efficiencies of time, resources, and effort. The same thing will be true for theoretical constructs in information sciences and their related engineering branches as they develop.

In summary, science and technology both require measurements, and each has its separate needs. For technology, measurement is used to guide the engineering process and to check both the process and its products against requirements (the term often used is "validation and verification"). Often, however, intermediate measures can be found during the engineering process that turn out to have predictive value as to the final performance of an artifact. These measures may be indicative of important theoretical constructs that can enrich understanding within an underlying science. Science can exist by itself, and it originates in a basic human need to understand the world. Technology has existed for longer, as long as people have had a need for artifacts. Science and technology enrich each other, and measurement enriches them both.

Evaluation of Intelligent Systems: The High Performance Knowledge Bases and IEEE Standard Upper Ontology Projects

Adam Pease
Teknowledge
apease@ks.teknowledge.com

We can consider two ways in which intelligent systems can be analyzed; with respect to a particular task and *a priori*. In this paper we discuss a particular knowledge based system and its performance on a task, as well as the *a priori* metrics which may be applied to ontologies.

The DARPA HPKB (Cohen et al, 1998) project was a large (>$30M) effort to develop large knowledge based systems that would be significantly more competent on a wider range of tasks than the expert systems of the past. In order to motivate rapid development, the program was arranged as a competition between sets of developers. Three *challenge problems* were developed as tests of performance for the systems that were created. These were *battlefield engineering* which produced reasoners which constructed plans for repairing infrastructure such as roads and bridges, *course of action analysis* which produced reasoners which critiques Army plans, and *crisis management* which produced reasoners that gave advice on aspects of international crisis situations.

In each challenge problem, the participants were presented with a set of background knowledge, expressed in English, and a set of test questions that were either expressed in English in the case of battlefield engineering and course of action analysis or in a structured language in the case of crisis management. The participants were provided the opportunity to translate background knowledge by hand or semi-automatically over several months prior to the tests as well as taking sample tests that were "graded" by human experts. The actual tests were conducted in several phases over 2-4 weeks with the results again graded by humans.

While performance was a primary metric that was assessed by number of questions answered correctly, there were additional measures that included the amount of effort expended both before and during the test by project personnel (person-hours). In (Cohen et al, 1999) an analysis was conducted after the fact to see how much knowledge based content was reused from one test to the next. This was a critical measure since one of the purported advantages of knowledge-based systems is reuse of knowledge across tasks. We believe that there is modest support for this assertion. It was found that broadly 1/3 of the most general-purpose upper level content was reused. One-third of the reuse was of "middle-level" content. This is content that addresses a particular area of knowledge such as human social interaction or common-sense knowledge about vehicles, but can be applicable across many domains. One-third of the knowledge needed to answer any particular test question was created at the time of the test.

While one might have expected to have greater reuse, these measures are somewhat conservative since they consider only the appearance of terms or axioms in the trace of the solution to a particular test question. They do not consider the considerable benefit to the knowledge engineer from having a large ontology present that aids in placing, organizing and defining brand new concepts. The authors of (Cohen, et al, 1999) discussed possible metrics for knowledge support but did not reach a set of metrics suitable for publication. More research is needed.

One key aspect of knowledge base performance is speed. The TPTP (Sutcliffe & Suttner) suite is a set of general-purpose theorem prover tests that assess both speed and expressiveness of inference systems. A compromise must often be made on creating expressive knowledge representations in order to reach acceptable speed of inference. Description logic is one class of logics that have good theoretical performance aspects that are traded off for a language that is more limited in expressiveness than full first order logic.

We will now consider *a priori* metrics and guidelines. Some guidelines were introduced in (Pease et al, 2000). A balance must be achieved as to the *fan-out* of concepts. Either the extreme of a deep and narrow or shallow and broad ontology should be avoided. A deep and narrow ontology is likely to have many unnecessary distinctions that could be better represented as properties. A shallow ontology is likely to miss important intermediate concepts that enhance the reusability of an ontology.

Another key attribute of a good ontology is the *compositionality* of concepts. The more that complex notions can be expressed as combinations of functional application and properties, instead of being *compiled* into a single concepts which lacks explicit logical definition, the more reusable the knowledge base is likely to be.

A good ontology for practical computation should also take advantage of the lessons learned from analytical philosophy. Some of those lessons are addressed in

(Guarino & Welty, 2001) and include that: all instances of any sub-class are necessarily instances of the super-class, some properties (rigid properties) are ascribed to objects throughout their lifetimes, and some properties (non-rigid properties) are not permanently ascribed to objects, and that the conditions for membership in a class of the ontology should be specified as fully as possible. The IEEE Standard Upper Ontology effort (IEEE, 2001) is attempting to include these lessons in the construction of a general-purpose upper ontology. The challenge of this project is that direct measures of usefulness are not possible since no one particular application is the focus of the effort. The determination of *a priori* metrics is all the more critical. The IEEE SUO currently has two “starter documents” which are described in (Niles & Pease, 2001:1, 2001:2) and (Kent, 2001)

Cohen, P, V. Chaudhri, A. Pease, and R. Schrag (1999). "Does Prior Knowledge Facilitate the Development of Knowledge Based Systems", Proceedings of AAAI-99.

Cohen, P., R. Schrag, Jones, A. Pease, Lin, Starr, Gunning, and Burke (1998). "The DARPA High Performance Knowledge Bases Project", AI Magazine, Vol. 19 No.4, Winter.

IEEE, (2001). IEEE Standard Upper Ontology web site: http://suo.ieee.org.

Kent, R., (2001). A KIF Formalization for the IFF Category Theory Ontology, *in Proceedings of the IJCAI-2001 Workshop on the IEEE Standard Upper Ontology.*

Niles, I. and Pease, A., (2001). Origins of The IEEE Standard Upper Ontology, *in Proceedings of the IJCAI-2001 Workshop on the IEEE Standard Upper Ontology.*

Niles, I. and Pease, A., (2001). Towards A Standard Upper Ontology, *in Proceedings of the second conference on Formal Ontology in Information Systems*, to appear.

Sutcliffe, G., and Suttner, C., (2001). The TPTP Problem Library for Automated Theorem Proving web site: http://www.cs.miami.edu/~tptp/

Welty, C., and Guarino, N., (2001). Evaluating Ontological Decisions with OntoClean. Unpublished.

Meta Models to Aid Planning of Intelligent Machines

Paul K. Davis, senior scientist, RAND and professor, RAND Graduate School
and James H. Bigelow, senior mathematician, RAND
1700 Main St., Santa Monica, CA 90407-2138
pdavis@rand.org; James_Bigelow@rand.org

Abstract—Relatively simple low-resolution models are needed by human planners and probably by intelligent machines. Ideally, these should be high-level models developed in a multiresolution, multiperspective modeling (MRMPM) framework. That, however, is often difficult. We ask whether statistical meta modeling (i.e., development of response surfaces) can provide good low-resolution models if one already has a credible higher-resolution base model. We ask how meta models compare if they are derived from pure statistical methods, from a phenomenology-rich theoretical approach, or from some synthesis. To sharpen issues and generate insights, we have worked through a particular problem in detail. Our conclusions are generally negative about "purist" statistical meta models, which have serious shortcomings in explanatory power, in variance, and in ability to predict and explain the relative importance of contributing variables. Purely theoretical approaches, however, are often very difficult and not transparent. Fortunately, a synthesis of methods is feasible and likely to be fruitful. Some tentative principles are that: (1) a thoughtful "first-order" theoretical analysis conducted with MRMPM principles in mind can identify "aggregation fragments" to be used as variables in generalized regression and (2) this can also suggest structures to impose on the meta model that will assure dependences known to be important. Imposing such a structure can, e.g., assure that a meta model will predict failure of a system if any of its critical components fail. The theory-enhanced statistical meta model may also be much better than a naive statistical meta model in representing a system's performance when a competitor is systematically looking for a circumstances that will defeat the system. In that case, variables that are mathematically independent may be said to be strategically correlated. Although tentative, the suggested principles appear consistent with experience in theoretical and experimental physical science.

Index Terms— Multiresolution modeling, variable resolution modeling, response surfaces, meta models, model abstraction, planning models.

I. Introduction

This paper addresses the problem of how to develop low-resolution, meta models as part of a multiresolution family. In particular, it compares approaches based on phenomenological modeling with methods based on statistical methods. It then suggests some steps toward synthesis.

The paper begins with some background on multiresoluiton modeling and the reasons meta models are needed. It then discusses the ideal for phenomenological multiresolution modeling, which involves pure hierarchies. Although that ideal can sometimes be realized with considerable payoff, reality is often much more complex. As a result, developing phenomenology-driven multiresolution families proves quite difficult. This causes us to be interested in shortcuts, such as using statistical methods to develop meta models. The remainder of the paper is about our efforts to think about how statistical methods and more phenomenology-rich methods relate to each other and whether there is the possibility of combining features of both. We describe our initial hypotheses on the matter, the research approach we have taken so far, and observations to date.

II. Background

A. Planner Needs for Low Resolution Models

It is well recognized by now that intelligent systems need planning modes in which they are able to recognize and compare alternative courses of action.[1-4] This planning requires a *broad* form of testing—i.e., the courses of action need to be evaluated for a wide range of circumstances. This is the domain of *exploratory analysis*, rather than the domain of refinement. The objective is often the classic goal of satisficing—finding a course of action that will "do the job," not necessarily optimally, but well enough.

It follows that humans, at least, typically need low-resolution models for planning. This is not simply a matter of saving time or money, but rather due to the human need to *understand* the basis for choosing one course of action over another, and to communicate that rationale to others—perhaps to persuade, or perhaps to convey a clear sense of mission intent. This need might not exist if a perfect model existed with perfect data, and if everyone accepted whatever the model said. That situation, however, rarely arises in higher level planning.

A corollary is that the need for simple, low-resolution models will continue to exist regardless of increasing computer speed. The need is fundamental. It is tied to the limits of cognition and curse of dimensionality.

It might be speculated that intelligent machines can be different on such matters. They have no emotional need for explanation and they may not need to explain their reasoning in simple terms—at least when communicating with other intelligent machines. Nonetheless, it seems likely that when the intelligent machines have imperfect models, limited data, and uncertainty about prospective operating conditions, they will suffer the same problems of bounded rationality addressed famously by the late Herb Simon[5] a half century ago. If so, they will also need simple, low-resolution models.

This said, even those who gravitate toward simple, low-resolution models will agree that to be useful, such models need to be grounded in reality. It is frequently easy to concoct plausible and attractive simple models, but such models are often flawed—so much so as to be counter productive. Sound "simple" models should be rooted in higher-resolution work. Thus, to conclude that the planning function requires simple models leads in due course to the requirement for multiresolution modeling (MRM). Indeed, it is not just a matter of resolution. Substantially different representations of reality (different "perspectives") may be essential in order to understand different facets of the underlying phenomenon or to make effective use of diverse forms or empirical data. Thus, what is needed is actually multiresolution, multiperspective modeling (MRMPM).

For the remainder of this paper we shall focus on MRM, but the more encompassing concept of MRMPM is important to keep in mind.

Having established motivation, let us now discuss what is involved in MRM.

B. Idealized Multiresolution Modeling: the Role of Hierarchies

For a phenomenologist, at least, the natural way to proceed in developing an MRM family is to design hierarchically.[6,7] Figure 1 illustrates schematically an idealized construct. One has only a few top-level variables (those in the low-resolution model), but each of these is determined by higher-resolution phenomena. The next level of detail will be a model with more variables and it, in turn, will depend on events at still higher detail. In Figure 1, the resulting hierarchical trees are pristine.

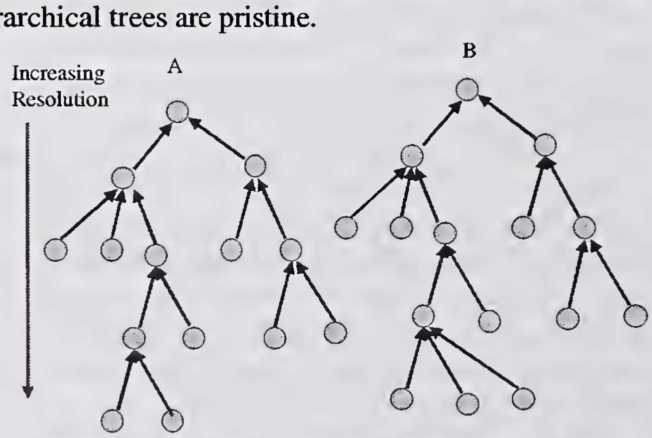


Figure 1—Idealized Multiresolution Modeling

Why is this "ideal?," or at least very desirable? For one thing, given such a multiresolution family, one can start at the top and then—as necessary—zoom to a higher level of detail, perhaps on only one part of the problem. For example, one might thoroughly understand variable A, but variable B might be uncomfortably abstract. If so, one could go down one or more levels of detail until the variables used are comfortable and sufficient—perhaps because they are explicitly tied to familiar empirical information. This zooming, however, would be on an as-necessary basis. Reasoning could be accomplished at as high a level, and with as few variables, as needed for comfort.

Such a multiresolution family would relate the microscopic and macroscopic worlds. It would provide a strong sense of "understanding" and the capacity to use diverse types of information. This relating of levels would not just be a matter of hand-waving. Instead, Figure 1 suggests that to establish good values for the higher-level variables when they are used as independent variables (inputs), one should conduct systematic experiments exercising the next higher-resolution model to generate appropriate "averages." Such experiments should be conducted over the entire n-dimensional space spanned by the independent variables of the higher resolution model. In some contexts, that is appropriately called a "scenario space."

Interestingly, the result of such calibration should generally be to produce *stochastic* variables. That is, if the higher-level (lower-resolution) model has two variables X and Y, and if we want to establish what reasonable values of X and Y might be, we should ordinarily expect that X and Y will need to be stochastic because of hidden variables.

Such idealized modeling is possible in many cases—if one thinks about doing it. Figure 2 shows an example drawn from recent defense work. [8]It shows the design of a module dealing with command and control issues in the evaluation of long-range precision fires. This model allows users to input directly the impact time of a weapon (measured relative to the ideal time of arrival at a target). This is often a useful quantity to parameterize and vary. However, the model also allows the user to work with more detailed variables as inputs. The second level of detail involves the descent time of the weapon (the time between when the weapon does its final target acquisition and tracking, when it is overhead, and when the weapon impacts) and the standard time-of-arrival error measuring the variation due to imperfect guidance system. At the most detailed level, the user must input the weapon's flight time, the delay between the receipt of sensor data on targets and the time that the data was valid, and so on.

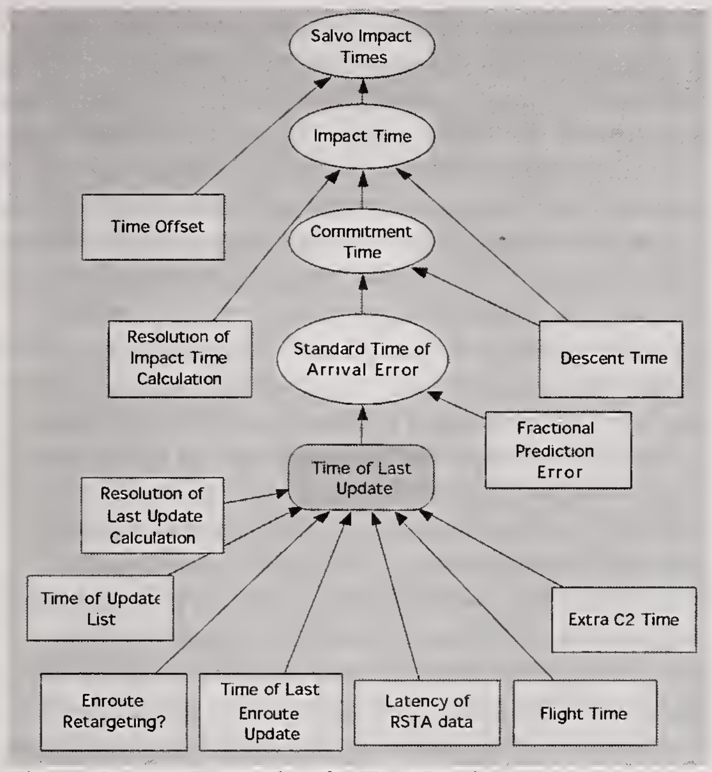


Figure 2—An Example of MRM Design

Idealized hierarchical design is unusual. If we look at an existing model and depict its relationships graphically, a more typical picture would be as in Figure 3. Here we see a good deal of cross talk and breakdown of the hierarchies. A common observation here is "Everything is connected to everything." Often, it is not evident how to simplify to something more like Figure 2.

This may be puzzling to those who know about and accept the principle that natural complex adaptive systems typically manifest the principle of nearly decomposable hierarchy:[5] that is, when viewed in the right way, the system can be decomposed into modules that interact only weakly. Such a decomposition is typically not evident when viewing the structure of existing complex models. Nor is it evident in freshly built models designed bottom-up with the common ethic of achieving verisimilitude. Indeed, it is not evident even in models built top-down if the designer is taking pains to include interactions that appear important. There are at least two points here. First, people only seldom design models with an image such as Figure 1 as a goal. Second, even if they try, they will find that their diagrams become muddled, as in Figure 3.

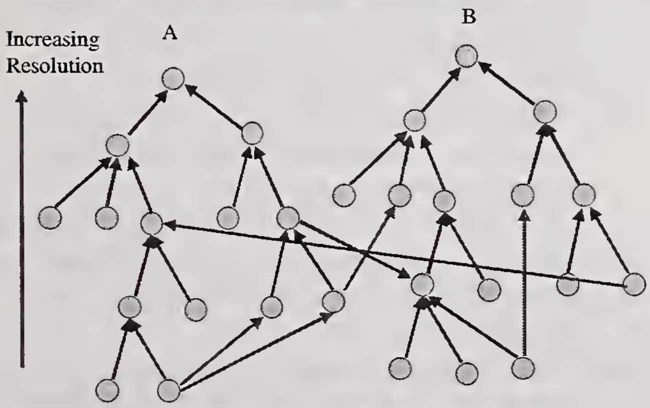


Figure 3—A More Typical Model Schematic

The solution, it might seem, is to recognize that approximations can eliminate the ugly interactions. Indeed, if one is willing to introduce approximations, then it *is* often possible to move much closer to the MRM ideal. And, if one does this right, one will rediscover the principle of nearly decomposable hierarchy.

C. Intrusion of Reality

Unfortunately, another fundamental reality intrudes here. The critical approximations are often valid only in limited domains. As one moves from one domain to another, the appropriate approximation may change drastically—not just through a change in some constant, but in the analytical structure. For example, aerodynamic drag may vary in one regime in proportion to an object's speed, whereas in another regime it may vary inversely with that speed. Yes, approximations are essential, but we should not expect to find simple, stable, universal approximations.[7]

The significance of this is that—once again—anyone attempting to develop a phenomenology-based MRM design in a given problem should not be surprised to find difficulties—difficulties great enough to comprise a PhD dissertation.

How, then, do we humans "get along" in this complex world? In fact, we do reasonably well. However, we are constantly changing the frames in which we operate (the approximate depictions of the world that allow us to reason and act). We do this so seamlessly that we often are not even aware that we have changed frames. The attribute of being able to carry along contradictory ideas at the same time—most celebrated in discussion of eastern philosophy, but actually a universal attribute—is arguably a manifestation of this.

What about machines? How will intelligent machines develop the diverse frames and skills to adopt the right frame at the right time? This remains very much a research question.

To complete our background discussion, let us summarize by observing that while simple, low-resolution models are needed, and while they need to be rooted in a multiresolution framework, achieving one is often difficult. Learning how to achieve MRM structures efficiently would be very desirable.

III. Can Statistical Meta Modeling Provide a Shortcut?

A. General Issues

The difficulties to which we have alluded so far are all tied to attempts to build phenomenological models—i.e., models rooted in theory and attempting to describe causes, effects, and other relationships. Suppose, however, we back away from this and ask whether an alternative approach is possible. The most obvious is statistical meta modeling, the very purpose of which is to develop simple "models" that represent well the behavior of systems on which some kind of data exists. The system in question may be a physical system and the data may be empirical. Alternatively, the system may be a detailed model (e.g., a simulation of a system) and the "data" may be outcomes of simulation runs. In some instances, the detailed models are large, complex, impenetrable, fragile, and slow. In other cases, they may be virtuous in all respects other than requiring expensive care and feeding. Typically, the base models are imperfect, with both known limits of applicability and errors.

In all of these cases, one can apply well known statistical methods to generate meta models. If a reasonably well accepted detailed model exists, why should we not adopt these methods to generate the simple, low-resolution models needed for planning?

This is the question we have been studying. We have sought to understand better the strengths and weaknesses of the phenomenological approach and the approach of statistical meta modeling. And we have sought opportunities for synthesis.

B. An Aside

One reason that pursuing this matter was of interest is that it highlights a substantial cultural divide, which can be characterized—with literary license—as follows. Suppose we ask whether using statistical methods to generate simple low-resolution models for planning is sensible. The responses from Cultures A and B might be::

> *Culture A*: "Of course they make sense; all that matters is representing behavior of the base model. I don't even *want* to understand the black box." (statisticians, some operations researchers, many social scientists,...?)
>
> *Culture B*: "No no no; the simple model should be a model, not some lousy regression. I'd rather calibrate a model that makes sense than work with a mysterious black box." (physical scientists, engineers,...?)

Culture A and Culture B even mean quite different things by the word "model." Fortunately, translations are possible.

IV. Approach

In our first assault on the issue, we proceeded on two tracks. On the first track, we theorized in the abstract, using simple examples to help, but without attempting anything rigorous. The purpose was to generate hypotheses for experiments. For our second, experimental, track, we decided to work though a particular nontrivial example drawing on a currently interesting military problem with which we were familiar. For that second track, we decided to

1. Construct (by embellishing an existing model) a complex, nonlinear model that we would treat as correct
2. Use standard methods to develop statistical meta models
3. Throw different degrees and types of theory at the problem—providing "hints" before applying the statistical apparatus.
4. Observe, compare results with differing levels of theory, compare results with expectations from initial notions, and learn.

More ambitious theoretical work would certainly be possible, but this hands-on experimentation was suitable to our state of knowledge and the limited time available for the research (in between our principal research efforts). Although our example involved a specific military problem (assessing military capability of alternative military forces to halt an invading army by using long-range fires in the form of aircraft and missiles), we convinced ourselves that the example would illustrate many generic issues.

The base model (called EXHALT-CF)[9] has input variables such as the number of resources always available (forward-deployed shooters, such as fighter aircraft), the rate at which those can be increased (deployment rates), the times at which partial and full rates of increase would be initiated (related to strategic warning, time of decision, time at which access to bases is granted, etc.), and so on—to include the effectiveness of the resources (kills per shooter-day) and the size of the task to be accomplished (the number of threat divisions, etc.). An important output is the distance that would be moved by the attacking army before it is halted. The meta model, we would hope, would be able to predict this distance from a much smaller set of inputs. The inputs could be a subset of the original model's inputs or a set of composite variables such as the sum of two high-resolution inputs (or, realistically, something much more complex).

V. Issues and Hypotheses

Before beginning the experimental phase of our study, we developed a set of issues and hypotheses to guide our exploration. These included the following:

- Black-box models (such as statistical meta models) are less useful to decision makers than phenomenologically motivated models with clear physical interpretations. Thus, if they are to compete effectively, they must be accurate and reliable.
- Statistical meta models may be relatively accurate "on the average," but may be seriously misleading for predicting sensitivities and variation.
- Statistical meta models may be seriously misleading on crucial "system issues" (to be discussed below).
- Some statistical methods may yield expressions with meaningful physical interpretations by "discovering" composite variables.
- The potential advantages of models based in theory (i.e., phenomenological models) may not be realized in practice because the resulting analytical forms turn out to be ugly, complex, and opaque.
- A synthesis of approaches may be desirable: one in which theory is used to guide application of statistical tools.

The first of these reflects our ingoing attitude (statisticians might say bias). In candor, our effort has not really been devoted to finding new statistical methods to improve accuracy. Many first-rate researchers work on such matters and a considerable literature already exists. Instead, our real objective is suggested by the last item in the list: the belief that a synthesis of theory-based and statistical methods might prove practical and attractive. As indicated by the middle items, we also were suspicious about how meta models developed with relatively standard methods—could be on issues of interest to us. Particularly interesting to us here was the "system issue." By this we mean that many important problems are about assessing the capabilities of systems with multiple individually critical components. Such systems depend for their success on all of these critical components separately proving successful. Not all systems are of this type, but many of interest are. Analytically, to say that a system depends on each of subsystems A, B, and C being successful suggests that overall capability depends on something more like a product of capabilities, $C_A C_B C_C$, than a sum. Figure 4 shows in the representation of a fault tree the structure of the halt problem on which we focused for our example. This fault-tree representation highlights the system character we have in mind: success in achieving an *early* halt of an invasion requires success in each of the four components indicated by branches.

We would not expect normal linear regression to generate good meta models when such system effects are present. Even generalized regression methods, which consider various nonlinear composite variables, typically do not include triplet products. This justified our suspicion, but proved nothing because in practice statistical models often do much better than one would expect a priori. Further, dependences among variables, such as represented by product terms $C_A C_B C_C$ can sometimes be reasonably approximated by a sum of terms such as $C_A C_B$, $C_A C_C$, and $C_B C_D$. We were also impressed by the common lore among statisticians that pair wise interactions among variables are typically sufficient for meta modeling—that diminishing returns sets in quickly in considering interactions. This lore was in conflict with our theory-based reasoning, but merited respect as we constructed hypotheses to explore. Finally, several advanced statistical methods (e.g., cluster methods) appeared to merit investigation if time permitted.

VI. Selected Observations

With this background of motivation and approach, let us now describe briefly some of the observations we have made to date, based on our experiments—which should be viewed more as developing a case history and making observations about it, than as something rigorously systematic.

A. Success of the Statistical Meta Models

We ran 1000 cases of our base model, generating them randomly from the input space of the model by representing the input variables with random distributions. We then developed a series of increasingly sophisticated statistical models while avoiding insertion of phenomenology. The meta models were based, in increasing order of sophistication, on:

- Conventional linear regression of all the input variables
- Modestly extended linear regression in which the variables used as the basis for linear regression were composites of the original input variables—composites motivated by looking for consistency of dimensionality in many of the variables regressed. In particular, we constructed a number of composite variables with the dimensions of distance.
- More generalized regression using as the basis not just the original input variables $\{X_i\}$, but also the various product terms $\{X_i X_j\}$.

As expected, the linear regression did not do particularly well (although better than one might expect), but with the embellishments, we obtained fair agreement with the predictions of the actual base model. This conclusion, however, applied only so long as we focused on "standard" measures, such as R^2 or, better, root mean square error. Root mean square error varied from about 60-100 km, depending on which statistical model we attempted. Since the goal was to achieve a halt distance less than 100 km, this degree of variation was not really satisfactory—although, again, it was better than one might expect given the complexity we believed existed in the original model.

When viewed in a more fine-grained way, results were worse. For example, some of the coefficients had nonsensical signs and the errors of individual cases made no sense. But why should they have made sense when the "models" used had little physical content?

Most important, the statistical meta models did not do well when used to compare the relative importance of variables. A basic reason for this is that the statistical meta model is created by reducing average error over the entire input domain. However, in many problem areas—such as military problems where one has a thinking adversary, or an economic domain in which choices are not made randomly but to maximize profit—small "corners" of the input space can be sought out. For example, an adversary may minimize warning time *and* invade rapidly *and* use various tactics to degrade the defense's capabilities—even if temporarily. Predicting outcomes for a corresponding war might mean running the model for a set of inputs that would be regarded as extremely improbable if they were independent and random. One way to think about this is to refer to the inputs as mathematically independent, but strategically correlated.

It is easy to understand how a purely mathematical effort to assess the relative importance of variables can run into trouble. Such an effort might, for example, measure the average effect of a 1% change in a given variable when averaged over all of the rest of the input space. If that variable was extremely important only in one "corner" of the space, that fact would be lost as the result of the broader averaging.

Another way to think about the problem is to look at graphs comparing predictions of the meta model with the base model. Not uncommonly, the meta model will do poorly in one domain and poorly (but with opposite sign in the error) in another domain. It will also do extremely well in some domains and quite poorly in others, even though, on average, it will do fairly well. When one asks about the validity of an approximation or the relative importance of a variable in such a case, the result will be correct on average but potentially quite misleading.

The problem, some might respond, was in considering too large an input space. In a sense, that is true. However, which "corner" of the space is of interest depends on details of context that are difficult to predict in advance. Nonetheless, this is the essence of the problem.

B. An Infusion of Theory

What happens, then, when we add bits of theory before generating the statistical meta models? Suppose, for example, that a problem has three inputs X,Y, and Z. Adding theory might be to assert that that meta model should have the form $C_1XY/Z + C_2X$. The composite variables forming the dimensions for regression, then, would be Q_1 and Q_2, where $Q_1=XY/Z$ and $Q_2=X$. We have elsewhere called these "aggregation fragments." suggested by theory. Linear regression could then be used to determine the coefficients C_1 and C_2. And, if one were lucky, perhaps C_2 would be small and the meta model could be simply C_1XY/Z.

In more realistic cases, of course, the base model might have dozens of inputs and the composite variables might be complex as well. Further, it might or might not be possible to use linear regression straightforwardly. In the case we worked in detail, for example, the form suggested by theory involved Max and Min operators, which can cause trouble. Tricks can often be applied, however, such as breaking the data into groups and applying the methods of linear regression on the groups separately, or ignoring the Max and Min operators until after finding a regression model and then applying the operators. What is valid depends on details of the problem.

What we learned from our experimental application of our ideas was the following:

- Infusing the approach with theory-motivated aggregation fragments may or may not improve the meta model significantly if the only measure of goodness is something like R^2 or root mean square error.
- However, the resulting meta model will at least have pieces with understandable significance. That is, its descriptive value will be higher.
- Further, the enhanced meta model may be more accurate in predicting relative importances and may help users avoid serious pitfalls. If, for example, one knows that it is the product XY/Z that matters most (although X,Y, and Z may also appear in the definition of some of the less important composite variables), then that could be quite useful in drawing valid conclusions–and ignoring artifactual conclusions—about relative sensitivities. Also, if theory were to tell us that an aggregation fragment $Q_1 = \sum_i^n X_i$ should be important, then one could avoid the error of concluding from a more naive meta model that the individual variables $\{X_i\}$ are unimportant. That is, the coefficients of a naive regression might be only a third as large for each of the X_i, as that for, say, X_{n+1}, but if n were 10, then Q_1 would be more important than X_{n+1}—if only one knew to look for Q_1.
- Most important, perhaps, our experiments confirmed the potential value of imposing a theory-motivated "system structure" on the meta model.

To illustrate this trivially, suppose that we were interested in the rate at which something could be detected from searching an area. Elementary theory would tell us that the rate would depend on the product of search rate R and the probability of detection when viewing an area that in fact contains the item of interest. At a more microscopic level, there might be a great many variables such as the search vehicle's speed, time on station, turnaround time for refueling and repair, search pattern, and so on. Also, the detection probability in the sense that we mean it might not appear. Instead, one might have inputs for the power and aperture of a radar, its scan rate, the radar cross section of interesting objects, the probability of recognizing that a particular moving object was an example of the item in question, and so on. A linear regression of these variables might produce something useful, but would not pick up the right form. If instead the meta model were assumed to have the form RP_d, where R was constructed from the search vehicle's attributes using even something as simple as dimensional analysis, and where P_d was assumed to be a product of the sensor attributes and target cross section (but limited to 1), then the resulting meta model would be guaranteed to have the characteristic that the search would be predicted to be a failure if either R or P_d were too small. That is, one would not make the mistake of predicting that one could compensate for a very poor search platform by upping the performance of the power and aperture of its radar.

In the actual problem that we worked through experimentally, the meta model that we concluded should be tried based on theory had the form shown in the equations below, where the independent variables were Obj (the objective sought by the attacker, corresponding to the distance from his border to a strategically important destination), V (the initial movement rate of the attacker), ξ (the number of attacker armored vehicles that the defender must kill to halt the invasion), δ_{max} (the number of kills each defender shooter can kill each day using the best weapons available), δ_B (the same quantity, but for a poorer weapon available in large numbers), T_{SEAD} (the time required to suppress the attacker's air defenses so that shooters can operate effectively), T_x (the time at which shooters begin their attack on the armored column), R (the rate at which shooters deploy to the region), A_0 (the number of shooters present when the war starts), A_{max} (the maximum number of shooters that can be in the theater), N_{awpn} (the number of top-quality weapons), and Ω (the slowing of the invader's movement for each vehicle killed per day).

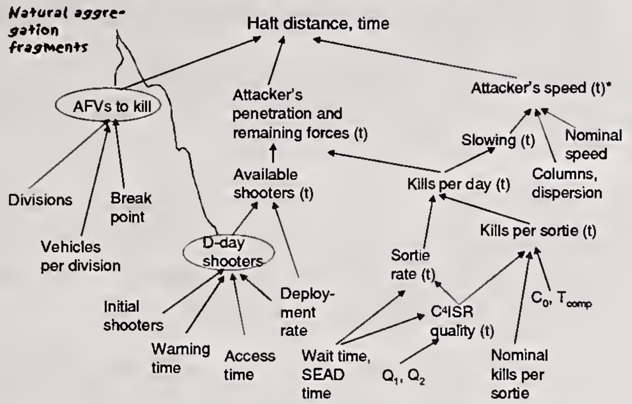


Figure 4—Finding "Aggregation Fragments"

Details are not of interest here, but note that the theory-motivated meta model is quite nonlinear and that it has recognizable "system features" in that, for example, the distance gained by the attacker can be large if it the attacker's size ξ is large or if the defender's per-shooter-day effectiveness δ_{max} or δ_B is low or if the defender has too few shooters on average. The form is not that of a simple product because there are other complications, but that "product" feature is prominent in the expression for the composite variable D_2.

$$D = Max[Min[D_2 - C_1 T_{delay} Obj],0]$$

$$D_2 = C_2 \frac{\xi_A}{\bar{A}\delta_A} + C_3 \frac{\xi_B}{\bar{A}\delta_B} - C_4 \Omega \xi$$

$$\xi_A = Min[N_{awpns} \frac{\delta}{SN_a}, \xi] \quad \xi_B = \xi - \xi_A$$

$$\bar{A} \approx Min[A_0 + RT_x + \frac{1}{2} R(T_{SEAD} - T_x), A_{\max}]$$

Without elaborating, let it suffice to say that this theory-motivated meta model did spectacularly well—even embarassingly so. We say "embarassing" because the base model took months of work to develop, code, and debug, and is in no way simple and transparent. Nonetheless, the underlying factors driving its results are largely those summarized in the compact expressions above. To someone interested in this particular problem, the structure of this expression and the various terms can be explained clearly in a matter of minutes.

As one would expect, the theory-motivated meta model did well when asked to predict sensitivities and relative importances.

In our experience with this and vaguely similar problems, it has proven possible to develop "smart" suggested meta model forms with hours, days, or a few weeks of work, rather than months. To be sure, this requires shifting mindsets from that often associated with procedural programming to that like more traditional analytical modeling—even with use of paper, pencil, and a whiteboard.

In summary, our experiments tended to confirm the initial hypotheses and to give them sharper meaning. We can hardly draw universal conclusions from such experiments, but we are encouraged that the traditional methods of mathematical modeling and statistical meta modeling can be merged in developing useful low-resolution models that are reasonably suitable for the kind of high-level exploratory analysis needed for both policy planners and certain kinds of intelligent machines.

C. Other Observations

Finally, let us comment briefly on some issues that we had found puzzling at the outset. One of these was the common belief among statisticians who generate meta models using experimental designs to sample the results generated by a physical system or base model that interaction effects can typically be ignored beyond those of pairwise interactions. The reason for this is probably just that the applications are limited to problems in which a single nicely behaved "response surface" applies. If that is the case, then—by analogy with Taylor's theorem in calculus—one would expect the quadratic approximation would often be reasonably good. However, in policy problems—including the one that we used for our example—the non linearities caused by thresholds of various kinds result in a more complex and non monotonic structure. No single response surface suffices. Furthermore, in problems with which we are familiar the empirical data or realm of validity for the base model is often quite limited. It is important to be able to extrapolate the meta model's predictions well beyond the region for which it was calibrated. When this is so, it should hardly be surprising that a theory-motivated meta model (perhaps with various If-Then-Else constructions distinguishing broad regions) can be far better than a more naively generated statistical meta model.

VII. Conclusions

In summary, there is great potential in marrying the techniques of statistical meta modeling with the insights of theoretical, phenomenological, modeling. The benefits of such a synthesis are likely to be quite high when attempting to represent systems with individually critical components and complex systems with substantially different behaviors in different regimes of their input variables, and in predicting system behaviors for circumstances significantly different from those for which one has empirical data.

The synthesis we are suggesting rejects the "purist" approach of some statisticians, which is sometimes characterized as "Let the data speak," by which is meant that one should explicitly *avoid* postulating a theoretical structure to the model and instead see what the statistical analysis reveals. Such an approach has much to offer in many problems, but not the ones we are addressing. In our problems, it usually pays to have theory. The payoff is quite high in terms of its cognitive benefits (related to the model's expanatory power), which may be even more important than modest differences in the accuracy or precision of prediction. We believe that will continue to be the case for strategic planning. It may or may not be true in the long run for robots in cases where the data available for calibrating a meta model is massive and credible, but we suspect that paucity and unreliability of data will plague intelligent systems used in complex environments (e.g., planetary explorers rather than spot welders).

In attempting a synthesis of approaches, we suggest several principles:

- Attempt to characterize the problem using the methods of multiresolution, mutiperspective modeling (MRMPM)—especially the method of hierarchical or nearly hierarchical decomposition.
- Attempt to find meaningful simplified structures by sharpening the hierarchicies—i.e., by identifying approximations (perhaps case-dependent approximations) that create nearly decomposable hierarchies.
- In doing so, however, be guided less by the intuition or preferences of pure mathematics (e.g., independent events) than by the character of the actual problem. Worry about what we have called "strategic correlations."
- Attempt to characterize the problem "formally" even if one cannot as a practical matter accomplish the various computations implied. Attempt to structure the problem so as to "see" system features where one knows they should exist, but allow structurally for complications (e.g., even if unusual, it may be possible for one component—if present in quantity–to substitute for another thought to be individually critical).
- Abstract from this theoretical work both "aggregation fragments" and structure that can be used to inform statistical meta modeling.
- Try to identify variables that are being short-changed in the proposed structure and then avoid using the meta model for predicting the consequences of change in those variables, even though the meta model depends on them.

We are nowhere near providing firm principles or recipes for success, but we believe that the approach we suggest will prove quite useful. One reason for our belief here is that the suggestions appear to be in some respects a restatement—for a new context of inquiry—of methods that have long been applied by physical scientists and engineers.

ACKNOWLEDGMENT

This paper is based on research accomplished for the Air Force Research Laboratory. It also draws on work accomplished for the Office of the Secretary of Defense.

REFERENCES

[1] National Institute of Standards and Technology, Measuring Performance of Systems with Autonomy: a White Paper for Explaining Goals of the Workshop, final draft, Alex Mystel, July, 2000.

[2] Meystel, Alex, "Evolution of Intelligent Systems Architectures: What Should Be Measured," PERMIS 2000 proceedings.

[3] Meystel, Alex and Elena Messina (ed), *Measuring the Performance and Intelligence of Systems: Proceedings of PerMIS 2000*, National Institute of Standards and Technology, Gaithersburg, 2001.

[4] Messina, Elena, Alex Meystel, and Larry Reeker, "PERMIS' 2001 White Paper: Measuring Performance and Intelligence of Intelligent systems,"PERMIS' 2001, Mexico City, 2001.

[5] Simon, Herbert, *Sciences of the Artificial*, MIT Press, 3d. ed., Cambridge, MA, 1996.

[6] Davis, Paul K. and Richard Hillestad, "Families of Models That Cross Levels of Resolution: Issues for Design, Calibration and Management," *Proceedings of the Winter Simulation Conference*, 1992.

[7] Davis, Paul K. and James H. Bigelow, Experiments in Multiresolution Modeling, RAND MR-1004-DARPA, Santa Monica, CA, 1998.

[8] Davis, Paul K., James H. Bigelow, and Jimmie McEver, Effects of Terrain, Maneuver Tactics, and C4ISR on the Effectiveness of Long-Range Precision Fires: a Stochastic Multiresolution Model (PEM) Calibrated to High-Resolution Simulation MR-1138-OSD, RAND, Santa Monica, CA, 2000[2]Meystel, Alex, "Evolution of Intelligent Systems Architectures: What Should Be Measured," *PerMIS 2000 Proceedings*, National institute of Standards and Technology, August, 2000.

[9] Davis, Paul K., Jimmie McEver and Barry Wilson, *Measuring Interdiction Capabilities in the Presence of Anti-Access Strategies*, RAND MR-1471-AF, forthcoming.

Performance Evaluation in Computing With Words

A. Meystel
Professor of Electrical and Computer Engineering
Drexel University, Philadelphia, PA 19104

Abstract

We propose a theoretical background and a computational technique that evaluates the performance of systems of natural language processing. The system of our interest analyzes natural language texts (narratives of the questions presented by the analyst, and narratives of the sources, i.e. relevant documents), and generates new texts under various focus of interest (the *meta-intent*) and with various degree of compression. The narrative of the questions serves the purpose of determining the meta-intent and the required degree of compression. This is an equivalent of the set of goal, purpose and/or command that arrives from the upper level in any large complex system. The narratives of the sources can be considered the totality of the Elementary Loop of Functioning. It can have many levels of resolution, too. The engineering object of analysis is a software package whose inputs are a) the question, b) general information related to the analyst's foci of interest, and c) the set of sources (natural language texts). The products at the output are "action items": the list of recommendations. In many cases, it includes: a) the answer to the original question, and b) the knowledge structure that incorporates knowledge from the processed sources. Both parts of the output are natural language texts, too. The purpose of this analysis is evaluating the quality of the result.

Keywords: action, actor, behavior generation, compression, corpora, document, ELF, interpretation, knowledge representation, narrative, natural language, object of action, summary, summarization, text processing

1. Introduction

Unlike many existing devices for goal oriented text generation, the overall system of our interest employs mechanisms of a) constructing the architecture of knowledge contained in a particular text, and b) subsequent use of this architecture for constructing texts representing this knowledge with the desired degree of compression. The system learns from experience because its knowledge structure incorporates everything it learned from the sources and from the questions. The validity of knowledge can be judged by a human operator.

The processes of extracting relevant information from natural language documents require constructing an adequate knowledge organization based upon multiple sources. We believe that a meaningful interpretation of an analyst's question is possible, too, only within a framework of a particular knowledge structure (which might be different from the knowledge structure built upon the sources). Thus, the hub of our efforts is situated in construction of proper knowledge structures. We build them following the conceptual paradigm of the multiresolutional approach. Especially effective are our multiresolutional techniques of disambiguation, as related to the elements of natural language texts. The validity of disambiguation can be judged by the convergence of the processes of disambiguation.

The special advantage of our method and software package is that it builds up a knowledge representation and learns additional knowledge from each new text submitted. This new knowledge is used for the subsequent compression of texts even if it has not been directly represented in the expected sources: every new text of a particular domain is being compressed by a "more knowledgeable" package. These new texts generated by the system are the answers, and the "density" of the answer depends on the request of the asking analyst. Method allows for processing not only single texts, but also groups of texts. It can answer questions, groups of questions, refine questions, and disambiguate answers. It allows for preparing surveys, and maintains topic-oriented and context-oriented knowledge bases for a variety of decision support needs. The quality of decisions judged by the result of applying these decisions.

2. The Concept of the System for Processing the Questions and the Set of Sources

The Problem of Knowledge Extraction and the Problem of Text Compressing. Extraction of knowledge from texts seems to be of crucial importance for solving the problem of synthesizing the proper answer if the question is submitted and the sources of knowledge are available in the form of natural language texts. The skills of *abridging, summarizing, abstracting and surveying* are highly important for solving the problem of question answering. People are doing all these things intuitively and often fail. They tend to overemphasize trivial passages and overlook hard to infer connections hiding potential breakthroughs. They focus upon particularities and losing the larger picture. Obviously, the *text* of the document is not equivalent to the *knowledge* that is conveyed by this document, not to speak even about its *meaning* that for the same text can be different in the different *contexts*. There is no clear definition for "meaning" because there is no single way of conveying the content, thought, emotion or even a mood by using the arsenal of natural language. Currently, the meaning is judged by the human operator.

Until recently, the process of question answering was usually done by humans-experts. They "extract meaning" and "summarize" intuitively, they "survey" multiple sources based upon their instinct of relevance and their skill of generalization. If the multiple text bundling is required, of if the text compressing should be performed, people rely upon experts. When we need to use experts, and to make their labor less expensive, we often employ experts' "natural" ability to quickly compose answers, summaries, and surveys. (The terms "abridged," "compressed," "condensed" are typically understood as "summarized"). The summary of the situation, actually represented in a thoughtful answer to a fuzzy question, should give an abridged image of the essence of knowledge contained in the document. The need in the condensed "knowledge" contained in the document demonstrates our need in the meaning of this knowledge and in the validation of the results of knowledge processing..

The Existing Efforts in Joint Processes of Knowledge Compression and Question Answering (KCQA). KCQA is, in fact, the essence of the answering a question of a very vague type: "What this article (or a set of articles) is all about?" Thus, the question-answering process in numerous cases can be divided in two interrelated stages:

Stage 1. Find a package of sources relevant to the question, and

Stage 2. Categorize them, i. e. formulate, what this set of sources is all about.

It is not difficult to demonstrate that additional stages can produce further focusing of attention and end up with the regular paradigm of asking-answering. The bottom line will always be in searching for a relevant subset of sources and generating a text that could be considered a compression of the set of sources for question answering, i.e. KCQA. The latter is required in many domains starting with business of publishing and ending with funding agencies that are swamped by the overly long descriptions which should be understood and responded to. Again, an expert is the only hope. The art of summarization has not been yet formalized so that we could learn it, teach it and even more, to delegate summarization to a computer.

The existing efforts in summarization are oriented toward receiving nicely looking short statements of contents, abstracts, or summaries by the virtue of imitating prior results in summarization (the superficial "tokens" of a good summary are used). The efforts in discovering the essence of a text are dealing with the most intimate component of human information processing. Well known rules of thumb like "use first paragraph of an article as its summary" rely upon a frequent maxim of newspaper reporters to use the first paragraph as an abstract. However, in most of realistic cases this maxim fails. Usually, the intention to mimic human activities automatically lead toward cryptic, garbled, almost illegible documents where subtitles, titles of the figures, and bulletized statements are mixed together. This happens because there is no method of telling the significance of one sentence from another from the point of view conveying the meaning.

The system with KCQA employs a method of "knowledge structuring," "text compression," and even "meaning extraction" that would outline the steps of text analysis and text generation leading toward a harmonious document which could be easily understood and practically applied by the end user. In our product, text processing is based upon visualizing the structure of knowledge contained in a text as a multiresolutional web (or multiresolutional network) of text units. In order to understand the method, some preliminary information should be acquired and taken in account so that we won't need to go to the expert for explanation of words "knowledge" and "meaning" (see [1, 2]).

Novel text processing tools outlined in this paper have been developed by Cognisphere, Inc. They allow for pursuit of the meaning during the multiresolutional decomposition of the sets of texts. The *meaning explication (or discovery)* processes can be totally independent ("thesaural meaning") and can be guided by an assignment, bias, context, etc. The package developed by Cognisphere relies upon techniques for constructing the architecture of a text based upon the concept of multiresolutional text decomposition and aggregation. This concept presumes that entity-relational networks (ERNs) constructed for more simple (higher resolution) units of text are nested in more complicated units that can have a separate label.

The simplest and the most practice oriented outcomes of this development are the new tools for text compression and new text generating algorithms that can be applied in a multiplicity of the areas: for question answering, for summarizing documents, papers, books, for preparation of brief reports of meetings, for document searching in the large document bases and on the Web, and many others. This implies that the network constructed at high resolution can be substituted by a generalized but computationally simpler network constructed at lower resolution if the groups of high resolution ERNs will be considered a lower resolution units and even might be substituted by separate labels. Similar consideration can be applied to the lower resolution network and even lower resolution units can be constructed. If this process is recursively repeated bottom-up, a hierarchy of representation is obtained.

The development of compressed documents by humans is frequently considered a guesswork. In the available examples of automated summarization, the emphasis is done upon creation of a new, shorter document which will include some elements of the initial text considered its milestones: highlighted words and phrases, frequently used sentences, subtitles, pieces of the tables of contents, and so on.

The results are mostly unsatisfactory and often, very disappointing. Indeed, the summarization software packages produce at their outputs garbled texts which require strong editing - at best. As a result, all leading companies, searching the Web, have practically abandoned meaning-oriented summarization. They use "token-driven" summarization: they extract several lines from the beginning of the document, or a list of sub-titles, and so on, to give the user some hint about the text.

Joint Decomposition and Compression as Parts of General Text Analysis. The meaning-oriented text compression (e.g. summarization, or abstracting, or extracting the essence) is a sub-task of a more serious problem: to perform the text transformation and analysis that would organize the text in a system of generalized units without sacrificing the contents. We are talking about constructing *a multiresolutional system of knowledge representation* for a particular text. This can be done only by *generalizing* and subsequently, *contracting (consolidating, encapsulating)* the descriptions that are wordy and contain details of the second order of importance. Apparently, a device for the text compression should be capable of distinguishing the first order of importance (with larger, or coarser "granules" of the text) from the second order (with smaller, or finer "granules").

The term "granule" here is equivalent to the ERN unit that "has a separate meaning" and can be substituted by a separate label. By constructing granules of high resolution, then of lower resolution and so on, one performs consecutive bottom-up generalization of the text. Certainly, such generalization is different from the mechanical text filtering. It presumes constructing a new, generalized text by using words and expressions that not necessarily are the part of the document under consideration. It presumes substitution of the detailed description by metaphorical "short-hand" passages, and/or metonyms.

As a result of summarization, the user is supposed to discover within the texts the units of meaning that might be hidden *even from its author.* This can be done by putting it in a perspective of other texts which might be of interest for the user but are not necessarily known to (or taken in account by) the author. This is where the efforts in compressing the text gradually demonstrate their closeness to other important jobs in of text processing which are extremely time consuming and at present rely solely on human expertise.

It would be prudent to say that the consecutive bottom-up generalization is not a discovery of the author. The problem of compressing (abridging, generalizing, surveying, summarizing) a set of diverse statements, or documents and determining their joint meaning is well known. This problem is presently unsolved. The specifics of our approach is bundling together a multiplicity of related problems based upon similarity that can be found in their essence. Many additional jobs can be included in the problem as we visualize it. For example: development of the *group platform* of the associated documents demonstrating elements of similarity as far a particular situation is concerned. The group-platform problem is equivalent to a core problem of text processing for decision support systems.

The Central Concept of MR-Text Processing. When we are talking about text decomposition, we do not refer to the standard formal procedure of text parsing, a procedure which could rely on syntactic analysis. Certainly, the existing algorithms of syntactic parsing can be improved, some new algorithms of parsing can be created, the results of parsing can be taylored to multiple practical applications by using sets of rules which allow to notice new "tokens" of importance. These efforts for improvement parsing algorithms are very important, but they are incapable of solving the problem of *text compression via knowledge generalization, knowledge discovery,* and *knowledge mining*. In this paper, we rely on a software package that is capable of recognizing new units of knowledge that have a meaning corresponding to the meaning requested within the assignment for text processing.

Several new scientific developments are applied in this software package. One of them is a metonymic combinatorial text transformation. We employ a "multi-granular" organization of combinatorially constructed metonymic units of texts. This approach is based upon formation of the metaphors constructed via text generalization. We believe that this is potentially the most powerful mechanism of the text contraction. Finally, analysis of the structural loops gives an opportunity to discover among them the dynamic units containing new meaning.

The method is especially promising because it uses the same structure of information processing no matter what is the information medium: text, visual images, audio, etc. As the need in multimedia processing is growing, our package allows the uniform solution that can be used for improving convergence in the processes of disambiguation described later. The procedure of constructing the representation of REALITY (natural languages, visual images, audio information, physical reality) is described in [1, 2]. Entities to be encoded, put in correspondence as ERN and interpreted exist in REALITY but are not recognized and encoded.

An intelligent (human based, or automated) classifier should recognize and encode the entities. This requires transforming information into a perceivable carrier (signal). The signal inputs the system. Initially it is perceived as a "chaos." The subsequent classification is performed within the intelligent observer (our software package). Within the input chaos, the observer perceives a multiplicity of zones of with various degrees of uniformity. The observer groups them into different classes. The sets of different classes of uniformity can be thought of as singularities by themselves. Thus, the singular zones of signal uniformity in addition to singular entities are determined as a result of perception. Then, the resolution of classes distinguishing is increased, the scope of dealing with input information is reduced. What was "uniform zones" gives an opportunity to produce its further classification. The whole host of singular objects is informationally reorganized, too. As a result, new sets of objects are formed pertaining to new level of resolution. The process continues top-down. At each level of resolution there are additional singular objects: those, that has not been noticed during previous grouping processes because of their low resolution. These "left-out" entities supplement the multiresolutional system of entities that has been received. After this, a new iteration of grouping is supposed to be performed at each level of resolution[1].

The system of singular objects by itself is not sufficient for interpretation. At each level of resolution a loop of closure should be defined to perform the process of interpretation. All components of semiotic analysis (syntax,

[1] The process can be made more understandable by the following clarification: the entities that contain a meaning have more than one element: they contain information about an acting object (an ACTOR), about the ACTION produced by the ACTOR, and about the OBJECT upon which the ACTION was extended. Many entities containing experiential knowledge of this sort allow to make a generalization about a preferable rule of action in a variety of recorded situations. Thus, entities can be grouped into the *experiential* and the *normative* statements (the latter are called *rules*).

semantics, and pragmatics) should be put in correspondence with the elementary loop of functioning (ELF) defined by the closure at a level of knowledge representation [1, 2].

The circulation of knowledge within ELF is done by the virtue of communication which changes the incarnation of knowledge from a node to a node passing through the stages of encoding (in SENSORY PROCESSING), representing and organizing (in WORLD MODEL), evaluating (in VALUE JUDGMENT), interpreting, anticipating, intending, and planning (in BEHAVIOR GENERATION), generating (in ACTUATORS), applying (in the WORLD), and transducing (in SENSORS)—all considered as different forms of communication (mappings from one language to another). As something happens in the World (discourse, set of texts, additional document arrived, additional AUDIO was submitted, etc.), it is transduced by sensors into an appropriate form and the process of representation begins. The role of Perception is to represent the results of sensing in some organized manner using *signs*. This process of shaping up the organization is called Syntax. It starts at this point, it continue at all subsequent stages of dealing with Knowledge while it is more and more generalized. The initial structure becomes Knowledge as the latter gets more and more generalized so that after representation is completed, interpretation is possible. Interpretation enables the process of Decision Making including Planning within the module of Behavior Generation in which Semantics joints Syntax to create the *interpretant*.

The interpretant materializes in the process of Actuation, which is analogous to generation of new knowledge and then, in a new text. As a result of this process new Narrative arrives into the World, creates changes in the World — physically and/or conceptually. New *objects* emerge; they can be of physical and/or of linguistic nature. The sensors change their output signals and the new cycle starts of the loop of closure.

The successful functioning of the loop dwells upon creativity of Decision Making processes in the module of Behavior Generation. The hypotheses enter the subsystem of Behavior Generation as a substitute for the rules, the decision for an action is made, the action is performed, changes in the world occur, the transducers (sources of information) transform them into a form that can be used by Source Code Processing units, and the long and complicated process of moving from signs to meaning starts again. Now, the enhanced set of experiences presented in the text brings about another hypothesis that can confirm or refute the tested ones. This is when the *symbol grounding* happens.

After multiple tests, the hypotheses can cross the threshold of "trustworthiness" by constantly exercising symbol grounding, and a new rule is created. Further generalization of a rule (or a set of rules) within a particular context is considered to be "a theory". At each step of this development, the unit under consideration undergoes a comparison with other kindred units confined in corresponding databases (of Experiences, of Rules, and of Theories). Then the symbols tentatively assigned to some "unities", "entities", or "concepts" enter their place within the database of concepts (which is a relational network of symbols).

3. Texts Analysis: Decomposition and ERN Construction

Each unit of the text carries its meaning that should be interpreted within the part of context belonging to the ELF at a particular resolution. The hierarchical decomposition of context assignment is presumed. The domain assigns the context to the document (i-th level), the document in turn (within the overall domain) assigns the context to its sections ([i-1]-th level). The section (together with the whole document) assigns the context to its paragraphs. The paragraph and its neighbors-paragraphs assigns the context to its compound sentences (CS). The CS (together with other sentences around) assigns the context to its simple sentences (SS). Each SS (jointly with other SS and CS of the paragraph) assigns the context to its smaller scale components (SSC_i, I=1, 2,...). Each SSC (of this particular SS and other SS and CS of the vicinity of attention) assigns the context to its smallest SSC-units called M-seeds (the seeds of meaning).

Each M-seed (together with its neighbors) conveys the context to its words (2nd level), and each word (and its neighbors) conveys the context to its parts (1st level). We can see that the text becomes a multiresolutional ERN (entity-relational-network) which can be considered a web with interrelationships of belonging and contextual influences. This web carries meaning and interpretation and should be discovered and processed. A measure of significance can be assigned to all units of text.

This measure is called "value of significance" and is based upon the size of the unit, frequency of occurrence in the text and the quantity of associative links with other units in the text.
This measure directly affects the quality of results. The following stages perform the preliminary text analysis and transform the narrative of the input natural language text into the multiresolutional hierarchy of knowledge representation.

Stage A1. Consecutive decomposition of the narrative into the nested multiresolutional system of ERNs. A system of tokens was developed based upon conducting consecutive decomposition of English texts.

There are evidences that similar tokens can be developed for other languages, too. The system is similar to the one utilized for visual images and is adequately represented by Figure 1.

Stage A2. Top-down and bottom-up conducting of the process of *disambiguation* (see [3])which is supposed to end-up at each level of resolution either by converging or by generation of an inquiry in the form of a question or additional text request.

Disambiguation procedures are based on libraries of rules that reflect the formation of gestalt-routines known for a particular domain of activities and/or discourse. It is based upon formulating hypotheses and verifying them at the adjacent levels below and above [3]. We have developed a package of rules for a linguistic disambiguation for a particular type of activities (e. g. summarization). Since the premises are general, similar set of rules can be developed for other assignments, too. The loops of disambiguation exercise simulation of applying at levels (i+1) and (i-1) the function that was hypothesized at the i-th level (see Figure 2).

Stage A3. Putting in correspondence the result of Stages 1 and 2 with the knowledge architecture of the domain of interest; tracing the initial narrative within the joint knowledge architecture. Thus, within the same multiresolutional architecture, a multiplicity of various texts (narratives) can be represented by corresponding strings of pointers without changing the architecture.

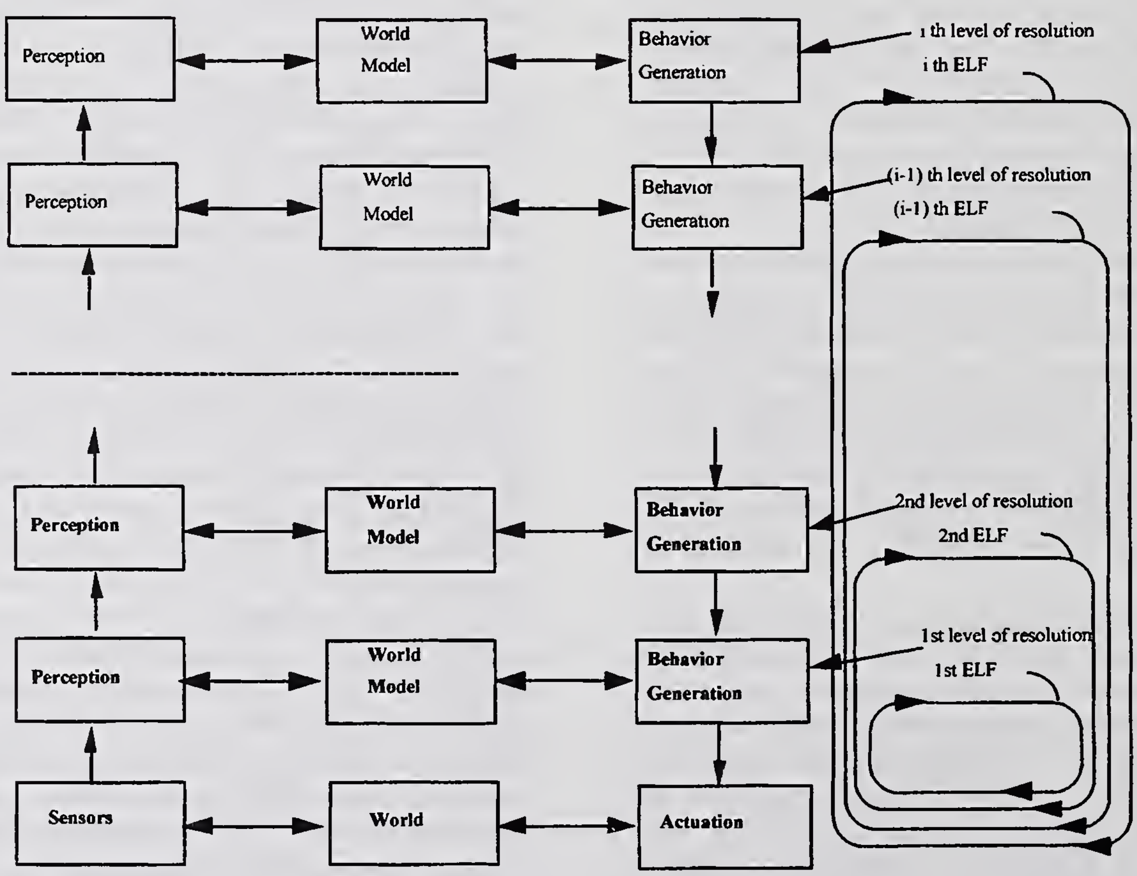
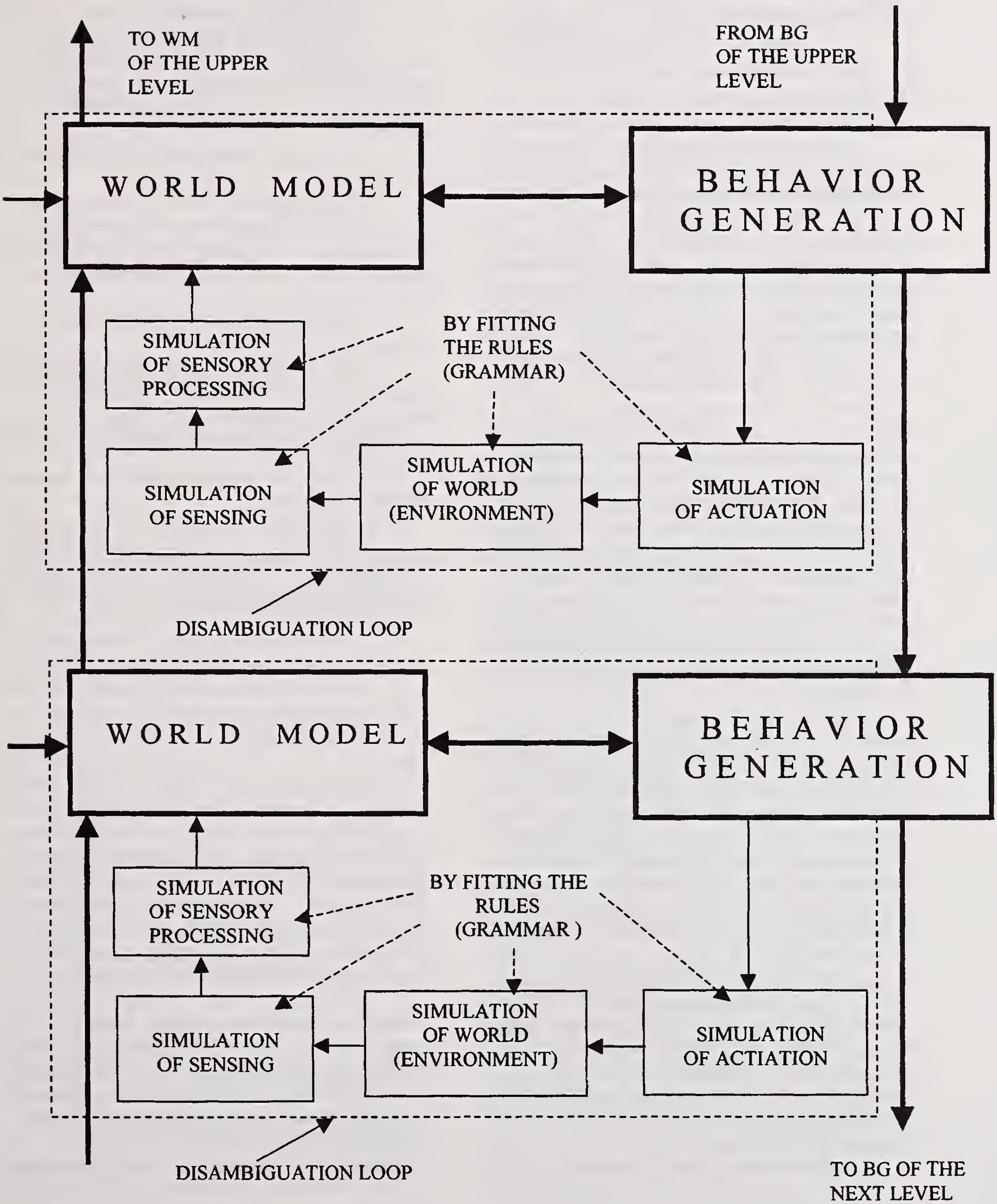


Figure 1. Multiresolutional text organization

Figure 2. Multiresolutional processes of disambiguation



As these three stages are completed, the multiresolutional ERN knowledge base is considered to be constructed.

4. New Text Generation: How Do We Receive the Answers

This multiresolutional ERN knowledge base is used for new text generation under a multiplicity of particular assignments: e.g. to construct summary, abstract, abridged text, summary upon the multiplicity of texts, survey of multiple documents, etc. The idea of new text generation is based upon the opportunity of constructing most probable ELFs out of available components.

The following stages should be performed for the new text generation:

Stage G1. The level of resolution are to be selected at which the expected text should be generated. Sometimes, the particular indications are given that determine the user's preference toward chosen particular aspects of the domain of discourse. In these cases, the values of significance are increased correspondingly for related units stored in the knowledge base. The pointers for tracing the narrative at this level are enabled, and the output text is generated by following the string of these pointers as shown for Stage G2.

Stage G2. The pointers are followed and the narrative is generated. The richness of detail of the output is determined by the levels of resolution selected for text generation. This procedure invokes several rules of text generation that allow for associating simple sentence components (SSC) with Actor, Action and Object of Action. These rules should be applied either prior to text generation or as a part of its process:

a) Generation of Generalized SSC_i

In all sentences, substitute SSC_i (or n units of SSC_i) for the GL-SSC_I (generalized label SSC_i) Replace the whole SSC_i with its generalized label, in a manner such that it's possible to go back (to recognize, what was in place of generalized SSC_i label and substitute it back to the original set of words).

b) Generalized SSC_i Clustering

Group together simple sentences with the same Generalized SSC_i. The clusters of Generalized SSC_i should be marked by their relative location in the sentence.

c) Categorizing the SSC_i Clusters

Recognize the groups of Generalized SSC_i Clusters related to actors, objects of action and actions. The groups should be marked by their relative location in the sentence and form an ERN.

d) Mergers within the Action related SSC_i Clusters

For a cluster of Action related groups, check against significant M-seeds on intersections. Temporary unify intersecting clusters, mark their relative location in the sentence.

e) Mergers within the Actor related SSC_i Clusters

For a cluster of Actor related groups, check against significant M-seeds on intersections. Temporary unify these clusters, mark their relative location in the sentence.

f) Mergers within the Object of Action related SSC_i Clusters

For a cluster of Object of Action related groups, check against significant M-seeds on intersections. Temporary unify these clusters, mark their relative location in the sentence.

f) Construct graphs for all resulting sentence structures for visual analysis

(An easily interpretable example of the graph is demonstrated in Figure 3)

g) Order the Graph as the Original Text Flow

Conduct permutations: start with arranging with Actor related SSC, follow with object of action related SSC, make intervals for permutations required (if necessary). Different graphs will be obtained for different size of the M-seeds and for different value of significance of them. The quality of the newly constructed ELFs is determined by the values of probability the new ELFs entail at all levels of resolution.

Using the ELF-based Activity Graph. The graph is a powerful additional tool for conducting the text interpretation. In the package by Cognisphere, the following opportunities of using the graph representation are exercised for the compressed texts generated at the output:

• Read the flow of connections from left to right by using balloons, or a window for displaying alternatives.

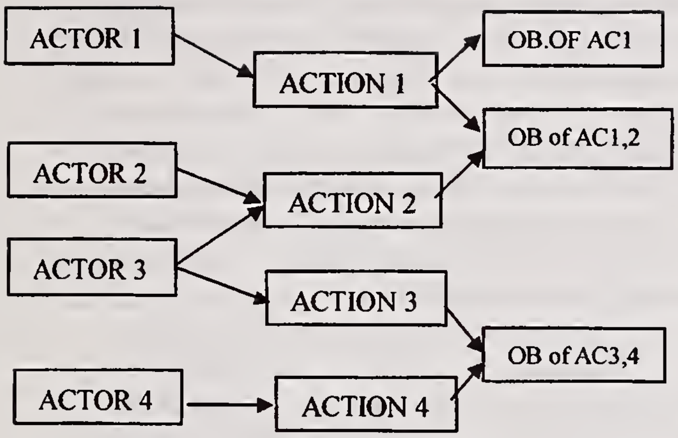


Figure 3 Graphs of Output Formation

• Request for evaluation of probabilistic validity for the triplets Actor-Action-Object of Action.

Transform the Graph into Text to be Generated. The process of transformation comprises the following steps: a) substitute all SSC_i by their original sets of words from the original text; b) sequence the sentences along (parallel) with the original text pointers tracing; c) sequence the sentences along with the original text pointers tracing; d) form paragraphs when the adjacent sentences do not intersect. The software package uses a proprietory set of rules for Output Text Generation; the rules are taken from the human experience of text analysis. Some examples of rules are given in this list:

- When two consecutive phrases have the same Actor and the number of words in their Action SSC_i exceeds that of the joint number of words in [ActionSSC_i + Object-of Action SSC_1] then unify them into one sentence with the structure: Actor SSC_i + (Action SSC_1 + Object-of-Action SSC_i)$_1$ + (ActionSSC_i +Object-of-ActionSSC_i)$_2$
- When two consecutive phrases have the same Object-of-Action SSC_i and the number of words in this Object-of-Action SSC_i exceeds that of the joint number of words in Actor SSC_i + Object-of-Action SSC_i, unify them into one sentence of the structure: (Actor SSC_i +Action SSC_i)$_1$+(Actor SSC_i +Action SSC_i)$_2$+Object-of-Action SSC_i.
- When two consecutive phrases have the same Action SSC_i substitute in the second sentence this Action SSC_i by the corresponding "Generalized Action SSC_i."

5. Research and Development Perspectives for the Evaluations

The techniques introduced in this paper can be applied for a cluster of activities. All of them are unified by the focus of analysis rather unusual for the engineering endeavor. Cognisphere, Inc. calls these activities Meaning-Oriented Analysis of Text Sets (MOATS). "Texts" can be explained as narratives representing REALITY (i. e. *descriptions*). Before using constructively, the descriptions should be mapped into a different structure: an MR-Natural Language Text Architecture (MR-NLTA). This construction uses the following elements as its building blocks:

- natural language passages including "factual," generalized, labeling,
- numerical data (explicit, implicit, tabulated, etc.), sometimes with related interpretations
- formal logical constructions based upon standards and conventions related to a particular discipline, or domain of knowledge
- pictures and graphs with, or without related interpretations
- complex structures of presentation encompassing all of the above elements

Our familiarity with the existing research results allows us to be optimistic in our evaluation of the advantages of the proposed system. The existing competitors do not seem to be able to achieve results similar to those we can provide within the scope of our proposal, since they have not yet incorporated the multiresolutional technology of text processing.

Presently, there are no methods of testing for system of text processing with summarization and other, more sophisticated intelligent capabilities of processing. Both the formation of the test-set (of texts to be processed), and the methodology of testing, including the interpretation of the results, are obscure issues today. Most of the sources attribute the skill of summarization to the most intimate faculties of human intellect. "Which one of two summaries, prepared for the same text, is good and which one is not?" is the question we intend to answer as a part of the testing methodologies that will include the following directions.

Direction 1. Analysis of Meaning and Consistency

Both, summarization and abstracting answer an instantaneous need in newly generated documents with a pertinent (but not necessarily deep) meaning. In fact, the results of our text processing allow for expanding beyond the initial target to prepare a relevant compressed version, categorize it, and find a relevant list of keywords. Additional opportunities comprise:

a) Determining of Clusters of Meaning

After determining hierarchical networks of semantic fields, numerous clusters of them emerge which are more informative than it is required by the typical task of summarization.

b) Interpreting Additional Messages

These semantic fields contain island of additional meaningful "messages" conveyed by a text, or by a set of documents.

c) Recognition of Hidden Problems.

The lack of consistency in a semantic network at one or more levels of resolution speaks for the existence of hidden problems (in the text and/or in the real world described within this text).

d) Planning of Actions

Determining the course of actions, which can be recommended for dealing with the hidden (and recognized) problems.

Since all these operations are substantiated algorithmically, the numerical measure ("metric") can be introduced for judging the quality of results. If additional considerations can be introduced by human operators, they can be taken in account in formalizing the metric only if they cannot be incorporated into the algorithm.

Using these operation presumes a preliminary process of learning of the system functioning with parallel human based evaluation of results in a variety of situations.

Direction 2. Visual Support of Meaning

It is our observation, and it is part of the practice of decision-making organizations that both formal models and linguistic descriptions are not fully instrumental in conveying the meaning. Numerous additional issues and components of the meaning are illuminated when the decision-maker is given an opportunity to put together a visual representation for the meaning. We are not talking about graphs and other visual tools of supporting a presentation when numerical data are give, or qualitative results allow for some quantitative representation. We are talking about some intrinsic capabilities of the conceptual essence of MOATS.

The tools of text processing employed by MOATS allow for creation of visual images that have the same spatial and temporal structures as the soft model of the text has. The visual primitives are selected from the table of correspondence between the concepts and percepts (a tool seeking for syntactic and morphological resemblance between conceptual and perceptual units is under development). These visual primitives are being organized into a multigranular structure similar to the one extracted from the texts [4].

As a result, a report of the Mutual Funds Headquarters might be mapped into a visual image where in the midst of the multi-color ornament a several salient objects demonstrate some persistent (and predictable) motion: several polyhedra are quickly rolling around a deformed, oscillating egg-shaped body with a fuzzy contour. Visualization appeals to the intuitions of the decision-maker affecting his perception of the descriptive units of meaning obtained as a result of the prior analysis. It can be used for evaluation if interpretation tables are composed at the preliminary stage of situation learning.

Direction 3. Extraction and Analysis of Kindred Texts Packages

Analysis of large data-bases of text presumes browsing all documents, when the assignment is given to find a subset of them related to a particular issue. This issue is presumed to be represented not by the set of key-words and key-expressions but rather as a description of a particular situation. Even more challenging is a problem of extraction of subsets of kindred documents when the issue of interest is not specified but should be discovered. MOATS has all prerequisites required for solving this problem in the future.

Analysis of sets of kindred documents (articles related to each other) is performed as follows: documents are processed together (in parallel), and the meaning, hidden problems and inconsistencies are determined for the set as a whole.

These functions will include (but not bounded to) the following list of activities:

- creation of abstracts and lists of key-words for all kinds of written texts
- determining and interpretation of text statistics including

a) construction of Zipf's and Zipf-Mandelbrot's laws

b) finding statistics of parts of speech and phrases
c) computing N-grams
- composing lists of the natural language passages containing "facts," generalizations, labels , and other predetermined types of expressions
- extraction and organization of all available numerical data (explicit, implicit, tabulated, etc.)
- extraction of formal constructions based upon standards and conventions related to a particular discipline, or domain of knowledge
- development of abridged documents and compendiums
- development of pictures and graphs reflecting the abridged documents
- preparation of complex structures of presentation encompassing all of the above elements

Services based upon the MOATS system will take advantage of a possibility to interaction with the user. Thus, it will be possible to take into account both the goal of the user in its various aspects, and the variety of meanings that is (and/or possibly can be) conveyed by these texts as detected by the operator.

Certainly, the learning period is required when test text will be submitted to the system as well as the results of performance evaluation done by human operators (a representative statistics of operator evaluations is presumed). The learning cycle of the MOATS system contains the following components:

- receiving representative texts as input
- discussion with the user the required personalized features of the job assignment and the output form
- texts processing using both conventional and innovative techniques (described above)
- composing summaries, abstracts, surveys, compendiums, etc. fitting within preassigned specifications
- composing a list of keywords (the number of them can be preassigned)
- categorizing texts for both cases: with a preassigned classifier, or without it
- evaluating the results within a particular category by using the algorithmic "metric"
- evaluating the same results by a number of individuals considered experts in this particular category of meaning
- constructing an ontology terms database and monitoring its subsequent use for the user's needs; evaluating the ontologies algorithmically
- evaluating the same ontologies by a number of individuals considered experts in this particular category of meaning
- answering the user's questions concerning with the text and with the system's functioning; evaluating them by jurors and algorithmically
- formulating the meaning of the texts and hypothesize on its extension; evaluating them by jurors and algorithmically
- proposing explanations for the issues of interest; evaluating them by jurors and algorithmically
- discovering and explicating hidden problems within the world represented in the test set of texts and outlining the contradictions within these texts; judging the automated results
- constructing and regularly updating a knowledge base for a particular user; judging the automated results
- supplementing text processing with tools of visualization for enriching the results of interpretation and meaning analysis; helping the user in analysis of images
- outlining alternative actions for dealing with the problems and contradictions found in the text

As a service tool, MOATS processing is specialized to perform the above functions of evaluation regularly in response to the needs of a user and for verifying whether the tuning of the system has a favorable dynamics.

References

1. A. Meystel, *Semiotic Modeling and Situation Analysis: An Introduction*, AdRem, Bala Cynwyd, PA, 1995
2. A. Meystel, J. Albus, *Intelligent Systems: Architectures, Design, Control*, Wiley, NY 2001
3. A. Meystel, "Multiresolutional Representation and Behavior Generation: How Do They Affect the Performance of Intelligent Systems" (in this volume).
4. V. Grishin, A. Meystel, "Evaluation of System Intelligence Via Pictorial Data Visualization," Eds. A. Meystel, E. Messina, Measuring the Performance and Intelligence of Systems: Proceedings of the PERMIS'2000 Workshop, Gaithersburg, MD, August 14-16, 2000, pp. 533-540

APPENDICES

APPENDIX A
WORKSHOP SCHEDULE

PerMIS'2001

The Workshop on Performance and Intelligence of Intelligent Systems is conducted in association with CCA/ ISIC'2001, in collaboration with DARPA in Mexico City, Mexico, see: http://www.control.rice.edu/

T3 _ Measuring Performance and Intelligence of Intelligent Systems (PERMIS'2001) **Tue., Sept. 4, 2001 09:00_18:00**
Organizers: E. Messina, NIST, A. Meystel, Drexel University, NIST, L. Reeker, NIST

Tuesday, September 4, 2001

09:00 - 10:00
Multiresolutional Representation and Behavior Generation: How Do They Affect the Performance of Intelligent Systems (Lecturer: A. Meystel)

10:00 - 12:00
Mathematical Aspects of Performance Evaluation (Chair: A. Meystel)

V. Kreinovich, U. of Texas, El Paso, TX, R. Alo, U. of Houston Downtown, Houston, TX
"Interval Mathematics for Analysis of Multiresolutional Systems"
D. Repperger, AF Research Laboratory,
"An Autonomous Metric (Polytope-Convex Hull) For Relative Comparisons of MIQ"
D. Repperger, AF Research Laboratory
"Decision-Making and Learning - Comparing Orthogonal Methods to Majority-Voting"
J. Shosky, American University
"A Top Down Theory of Logical Modeling"
C. Landauer Aerospace Corp.
"Implementing and Evaluating Intelligent Systems: The Need For New Mathematics" (paper not available for publication)
E. Dawidowicz, US Army
"Performance Evaluation of Network Centric Warfare Oriented Intelligent Systems"

12:00_13:00
Break

13:00 - 15:00
Testing For Performance Evaluation (Chair: E. Messina)

H. Yanco, U. of Mass, Lowell
"Designing Metrics for Comparing the Performance of Robotic Systems in Robot Competitions"
A. Jacoff, E. Messina, J. Evans, NIST
"Experiences in Deploying Test Arenas for Autonomous Mobile Robots"
A. Lacaze, NIST
"Hierarchical Architecture for Coordinating Ground Vehicles in Unstructured Environment"s
E. Messina, J. Evans, J. Albus, NIST,
"Evaluating Knowledge and Representation for Intelligent Control"
A. Meystell, J. Andrusenko, Drexel U.
"Evaluating the Performance of *E-coli* with Genetic Learning From Simulated Testing"

15:00 - 17:00

Performance Evaluation in Non-numerical Domain (Chair: L. Reeker)

L. Reeker, A. Jones, NIST
"Measuring the Impact of Information on Complex Systems"
R. A. Pease, Teknowledge Corp.
"Evaluating of Intelligent Systems: The High Performance Knowledge Bases and IEEE Standard Upper Ontology Projects"
P. Davis, J. Bigelow, RAND Corp.
"Meta-models to Aid Planning of Intelligent Machines"
A. Meystel, Drexel U.
"Performance Evaluation in Computing with Words"

17:00 - 19:00
General Panel Discussion:

Why Should Performance Evaluation in Intelligent Systems Be Different?

Panelists: J. Albus, E. Messina, A. Meystel, L. Reeker, D. Repperger