NISTIR 7951

A Vision of Cyber-Physical Cloud Computing for Smart Networked Systems

Eric Simmon
Kyoung-Sook Kim
Eswaran Subrahmanian
Ryong Lee
Frederic de Vaulx
Yohei Murakami
Koji Zettsu
Ram D. Sriram

http://dx.doi.org/10.6028/NIST.IR.7951

NICT National Institute of
Information and Communications Technology

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

NISTIR 7951

A Vision of Cyber-Physical Cloud Computing for Smart Networked Systems

Kyoung-Sook Kim
Ryong Lee
Yohei Murakami
Koji Zettsu
Universal Communication Research Institute
National Institute of Information and Communications Technology
Kyoto, Japan

Eric Simmon
Eswaran Subrahmanian
Frederic de Vaulx
Ram D. Sriram
Software and Systems Division
Information Technology Laboratory

http://dx.doi.org/10.6028/NIST.IR.7951

August 2013

U.S. Department of Commerce
Rebecca Blank, Acting Secretary

National Institute of Standards and Technology
Patrick D. Gallagher, Under Secretary of Commerce for Standards and Technology and Director

Executive Summary

This report is the initial outcome of a joint project between the National Institute of Information and Communication Technology (NICT), Japan and the U.S. National institute of Standards and Technology (NIST). A team of researchers from both organizations collaborated to conceive the Cyber-Physical Cloud Computing (CPCC) architectural framework presented in this report. One of the motivating factors for this research was the earthquake and tsunami that hit Japan in March 2011 and the resulting damage. The magnitude of the Japanese earthquake and tsunami highlighted the importance of a robust and reconfigurable disaster recovery systems. Recent advances in information technology provide an opportunity to create smart networked systems for power grids, transportation, and healthcare systems that will enhance existing disaster management solutions. Smart networked systems and societies will result from the emerging network of people, intelligent devices, and mobile personal computing and communication devices (mPCDs).

There are five technologies that are core to the concept of a Smart Networked Systems and Societies (SNSS): 1) networked computer systems, 2) real-time systems, 3) wireless sensor/actuator networks, 4) social networks and, 5) cloud computing services. A CPCC architectural framework **--** defined as “*a system environment that can rapidly build, modify and provision auto-scale cyber-physical systems composed of a set of cloud computing based sensor, processing, control, and data services”* -- integrates the characteristics of CPS and cloud computing into a single framework, and is a first step in achieving the SNSS vision.

The CPCC architectural framework supports the realization of an SNSS and the appropriate services to users and applications at the appropriate place and the appropriate time via the appropriate devices. This framework supports the deployment of large-scale and data-intensive systems, involving complex distributed decision-making. The benefits of the CPCC framework include: efficient use of resources, modular composition providing customizability, rapid development and scalability, smart adaptation to environment, scalable reliability, resiliency, and performance based on user needs.

The report describes two component systems developed at NICT in Kyoto, Japan that serve examples of existing systems that can easily be modified to support the concepts described in the CPCC architecture framework. The systems are: CPSenS (an information technology platform for on-demand integration of real-world sensing information with cyber cloud systems) and K-L Grid (an information service development platform for utilizing big data).

In conclusion, this report identifies challenges and research areas that need to be addressed to realize the CPCC architectural framework. The research areas include: the virtualization of sensors and actuators, interconnectivity between CPCC services, data integration, knowledge generation, resource orchestration, security, privacy, performance, reliability, resiliency, and metrology.

1. Introduction

1.1 Background

The computer has slowly been changing our relationship with the world. Today, computers are an integral part of everyday life. Our interaction with desktop systems, mainframe computers, mobile devices and embedded systems has led to increased productivity and better quality of life. With the emergence of social networks, we will witness a new computing paradigm that will combine social networking with the Internet of Things. This will have significant implications for both the market for advanced computing and communication infrastructure and the future product and service markets (for nearly 4.5 billion people) that these net-centric societies will create.

Historically, computing systems could be organized into two groups: traditional mainframe and desktop systems for data and information processing, and computer systems for controlling physical systems. While in the past these types of computer systems were independent of each other, present computing systems are being designed to interact with both the physical world and with humans simultaneously. This is enabled by the fact that networked computers are everywhere, not only in the form of personal computers but also in the form of cell phones, tablets, and smart appliances. In fact, almost everyone has a computer at home and a cell phone that they carry, both of which are networked. Smart power meters on houses have built-in sensors that are connected to the utility systems. Cameras on roads and in public spaces are web accessible. The devices mentioned above interact with each other on a regular basis and deliver information to people and other devices whenever they might need it. Terms such as ubiquitous computing, pervasive computing, and Internet of Things (IoT) have been used to describe this technology since it was envisioned as a new paradigm in computing in the early 90's [1]. Recent advances in social networking allow humans to share their interpretation of real world events. But taken together, the interlinked networks of sensors, actuators and processing devices create a vast network of connected computing resources, things and humans that we will refer to as a Smart Networked Systems and Societies (SNSS).

At the same time computing devices have become ubiquitous, our need and desire to use them has grown as well. Future, interconnected systems such as the smart power grid, traffic management, event management, smart healthcare systems, and many others need not only large amounts of real world data, but also a large variety of data. This data can be used for advanced, multivariate control and data analytics, such as providing accurate weather forecasts days ahead or integrating variable renewable-energy sources into the power grid. These advanced systems help us use resources more efficiently, decrease costs, and enhance our lifestyles.

A key aspect of the next generation SNSS is the bridge between the physical and virtual world. These systems that tightly interlink the cyber and physical worlds are often referred to as Cyber-Physical Systems (CPS) [2]. A CPS is the "integration of computation with physical processes" and uses sensors and actuators to link the computational systems to the physical world. The modern CPS inspires a new vision of "computing as a physical act" where the real world is monitored through sensors that transfer sensing data into the cyberspace where cyber applications and services use the data to affect the physical environment in real time. When combined with data provided by humans, CPS can be described as eco-social systems, with information feedback loops among various interconnected devices and systems (including humans) in the cyber and physical worlds (shown in Figure 1). Future CPS will help people to understand the physical environment and adopt the optimal action in a wide range of applications: healthcare, transportation, energy consumption, manufacturing, agriculture, disaster response, critical infrastructure, and business advertisement. Many computer scientists and engineers recognize the fundamental technological challenges [3] in implementing CPS infrastructures.

There are five technologies that are core to the concept of an SNSS: 1) networked computer systems, 2) real-time control systems, 3) wireless sensor/actuator networks, 4) social networks, and 5) cloud computing services.

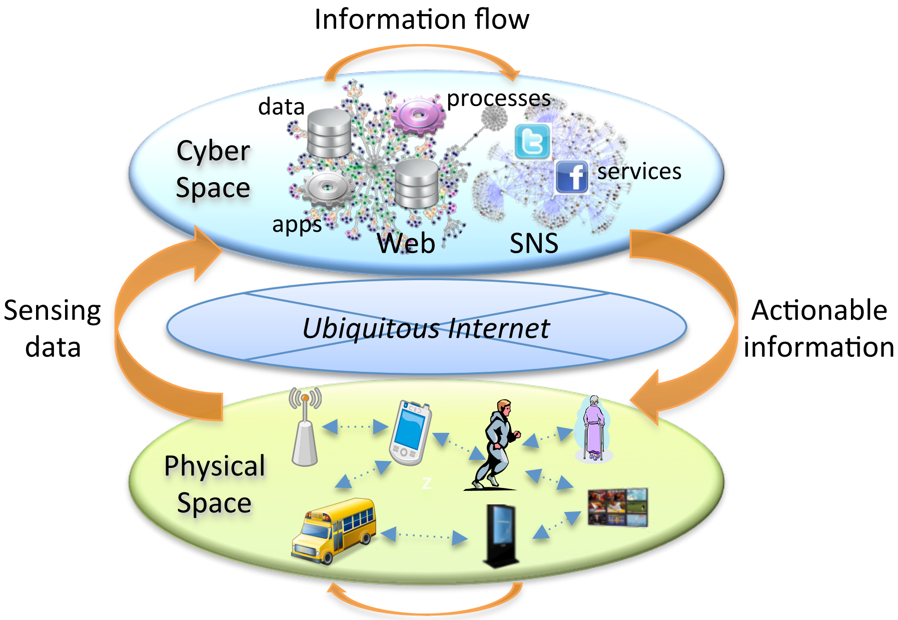


Figure 1 - Information Flow from the Physical to the Cyber to the Physical World

- ***Networked Computer Systems*** have become part of our everyday lives. Having not just desktop computers, but mobile devices, appliances, sensors and actuators networked is fundamental to the smart networked society. The networks themselves must be adapted as well. Not only are availability and bandwidth important, but latency and jitter become critical as well. The next generation networks must address these aspects to better support closed loop control systems.

- ***Real-time Control Systems*** are used in communication systems, aircraft or automotive control systems, home appliances, and so on. Most current systems are closed systems and restrict their capability at a specialized scale of networks and computations; i.e., small self-contained systems. In most cases, the individual components in a given SNSS are only used to a small fraction of their capabilities. Current SNSS are tightly coupled and are vulnerable to unexpected situations that were not considered in the design and development phases of the system. If these components can be made available to be used in other systems the overall efficiency can be greatly improved. While they are carrying out their own operations, SNSS can integrate and cooperate with each other in order to achieve a task and purpose. This can reduce the cost to build a new system, provide new data that is presently unavailable, and, minimize the effect of unexpected failures and uncertainties both on cyber and physical levels.

- ***Sensor/Actuator Networks*** are networks consisting of geographically distributed autonomous sensor and actuator devices (nodes) that cooperatively monitor and control physical or environmental conditions in many application areas, including environment and habitat monitoring, military applications, and personal healthcare systems. Widely available sensor networks enable instant dissemination of processing, gathering, and sharing of a variety of sensory information (such as temperature, sound, vibration, pressure, motion or pollutants, at different locations). For constantly monitoring and processing well-structured sensing data on special-purpose networks, wireless sensor network devices usually take into account energy-efficient networking and self-organizing methods. However, recent advances in pervasive computing, especially smart phones and wireless communication, facilitate the dynamic creation and distribution of information about real world observations created by people in the form of text messages, photos, and videos. Human beings participate in the sensor networks as virtual sensors and their experiences and interactions are transparently integrated into both the virtual and physical worlds. Consequently, a sensor network is required to understand human behaviours (mobility) and handle the large volume of structured and unstructured contents possibly generated by huge number of opportunistic users and devices. Furthermore, increased diversity and numbers of devices and users causes many problems in interoperability and scalability.

- ***Social Networks*** provide the ability for humans to provide information to the SNSS infrastructures for real-world information gathering and interpretation alongside sensor networks. In particular, social networking services allow people to share, store, and access their real-life experiences with linked friends at any time and any place using networked desktops or mobile devices that also feature various sensors such as GPS, voice recorders and cameras. Every mobile device is capable of sensing physical characteristics of the environment and providing human interpretation of the environment. The cyber-physical-social system is inevitable for the integration of humans and cyber-physical systems as discussed in [4, 5]. The use of data generated from a group of humans (a society) causes problems of reliable interpretation of the social data and its integration with traditional, more restrictive, sensor information.

- ***Cloud Computing Services***[1] provide a flexible platform for realizing the goals of the SNSS. The National Institute of Standards and Technology (NIST) defines cloud computing as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." Applying the characteristics of cloud computing to SNSS, any device (sensor, actuator, data archive, computation resource) is available as a service and can rapidly be provisioned for use in a virtual SNSS to meet the user demands. Cloud-based services allow SNSS devices to be used in multiple virtual systems at the same time providing more efficient use of those resources, resiliency during unexpected situations, and flexibility to change and expand as users needs change.

- ***Humans*** are not a technology, but are an integral part of the SNSS. Instead of only interacting with a black box as with traditional systems, Humans may provide data, process data, make decisions, and act on the data outputs.

1.2 Goals

As mentioned in the previous section, SNSS aim to support our life activities by connecting humans and cyber systems with physical objects through sensors and actuators. These new types of systems will promote technological innovation and improve the quality of everyday life. The terms "smart environments" or "smart things" describe the advanced capability of SNSS, which has been tabbed as smart systems. In Reference [6], "smart" is used to imply the ability to autonomously acquire and apply knowledge, while "environment" refers to our surroundings. A smart environment improves experience through smart devices that continuously monitor and collect data about inhabitants' surroundings and adjust the environment to better meet the needs of the inhabitants. This technology has begun to pervade all spaces including the home, office, and commercial buildings. The technology enables smart environments by connecting computers, sensors, and appliances inside these spaces. In the future, the technology will be implemented on a global scale.

The SNSS is not only about the communication of numerous devices to exchange information, but also creating knowledge from the information and allowing devices and people to collaborate. In the future, we will evolve toward a global smart environment that will include the integration of heterogeneous smart spaces (e.g., homes, buildings, roads), smart societies (e.g., humans, vehicles, sensors), and smart services as shown in Figure 2.

[1] The NIST Definition of Cloud Computing is available at, http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. Last accessed, 19 November 2012.

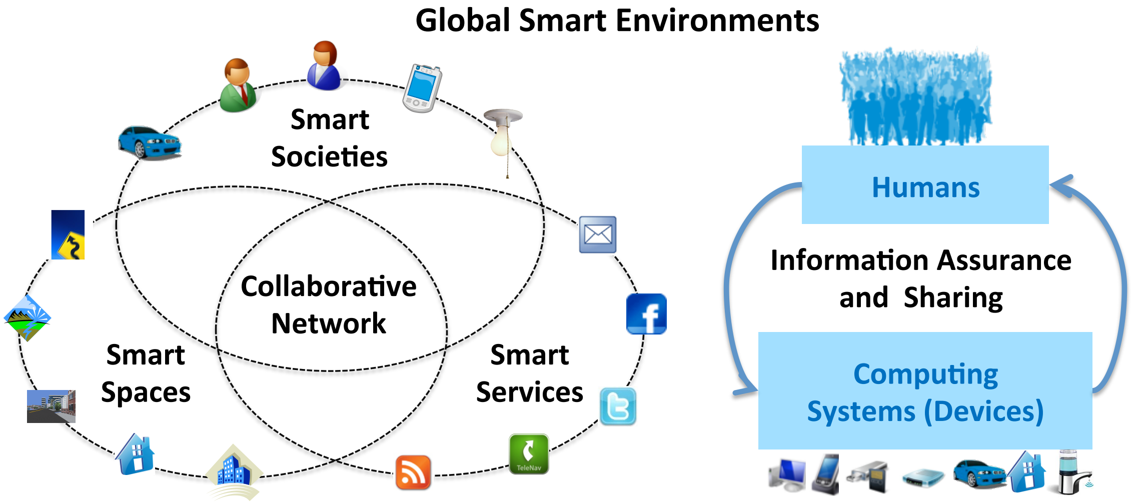

Global Smart Environments
Smart Societies
Collaborative Network
Smart Spaces
Smart Services
Humans
Information Assurance and Sharing
Computing Systems (Devices)

Figure 2 - Global Smart Environments - Collaboration of SNSS, Societies, and Services by Sharing of Information

Current SNSS have different designs, architectures and implementations, although they have the same purpose such as interoperability, flexibility, large-scale systems, distributed control, safety and security. For example, the smart grid is designed to alleviate stress on the electric grid; and, the smart charging of electric vehicles is regarded as one of the killer applications of the smart grid.

In global smart environments, an SNSS can share its components with other SNSS and replace a new resource at runtime when a subsystem is broken. Moreover, heterogeneous components can be integrated to form a new system while they are cooperating with existing systems. The interaction between SNSS is usually continuous, but also opportunistic to meet the emergent behaviour over the Internet (Figure 3).

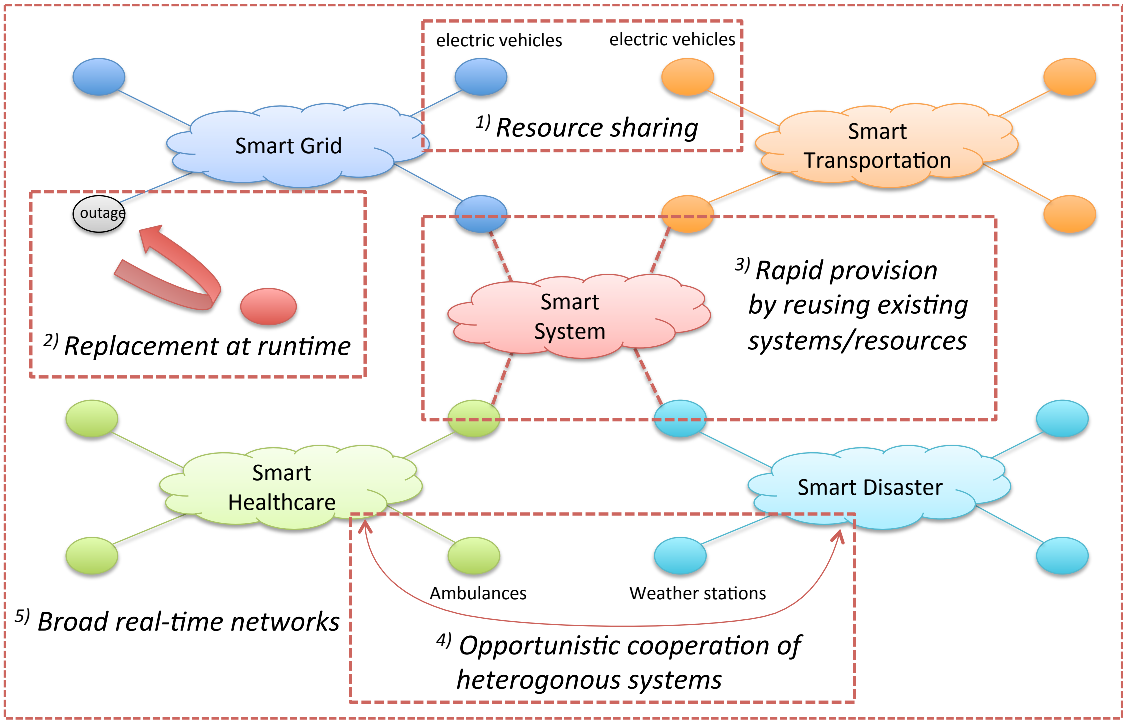

electric vehicles
electric vehicles
1) Resource sharing
Smart Grid
Smart Transportation
outage
2) Replacement at runtime
Smart System
3) Rapid provision by reusing existing systems/resources
Smart Healthcare
Smart Disaster
Ambulances
Weather stations
5) Broad real-time networks
4) Opportunistic cooperation of heterogonous systems

Figure 3 - Infrastructure for Smart Disaster Management

1.3 Cyber-Physical Cloud Computing (CPCC)

To implement the SNSS shown in Figure 3, there is a need to redesign the supporting infrastructure. We will discuss one such architecture that integrates CPS and cloud computing – Cyber-Physical Cloud Computing (CPCC) -- in this section. In a future report, we will discuss other architectures for merging CPS, social networks, and cloud computing.

1.3.1 Definition

An architecture for CPCC systems supporting SNSS combines the characteristics of cloud computing with cyber-physical systems as shown in Figure 4. Therefore, a Cyber-Physical Cloud Computing (CPCC) architectural framework is defined as "*a system environment that can rapidly build, modify and provision cyber-physical systems composed of a set of cloud computing based sensor, processing, control, and data services.*"

CPCC provides the technology for the realization of SNSS and also provides the appropriate services to users and applications at the appropriate place and the appropriate time via the appropriate devices.

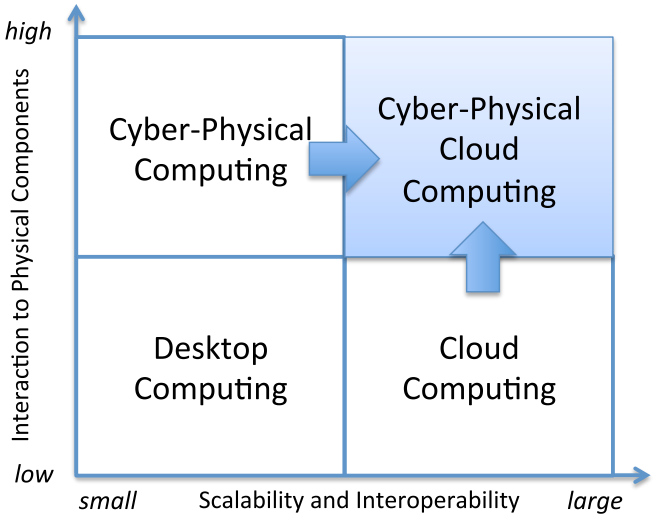

high
Interaction to Physical Components
low
small
Scalability and Interoperability
large
Cyber-Physical Computing
Cyber-Physical Cloud Computing
Desktop Computing
Cloud Computing

Figure 4 - CPCC in a New Computing Era

1.3.2 Benefits of Cyber-physical Cloud Computing

CPCC has the following benefits:

- Efficient use of resources
- Modular composition
- Rapid development and scalability
- Smart adaptation to environment at every scale
- Reliable and resilient

These benefits are elaborated further in the following sections.

1.3.2.1 Efficient Use of Resources

According to Reference [7], current approaches to create SNSS are costly and take a long time to design, build, test, and maintain. This is due primarily to the intricate composition of networked software and hardware components and the rigid structure of the systems. Even though there are common, cross-cutting aspects, domain stakeholders have independently built specialized infrastructures and standards for applications within their domain - for instance, energy, transportation, healthcare, and manufacturing. As a result, new problems including overlapping investments, interoperability, and productivity arise when multiple infrastructures need to interact and coordinate. Additionally, most components of CPS systems (along with traditional computing systems) are often underutilized. By allowing SNSS resources such as sensors and actuators to be shared among processes and systems, CPCC can offer much higher utilization, resulting in higher efficiency.

1.3.2.2 Modular Composition

The CPCC framework provides components that are service based, which makes the resulting systems flexible and modular. These modular components can be combined in different ways to create a variety of systems customized to individual stakeholders needs and supporting innovation.

1.3.2.3 Rapid Deployment and Scalability

The CPCC framework provides the ability to build and deploy systems quickly and easily. The infrastructure of CPCC will orchestrate multiple and cross-disciplinary systems that pool their resources and capabilities together to create a new smart system when needed and release system resources when no longer needed.

1.3.2.4 Smart Adaption to Environment

Context-awareness is one of the main attributes of modern devices and services that will improve the autonomous behaviours of systems by providing suitable information and services based on context. With them, we can extend the definition of context from "location, identities of nearby people and objects," [8] to "information that can be used to characterize the situation of an entity, such as a person, place, or object that is considered relevant to the interaction between a entity and an application system." In global smart environments, the context is not just the interaction between a user and an application but also includes humans, services, machines, and systems. A CPCC system in a global smart environment will collect data from the real world and understand the data relative to its current context. And, it may take action automatically to affect the state of the environment based on this knowledge and the cooperation with other entities (also having the same context awareness).

1.3.2.5 Reliable and Resilient

The ability to dynamically change the resources used by the system will help ensure the resulting systems are reliable (ability to prevent failure) and resilient (ability to recover after a failure).

The rest of the report is organized as follows:

- Section 2 describes high-level applications for CPCC
- Section 3 describes requirements for CPCC
- Section 4 describes an architectural framework for CPCC
- Section 5 describes the NICT Cyber-Physical Sensor Information System
- Section 6 discusses issues in creating and maintaining CPCC
- Section 7 discusses areas of research for CPCC

2. Applications, Configuration and Use Cases for CPCC

CPCC as an IT infrastructure allows for the creation of more-robust, resilient, and, flexible systems that can withstand natural and man-made catastrophes by allowing distributed and moveable computing resources. In the following sections, we identify the types of applications well suited to CPCC. Subsequently, we illustrate a CPCC for a smart disaster management system, and discuss a use case in disaster management.

2.1 Example High-Level CPCC Scenario

In this scenario, cyber-physical cloud customers access resources with common interfaces via the Internet independent of location, devices, and vendors. CPCC manages the resources across sensor, actuator, processor and, data services transparently and automatically.

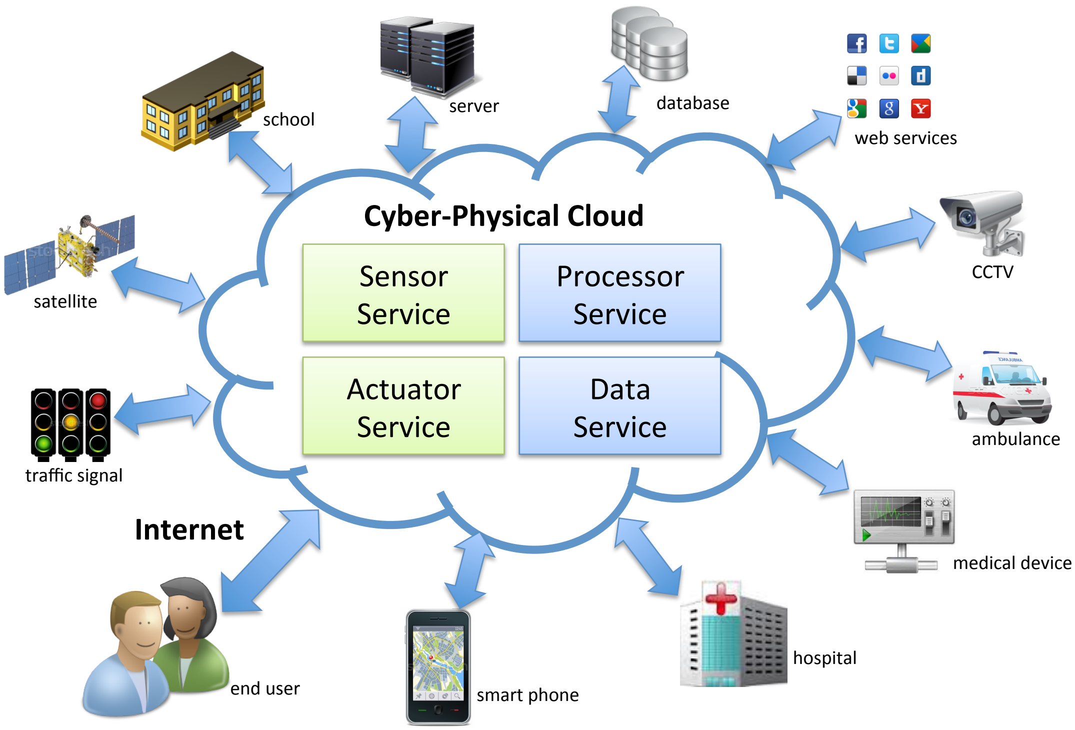


Figure 5 - Connecting Cyber-physical System to the Cloud

Figure 5 shows the meta-level scenario of CPCC with cyber and physical cloud resources. There are two customers: system builder and user. In this scenario, a system builder defines a system configuration and submits it to the cyber-physical cloud platform for provisioning. The platform creates the virtual system architecture based on the model and maps the necessary cloud services by referring to service catalogues. Then, the platform initializes the system in the cloud, and connects the system with constituent services that can interact with each other. The platform monitors the runtime status of the systems and adjusts them as needed, such as service replacement and resource expansion. The platform allows adding new functions or modifying the system configuration during operation.

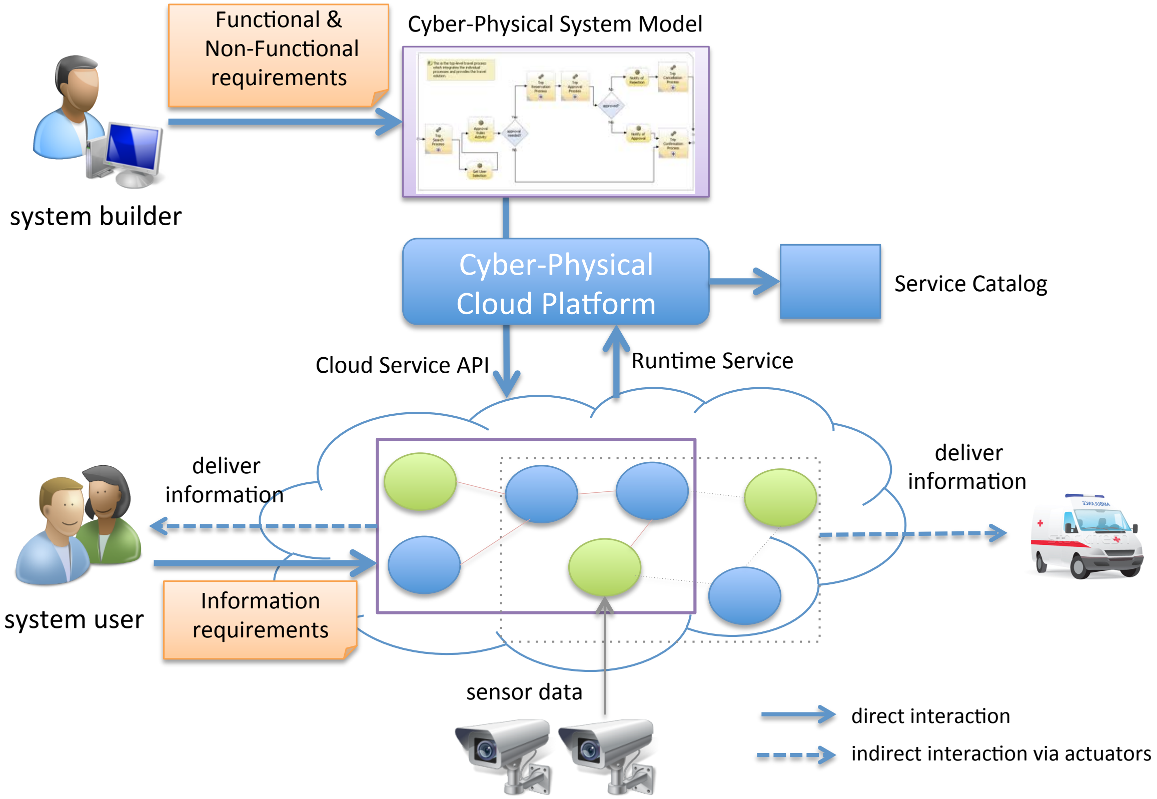


Figure 6 - High-level CPCC Scenario

2.2 CPCC Applications

This section describes proposed uses of CPCC. These application areas are, more often than not, large-scale and data-intensive, involving complex distributed decision-making. This paper will describe the applications and use-cases only to the extent needed to develop the requirements for CPCC. The incorporation of cloud computing into CPS for SNSS is important to allow for resiliency in data recovery, robustness of operations, and location-independent storage and computing.

Examples of systems that CPCC will benefit include:

- **Power grid**: A smart grid combines a networked IT system with the electric power system to achieve greater system monitoring and control. Its goals are to enable more reliable grids, bi-directional power flow, smarter protection systems, better power flow optimization, and sustainable energy systems. Smart grid is a networked system connecting a large number of heterogeneous devices (sensors, computing resources, controls) and subsystems that interact to achieve these goals. Presently, the main focus of smart grid is interoperability - connecting individual subsystems to other subsystems to enable very specific functionality. This focus is necessary due to the legacy equipment already installed on the power grid. All the complex components in the power grid can be broken down into sources, loads, and the devices that transfer the power from the sources to the loads. These are connected to an IT system (providing monitoring, analysis, and control) through sensors and actuators. Therefore, a smart grid is a Cyber-Physical System. By implementing the smart grid using a cloud-based approach, the sensors and actuators can allow reconfiguration of the grid in the case of failure. This allows systems to be adaptable as the power network changes over time, and it allows us to create a new knowledge and control system beyond what is presently being implemented.

- **Traffic Management**: Congestion is a major problem for transportation systems. A goal of the smart traffic systems would be to control traffic movement by monitoring traffic flow, controlling traffic signals, and providing navigation aids such as alternate routes to individual vehicles. The gathering of data on traffic can occur in a number of ways including sensors in the cars, closed-circuit television cameras, sensors on the road, and through inter-vehicle communication. This data can be used to rearrange traffic patterns, thus alleviating traffic congestion in a particular area. The system can operate on a push or pull mode. In the push mode, the system will inform the driver to take an alternate route. In a pull mode, the driver requests information and makes a decision based on that information. Based on the nature of the request, the system will mobilize its resources in terms of the sensor data and other prior data that will be available in the cloud, and provide the requested information in an appropriate manner to the driver. Traffic congestion is a combination of small-scale, local events (such as accidents or too many vehicles using the same road resource) and large-scale, widespread events (such as weather conditions). The data requirements vary based on whether the events are localized, large-scale, or a combination. Under these conditions, the data and processing required to provide advice would require a combination of broadcasting messages to all vehicles as well as sending specific instructions to targeted vehicles. In this application therefore, there would be a need to allocate computing and storage resources from multiple locations to generate the right advice to the driver and also to specific traffic signals.

- **Healthcare**: There are many potential applications in the healthcare domain. The following use case demonstrates one SNSS application in healthcare [9]. In this use case, the tightly connected (networked) medical devices, services, patients, and doctors access and share the patient's electronic health record, and aid in the detection of the abnormal state of the patient. Most such systems are closed localized systems of care. Even within these configurations the interoperability of these devices is questionable. Nevertheless, the ability to collect sensor data on the medical condition of the patient along with other data including the health record and the hospital layout could provide a basis for delivering smart healthcare in a hospital. This requires the integration of multiple data sources, their access and processing at different locations in a co-ordinated manner. Cloud technologies become critical in managing these computational and sensing resources.

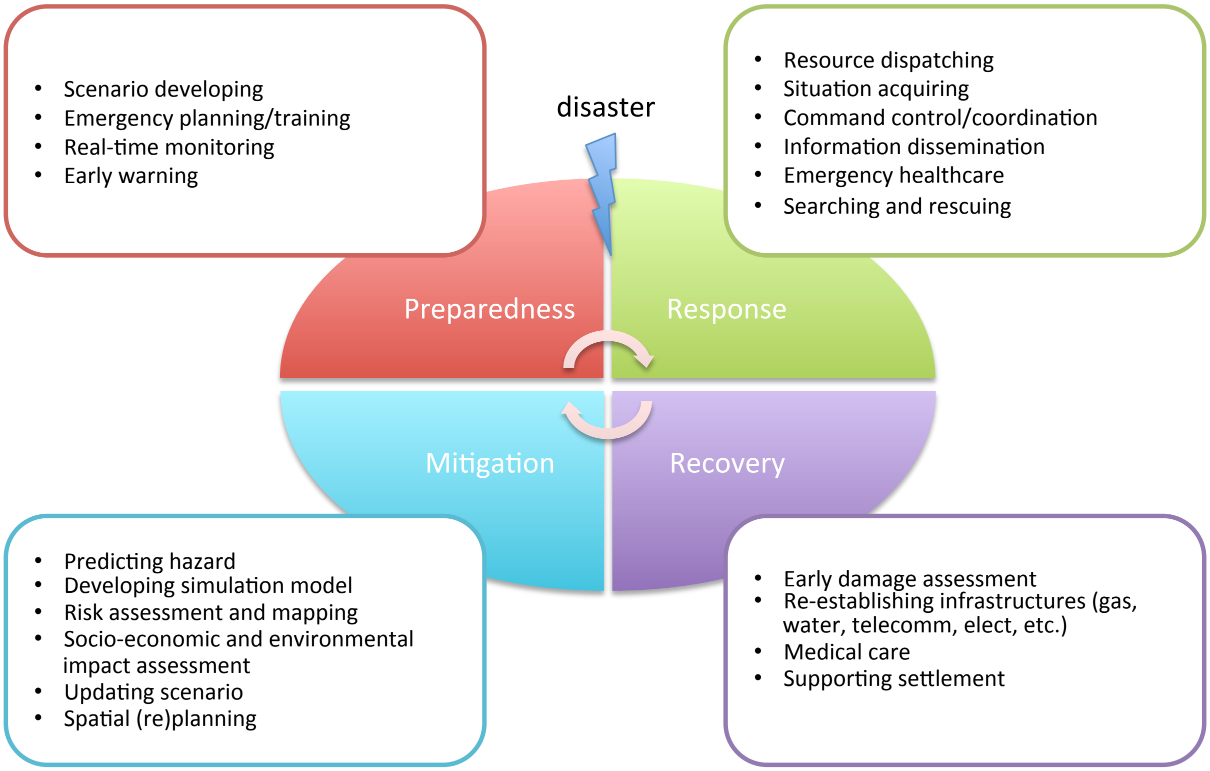
2.3 Disaster Response

The disaster management system can be one of several application types based on CPCC or a combination of application types. It needs the interaction and collaboration of multiple heterogeneous computing systems and human systems under the critical situation. The main goals of disaster management are to:

- Reduce, or avoid, losses to life and property relating to hazards such as earthquakes, tsunamis, landslides, wildfires, and air pollution events, and even human-induced disasters
- Assure prompt assistance to victims including rescues and emergency healthcare services
- Achieve rapid and effective recovery of effected environments

Figure 7 shows an overview of disaster management consisting of four phases: 1) mitigation, 2) preparedness, 3) response, and 4) recovery. All phases aim to minimize the effects of disaster, such as to reduce hazards, anticipate disaster scenarios, and prepare for emergencies. However, it is difficult to prepare for all actions and responses to various ranges of a disaster from heavy rains to hurricanes, especially an unexpected disaster. It is very expensive to develop, reproduce, distribute, and maintain big disaster management systems. In practice, all actions are not performed at all times and not all stakeholders are involved at all times. Depending on agencies, individuals, and events, disaster management systems are required to support different functions.

The flexible and scalable CPCC system can offer considerable benefits. The ability to reconfigure data, resources, and communications on the fly for monitoring and analysis before and during the course of a disaster will allow a system builder to quickly build out the disaster management system. Japan incurred enormous damages when an earthquake and tsunami occurred in a boundary zone of the northeast coast on March 11, 2011. Many critical infrastructures including hospitals, roadways, communication stations, and nuclear power plants were destroyed and about 20 thousand people were killed or injured during the disaster. The people in the affected areas are still struggling to come to terms with the effects of the disaster. A reason for the tragic loss of lives in Japan was the lack of preparation against tsunamis, compared to the preparation for the more frequent earthquakes. Usually, tsunami warnings are broadcast indicating the size and affected areas following an earthquake. These warnings are still not timely enough to evacuate people from affected areas before disaster strikes.



- Scenario developing
- Emergency planning/training
- Real-time monitoring
- Early warning

disaster

- Resource dispatching
- Situation acquiring
- Command control/coordination
- Information dissemination
- Emergency healthcare
- Searching and rescuing

Preparedness
Response
Mitigation
Recovery

- Predicting hazard
- Developing simulation model
- Risk assessment and mapping
- Socio-economic and environmental impact assessment
- Updating scenario
- Spatial (re)planning

- Early damage assessment
- Re-establishing infrastructures (gas, water, telecomm, elect, etc.)
- Medical care
- Supporting settlement

Figure 7 - Disaster Management Phases

On March 11, the people in the coastal areas near the epicentre had less than 30 minutes to move to safe evacuation places. However, 30 minutes would have been enough time to evacuate, if people had been fully aware of the situation. According to the reports of survivors, people did not hear or they ignored the emergency announcement that a tsunami was coming even after the structural damage of the earthquake (e.g. blackouts and destruction of housing) had begun. It is only after they saw the rising waves and flooding did they realize the severity of the situation. There had been no way to communicate the severity to others in a timely fashion to evacuate to the right place. The general information provided in the radio, TV, and other media were not localized (prefectures) enough to be useful to those affected in a timely manner.

The survey data in [10] show the different effects of the tsunami on the topology of the region. In a certain southern part of the Tohoku region, especially in the flatlands, the tsunami reached more than 5 km inland and caused a different kind of devastation. This example shows the need to take into account different sets of data (topography, population density and other data) for each region to provide the most appropriate response in contrast to the general warnings. However, it is almost impossible to prepare disaster management systems that consider all different scenarios.

2.3.1 Use Cases

Use cases are used to show how a user or actor interacts with a system or subsystem to achieve a goal. The UML use case diagrams[2] provided below show the various actors using a system and the use cases within each system that the individual actors use. This use case describes an example of using the CPCC infrastructure to build systems for supporting activities of stakeholders that participate in early step of the emergency response after a disaster warning. We describe the use case in terms of components of the systems starting with geographical scope of action (Geo-fencing), the main actors, and the subsystems in the early stages of a disaster.

2.3.2 Geo-Fencing: Deliver Local Information from Global Monitoring

The concept of a geo-fence is a virtual or pre-defined perimeter for a real-world geographic area such as a school zone, military district, or neighbourhood boundaries according to Wikipedia[3]. In recent years, many mobile applications make use of geo-fencing functionalities to trigger certain information when a user enters or exits a particular geographic location or area. For example, parents can get the notification when their child enters or leaves a dangerous or a defined area via short message service (SMS) or email. Another example would be for the system to send a reminder for specific events or tasks when a user is within a predefined distance from a given location. Geo-fencing is a powerful way to provide users with the regional information about disaster situations. The problem is we cannot predict exactly where or when a disaster will occur. Here, CPCC can help to relate the geo-fence to each user quickly and to provide useful services for the user autonomously based on this relationship. Figure 8 shows how to deliver information depending on the geographic information of users.

[2] http://www.altova.com/umodel/use-case-diagrams.html

[3] http://en.wikipedia.org/wiki/Geo-fence, Last accessed, 19 November 2012

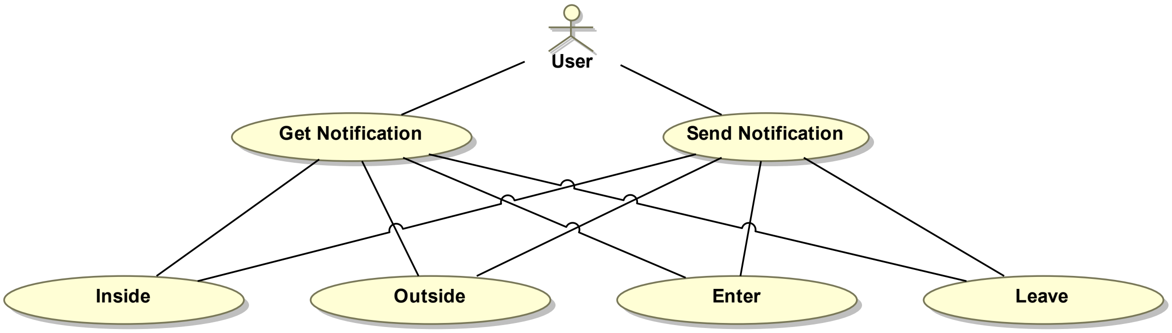


Figure 8 - Geographic Location Based Information Delivery

2.3.3 Main Actors

Figure 9 shows a scenario with five actors: 1) telecommunication companies, 2) citizens, 3) first responders, 4) hospital staff, and 5) control centers, and four examples of CPCC systems supporting the scenario: 1) Emergency Evacuation and Rescue Systems, 2) Person Find Systems, 3) Emergency Health Care Systems, and 4) Emergency Telecommunication Systems. All actors should get the notification of disaster warning regardless of their location. Each subsystem will rely on its own cyber-physical configuration. A centralized static CPS would not work well, since the complexity of data and processor allocation will become too complex to handle. In particular, in a disaster management environment, each CPCC system must be provisioned according to the need and the location of the service it provides. While they may be provisioned independently, these subsystems will co-operate together.

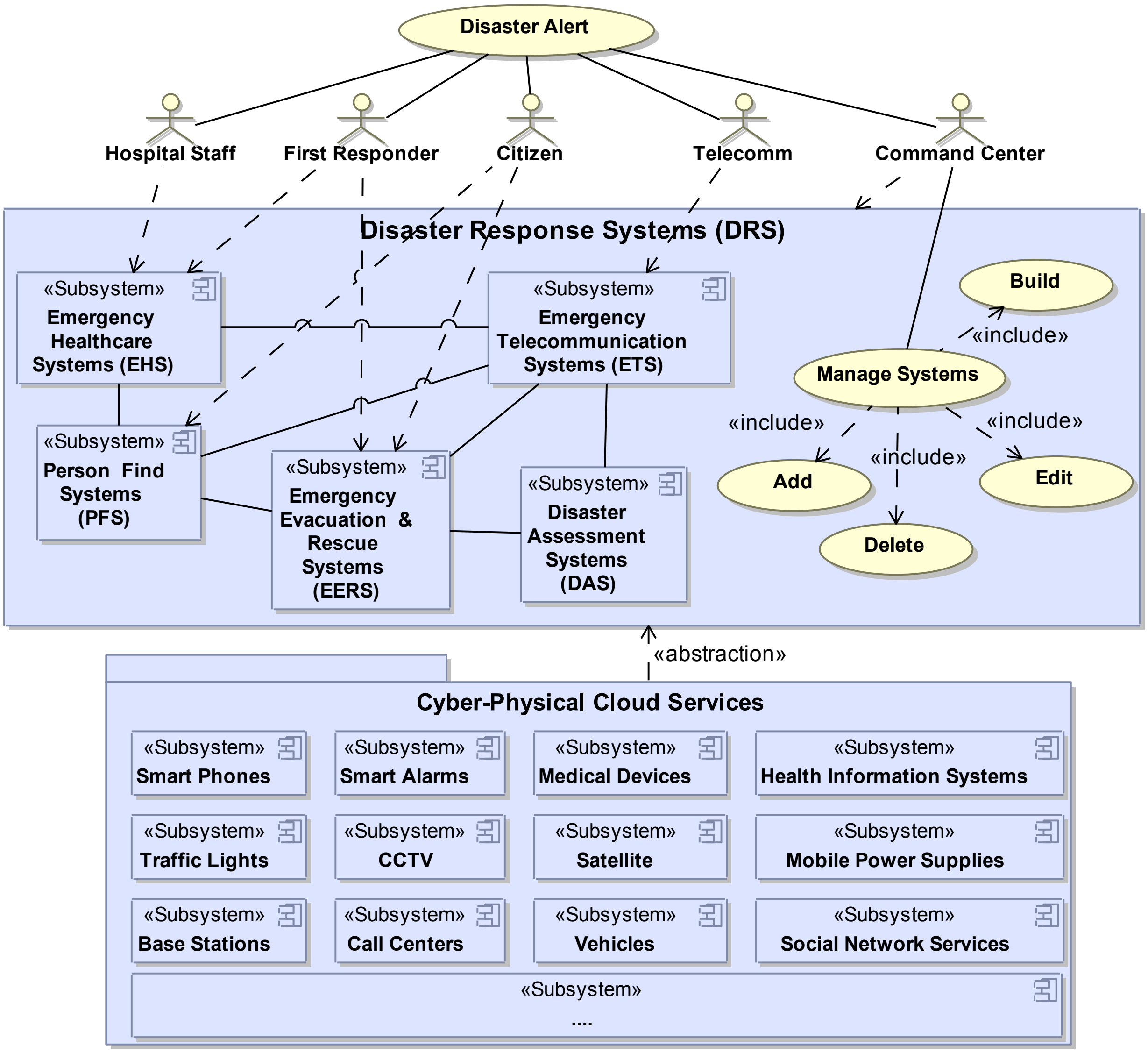


Figure 9 - Main Actor and CPCC Systems in Disaster Response Scenarios

2.3.4 Emergency Evacuation and Rescue Systems (EERS)

First, we consider a CPCC system for evacuation of people and animals and to help the rescue activities of first responders. In this scenario, three actors directly interact with the provisioned cyber-physical cloud services configured as shown in Figure 10. The command center is responsible for building the system by configuring functional and non-functional requirements and setting events, conditions, and actions to describe geo-fencing. After finishing the configuration, the CPCC platform starts the provisioning work based on cloud services. According to the definition of event, condition, and action of a geo-fence, the platform automatically coordinates the types of sensors, processors, actuators, and data cloud services and carries out the system test before providing the service to the users. The smart phones or other devices the citizens and first responders use inside the disaster area work as sensors and actuators in the system. Citizens are categorised as inside citizens (inside the geo-fence) and outside citizens (outside the geo-fence) because they have different requirements depending on a geo-fencing area, such as a tsunami warning area.

The evacuation system interacts with only citizens inside the tsunami warning area. For example, the inside citizens may get the information to evacuate from the danger and can provide messages to guide people to a safe place using information about the general disaster situation and the local surrounding environment. When a citizen sends a report, the system shares the information with the "Person Find Systems," which is explained in the next section. An outside citizen could then access the "Person Find Systems" to locate their friends or loved ones.

In our scenario, the system using CCTV, other sensors, and processing can detect when someone is injured and dispatches the most appropriate first responder (based on location, skills, equipment, etc.) Once a responder accepts the request to move to the location of the injured person, the cloud services compose the required service links such as the telecommunication systems for a priority communication link. The system will send the location and route to the responder and then co-ordinate traffic lights so the responder can arrive quickly at the injured person's location. During the course of this rescue operation, sensors monitor any unexpected events and the required services are appropriately configured on the fly.

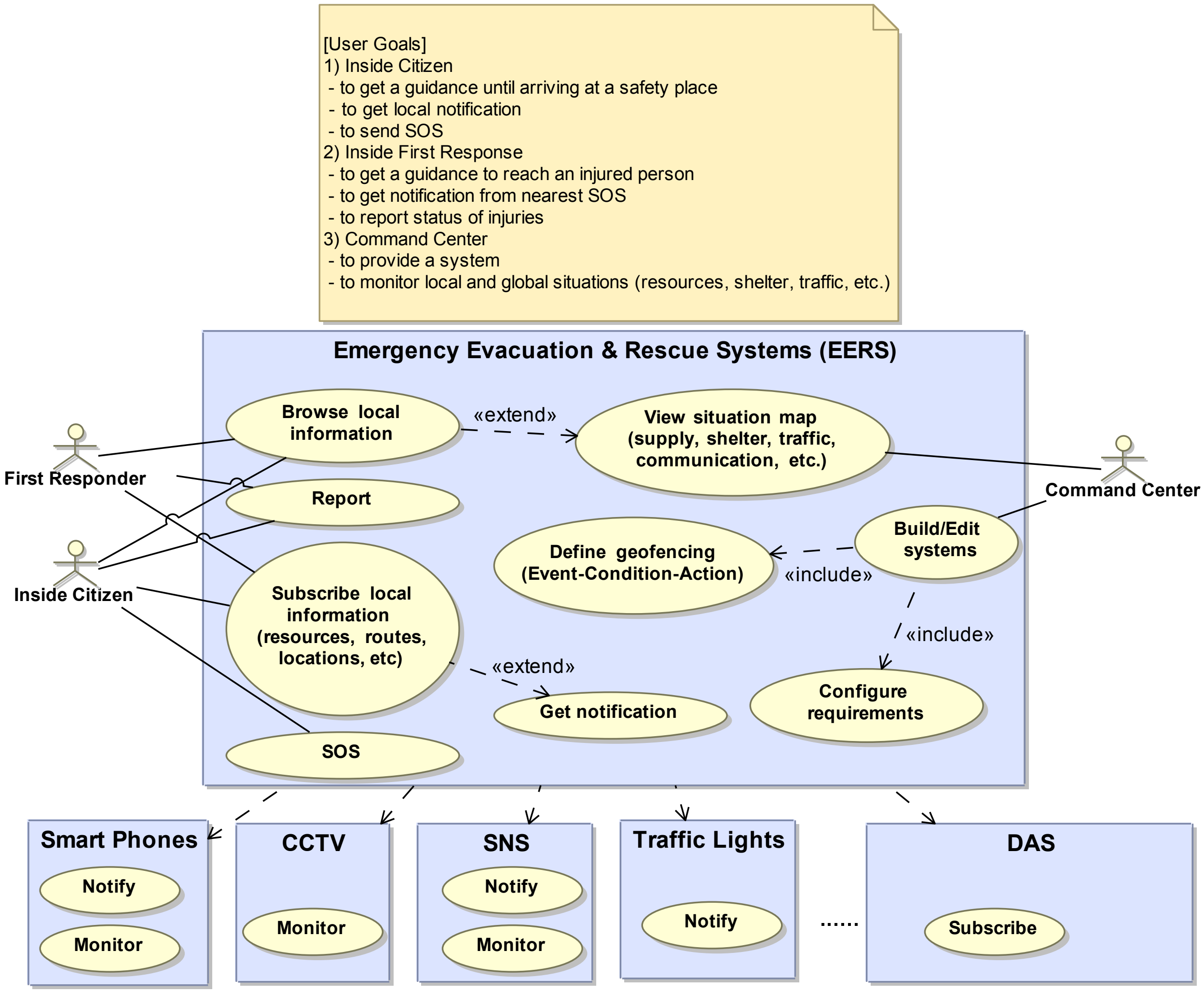

[User Goals]
1) Inside Citizen
- to get a guidance until arriving at a safety place
- to get local notification
- to send SOS
2) Inside First Response
- to get a guidance to reach an injured person
- to get notification from nearest SOS
- to report status of injuries
3) Command Center
- to provide a system
- to monitor local and global situations (resources, shelter, traffic, etc.)
Emergency Evacuation & Rescue Systems (EERS)
First Responder
Inside Citizen
Command Center
Browse local information
«extend»
View situation map (supply, shelter, traffic, communication, etc.)
Report
Build/Edit systems
Define geofencing (Event-Condition-Action)
«include»
«include»
Subscribe local information (resources, routes, locations, etc)
«extend»
Get notification
Configure requirements
SOS
Smart Phones
Notify
Monitor
CCTV
Monitor
SNS
Notify
Monitor
Traffic Lights
Notify
......
DAS
Subscribe

Figure 10 - Use Case Diagram of Emergency Evacuation and Rescue Systems

2.3.4 Person Find Systems (PFS)

The experience of the Japanese natural disaster in 2011 has illustrated the important contribution of the social media and crowd sources. During the same incident, the geo-location information could not be used effectively in the rescue operations. This condition was made apparent when right after the earthquake, large numbers of people were unsuccessful in their attempt to make phone calls and send an SMS to contact loved ones due to heavy traffic congestion and power outages. Interestingly though, social media and crowd-sourcing helped people to search for lost families, and access and disseminate vital information via web-enabled phones and smartphones. Social networking sites (e.g., Twitter, Facebook) and crowdsourcing websites (e.g., Google Person Finder, Ushahidi) played a significant role as a source of timely and detailed reports regarding disaster events and as a novel tool for the information sharing. However, one major problem is that these systems operate independently; i.e., they provide a platform for people to share information while not connected with other systems. Additionally, these social networking systems tend to spread false rumours rapidly during the disaster. The diffusion of false rumours may aggravate the situation and could have disastrous consequences. To prevent such outcomes there is a need to aggregate, verify and filter information that is gathered through crowd-sourcing methods. Figure 11 shows a system configuration to address this problem, which will aid in locating people and informing them and their families about their safety conditions during a disaster.

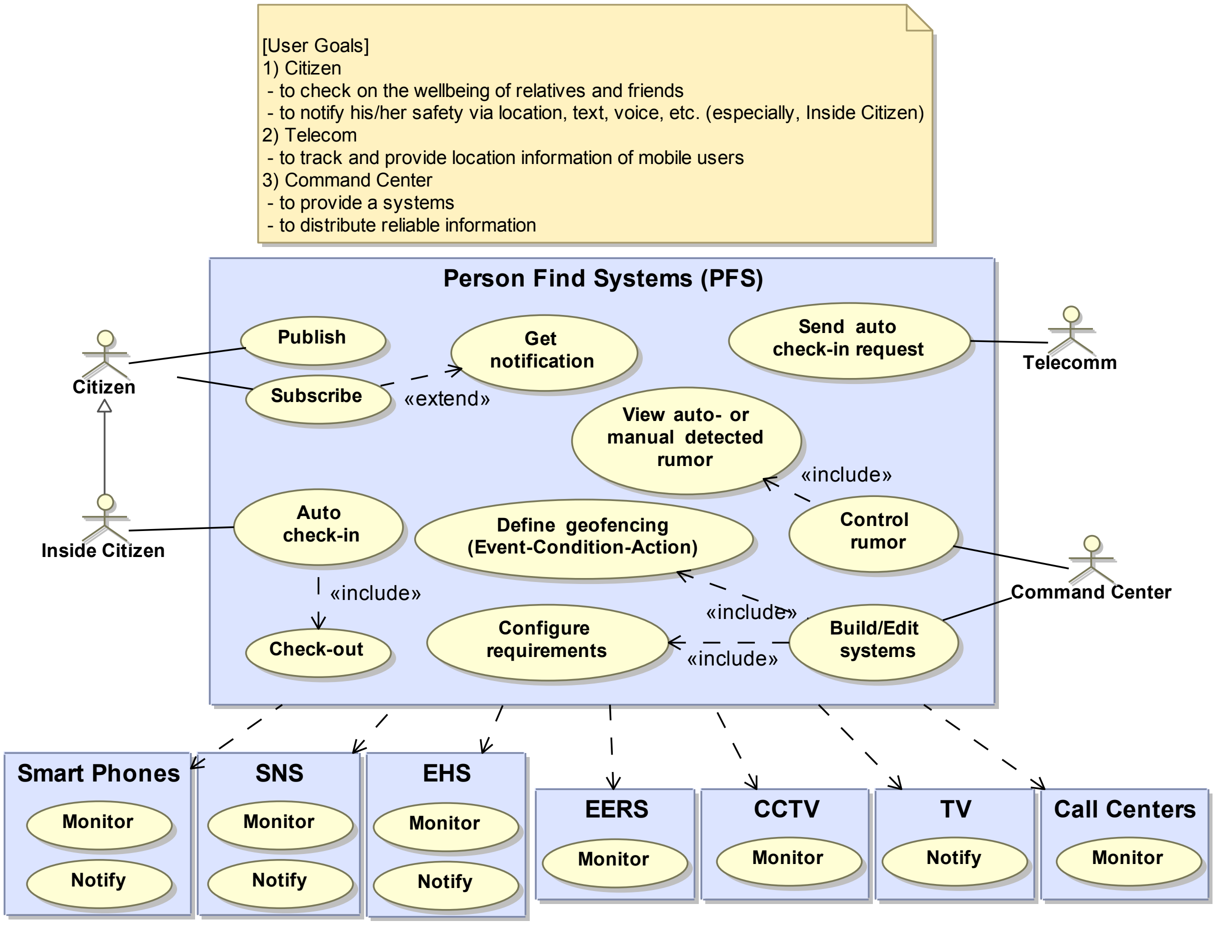
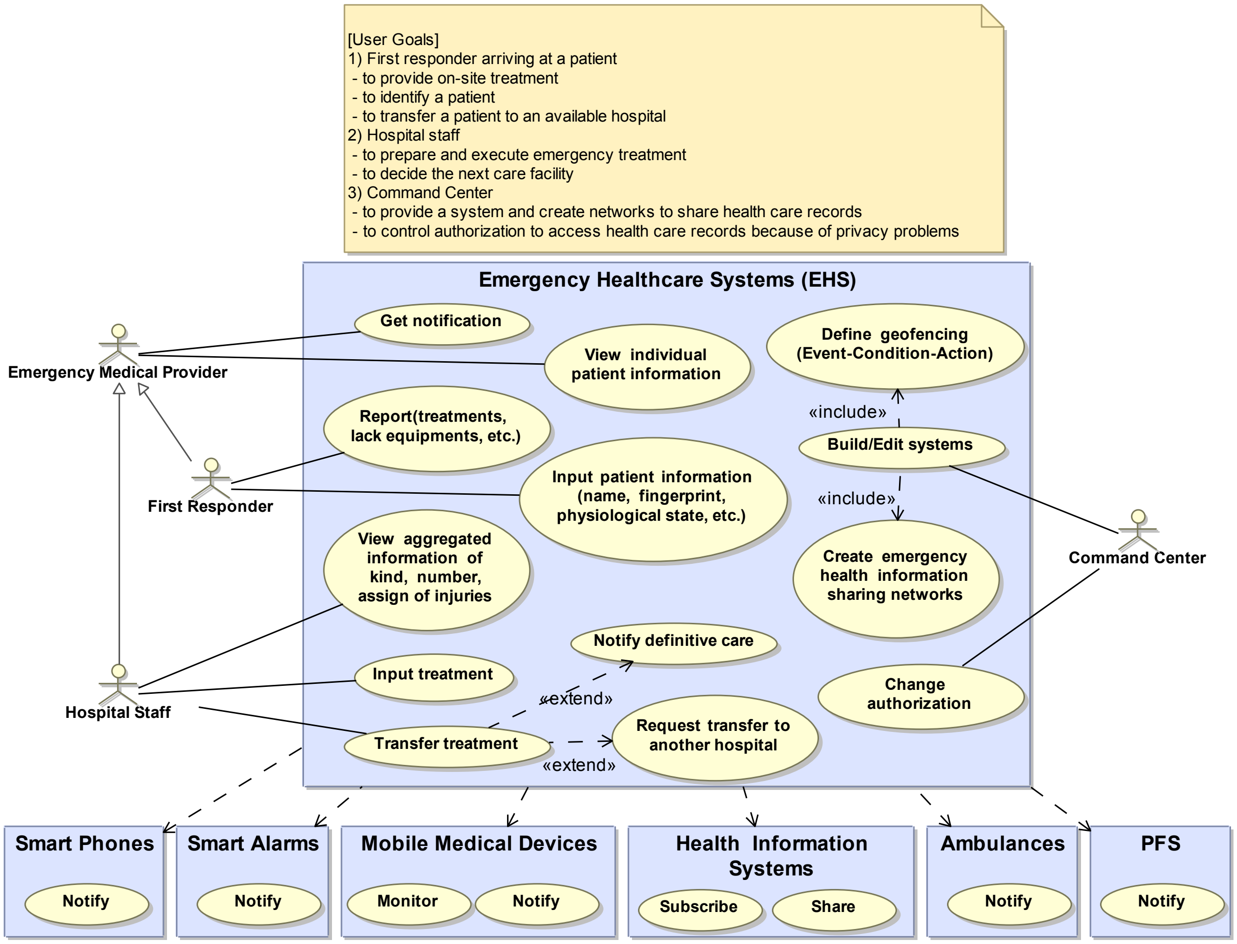

[User Goals]
1) Citizen
- to check on the wellbeing of relatives and friends
- to notify his/her safety via location, text, voice, etc. (especially, Inside Citizen)
2) Telecom
- to track and provide location information of mobile users
3) Command Center
- to provide a systems
- to distribute reliable information

Figure 11 - Use Case Diagram of Person-Find System

The function of the above system is to allow people to locate each other regardless of location. People searching for others send their status and whereabouts through their smart phones and other means to the system. The system collects geo-tagged information from specific citizen messages and from other sensing devices, social networking services, CCTV, and emergency services. The components of the systems interoperate with each in processing by sharing the collected information to take actions such as containing false rumours, locating and informing kith and kin of those who have sent and requested such information. Using personal geo-locations to provide the service uniformly raises privacy issues and makes sharing of this information legally difficult. As of now, this system will only interact with people inside the disaster area. The telecoms can use this information for co-ordinating other component services to provide an effective service to those outside the area.

2.3.5 Emergency Healthcare Systems (EHS)

Figure 12 illustrates a use case diagram for CPCC systems for emergency healthcare. The command center initiates the use case by creating an emergency health network for sharing health records using a highly secure connection. Health record information, when requested by third parties is mediated through the command center. Once the system is configured, the first responder will be notified about the patient condition based on patient provided information such as name, physiological state, etc. via their cell phone. The first responder will also be provided with the healthcare records obtained from the health information system through the emergency health network.


[User Goals]
1) First responder arriving at a patient
- to provide on-site treatment
- to identify a patient
- to transfer a patient to an available hospital
2) Hospital staff
- to prepare and execute emergency treatment
- to decide the next care facility
3) Command Center
- to provide a system and create networks to share health care records
- to control authorization to access health care records because of privacy problems

Emergency Healthcare Systems (EHS)
Emergency Medical Provider
First Responder
Hospital Staff
Command Center
Get notification
View individual patient information
Define geofencing (Event-Condition-Action)
Report(treatments, lack equipments, etc.)
«include»
Build/Edit systems
Input patient information (name, fingerprint, physiological state, etc.)
«include»
View aggregated information of kind, number, assign of injuries
Create emergency health information sharing networks
Notify definitive care
Input treatment
«extend»
Change authorization
Transfer treatment
Request transfer to another hospital
«extend»
Smart Phones
Notify
Smart Alarms
Notify
Mobile Medical Devices
Monitor
Notify
Health Information Systems
Subscribe
Share
Ambulances
Notify
PFS
Notify

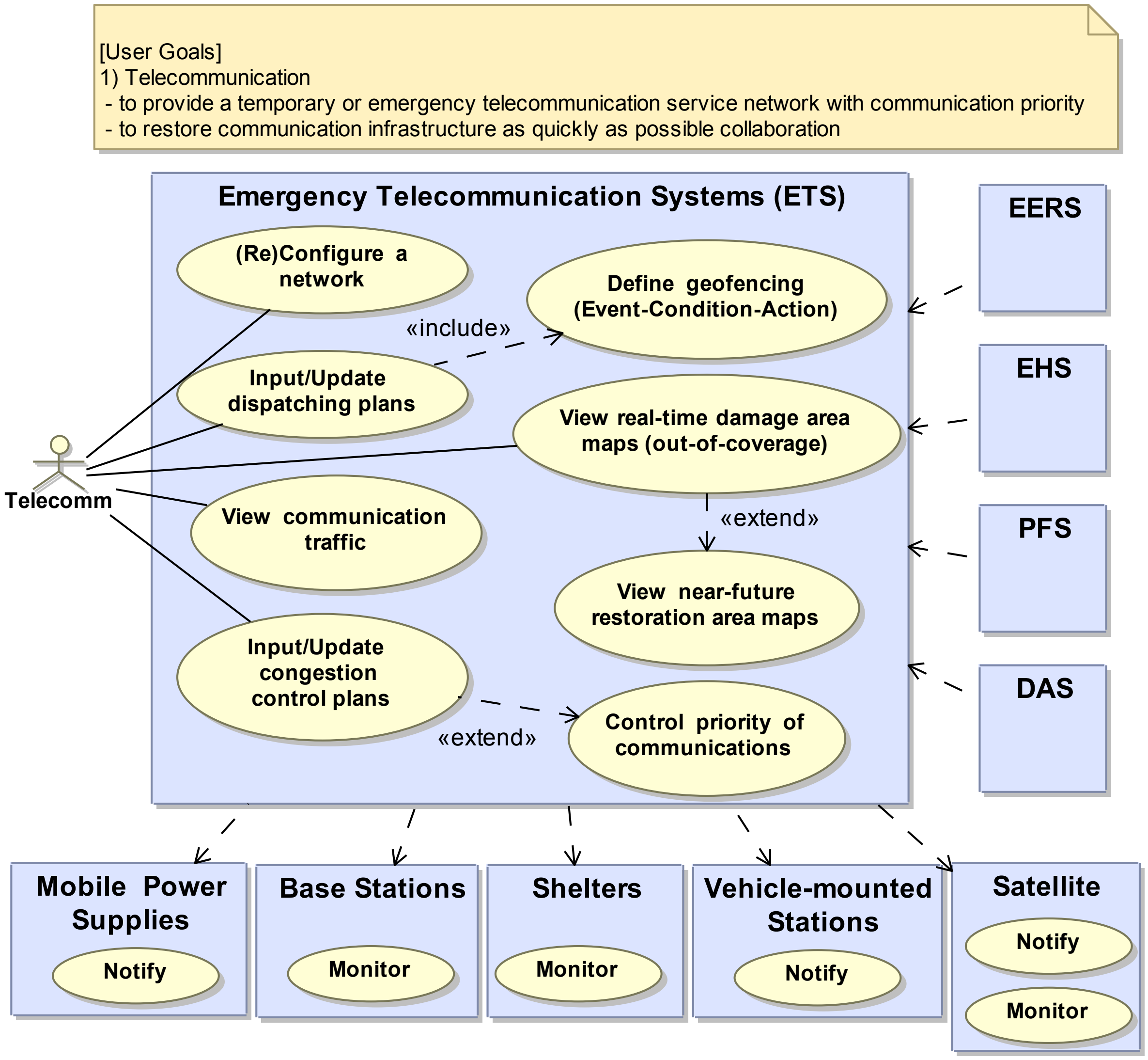
Figure 12 - Use Case Diagram of Emergency Healthcare Systems

Once the first responder makes a triage decision on the patient then all the patient information is collated and sent to the nearest available hospital with the necessary resources. The patient is transferred to the ambulance and is monitored remotely. Additionally, the hospital is readied to provide the next level care as deemed necessary. Through this process, the hospital will be able to keep track of the patients being assigned and inform the system of it capacity in real time to deal with subsequent patients. In case the patient condition cannot be handled by the assigned hospital, the patient will be referred to the next hospital with all the information that first hospital has gathered about the patient. Once the patient has been admitted to a hospital the person finder system will inform the patient location to his/her families or relatives.

2.3.6 Emergency Telecommunication Systems (ETS)

The magnitude of the Japanese earthquake and tsunami highlighted the importance of a robust and reconfigurable telecommunications system in a disaster. Land-based and mobile lines to millions of customers were not available due to extensive damage and congestion. There is a need for an emergency communication system that restores temporary and alternate emergency communication channels. The goal of this system will be to control communication resources dynamically by determining priority, routing and security as shown in Figure 13. In order to achieve this goal, the telecom services will have to monitor the overall coverage and outage to be able to configure alternate, portable, repeating stations and satellite communications links. To aid the telecom companies, the telecommunication system will have to identify system outages and damage assessment of the areas, and to locate mobile stations to configure the temporary network. Further, the need to identify location of non-operational shelters and communicate that information to direct people will be crucial to address the changing conditions on the ground.

The white paper on the next generation network ETS [11] describes the telecommunication requirements for end-to-end priority treatment. For example, “911” calls have a higher priority than private calls and the real-time video about disaster situations needs more bandwidth than voice communication. In Reference [12], International Telecommunications Union categorizes the requirements of public protection and disaster relief based on the range of the radio communication service, the radio operating environment, and the application. Depending on the requirements, different channel bandwidths are assigned. The ETS will monitor network traffic, control communication resources, and assign priorities to each CPCC system.



[User Goals]
1) Telecommunication
- to provide a temporary or emergency telecommunication service network with communication priority
- to restore communication infrastructure as quickly as possible collaboration

Emergency Telecommunication Systems (ETS)
(Re)Configure a network
Define geofencing (Event-Condition-Action)
«include»
Input/Update dispatching plans
View real-time damage area maps (out-of-coverage)
Telecomm
«extend»
View communication traffic
View near-future restoration area maps
Input/Update congestion control plans
«extend»
Control priority of communications
EERS
EHS
PFS
DAS
Mobile Power Supplies
Notify
Base Stations
Monitor
Shelters
Monitor
Vehicle-mounted Stations
Notify
Satellite
Notify
Monitor

Figure 13 - Use Case Diagram of Emergency Telecommunications Systems

3. Requirements and Characteristics of a CPCC System

The functional requirements are driven by the use cases described in section 2 and discussed in detail in section 4 while the non-functional requirements are driven by the characteristics of the system. In addition to the cloud computing characteristics described in section 1.1, the non-functional requirements include (listed in no particular order):

- Configurable / Agile
- Real Time Operation
- Big Data Support
- Reliable
- Decoupled Core Components
- Standardized Interfaces
- Software Defined Networks
- Secure & Private
- Social Interaction
- Synchronized Clocks

These non-functional requirements drive the architecture of the CPCC system and determine its main characteristics. In this section, we explore in more detail these core critical characteristics that represent how the system should perform.

3.1 Configurable / Agile

As described in prior sections, the CPCC system could be used in high-emergency, low-accessibility scenarios. The system should be configured dynamically to suit the current situation. Once configured the system should spawn to life and start servicing its purpose in a timely manner. The core components of the system should have the capacity to be provisioned and interconnected to meet the demand of the system's users easily and rapidly.

3.2 Real Time Operation

One of the critical characteristics of the CPCC system is its capacity to operate in real time. The real-time aspect is necessary to deal with a range of functions at different levels within the system. The cyber-physical nature of the system implies that it has the ability to process a constant stream of data from the sensors. This data stream needs to be coupled efficiently with existing knowledge and manipulated in order to provide accurate and fast actions that influence the physical world. For these actions to be meaningful, the delay between the sensing and the actuating must be kept to a minimum.

3.3 Big Data Support

In order for a CPCC system to execute efficiently and effectively during scenarios such as the ones described in section 2 of this paper, the system must be able to aggregate, store, and manipulate a very large amount of heterogeneous data. The CPCC system relies on sensors that will produce that more or less continuously. These sensors will produce a wide variety of data in different formats and accuracies. This data must be processed, verified and acted on in real time as a requirement of the system. Processing requirements for big data support will vary depending on the amount of data collected, retrieval of prior data and knowledge, and reaction times required by the systems to fulfil the required tasks.

3.4 Reliability

The system must operate robustly in situations where geographical disasters have occurred or are occurring. Some key infrastructures will be damaged and not functional. Many components including sensors, processors, and actuators might stop functioning during critical operations.

3.5 Component Decoupling

The design of the CPCC system should allow composition from independent, decoupled components for flexibility, robustness, and resilience to changing situations. Use of independent and interoperable components allows for ease of composition and the ability for reuse in new and different systems and contexts through their interfaces.

3.6 Standardized Interfaces

In order for the system to rely efficiently on decoupled service components, the interfaces to these components should be based on well defined, interpretable, and unambiguous standards. Further, standardization of interfaces will allow for easy provisioning of various services by a number of CPCC systems envisioned today and in the future. Lack of standardized component interfaces would require customized adapters and could potentially lead to poorer performance and even jeopardize real-time CPCC system operations. The notion of standardized interfaces has to be extended to service catalogues for service registry, service description, and service requirements for easy management and provisioning of services

3.7 Software Defined Networks

The CPCC system is based on a network of sensors providing data based on real world events, data stores of prior information and knowledge, processors that analyse data, actuators that effect change in the physical world, and the networks that connect these devices. In a software-defined network, the system chooses the network parameters according to requirements obtained in real-time. For example, social networks allow humans to share their interpretation of real world events. Based on specific requirements, the system determines what type of data should be gathered and where. The network virtual layer is then reprogrammed to meet these requirements. While a specialized, real-time network can be set up in a small-scale system, the CPCC needs to rely on the Internet for real-time operations of a national-scale cyber-physical system. A new network infrastructure with better support for different levels of real-time might be required.

3.8 Security and Privacy

The CPCC system has to be secure by design in order to ensure that the actions the system takes are the required ones and the information processed is protected. Securing a system based on existing cyber and physical infrastructures raises challenges [13] that need to be addressed.

As cloud computing evolves, there is a need to ensure that security is built in by design. Using cloud-based components for the major modules of the intended system should potentially facilitate the implementation and monitoring of security aspects. As for cloud computing, the privacy concerns will have to be addressed in a localized manner in order to juggle the different stakeholders' regulations efficiently.

3.9 Social Interaction

A big emphasis has to be put to the social nature of the information and the range of interactions the CPCC system has to handle. The system must handle highly unreliable information or actions due to the nature of the information source.

3.10 Clock Synchronization

To compare data points, it is necessary to know when a particular measurement was made. Time stamping of the data is needed for both real-time control and non-real-time data analysis. This issue is critical since the number of systems and their clocks may not be synchronized. Dealing with time stamping and synchronization will be critical to the effective functioning of a CPS, and hence CPCCs.

4. Architecture Framework for CPCC

The functional requirements driven by the use cases described in Section 2 and the non-functional requirements described in Section 3 were used to develop a conceptual CPCC architecture. This architecture is shown using a UML class diagram[4] describing the type of components (represented as classes) existing within the architecture (with their attributes and behaviors) and their relationships with other components within the architectural framework [15]. Since this is a high-level conceptual architecture, not all details are shown. Instead only key classes, attributes, behaviors, and relationships are shown.

4.1 Architecture Framework Class Diagram

The proposed conceptual architecture is shown in Figure 14. As stated earlier, the general goal of a CPCC system is to take data (based on attributes in the physical world) from a variety of sources, convert it to information in the cyber world, process it, understand it, and then transform the information into appropriate actions in the physical world.

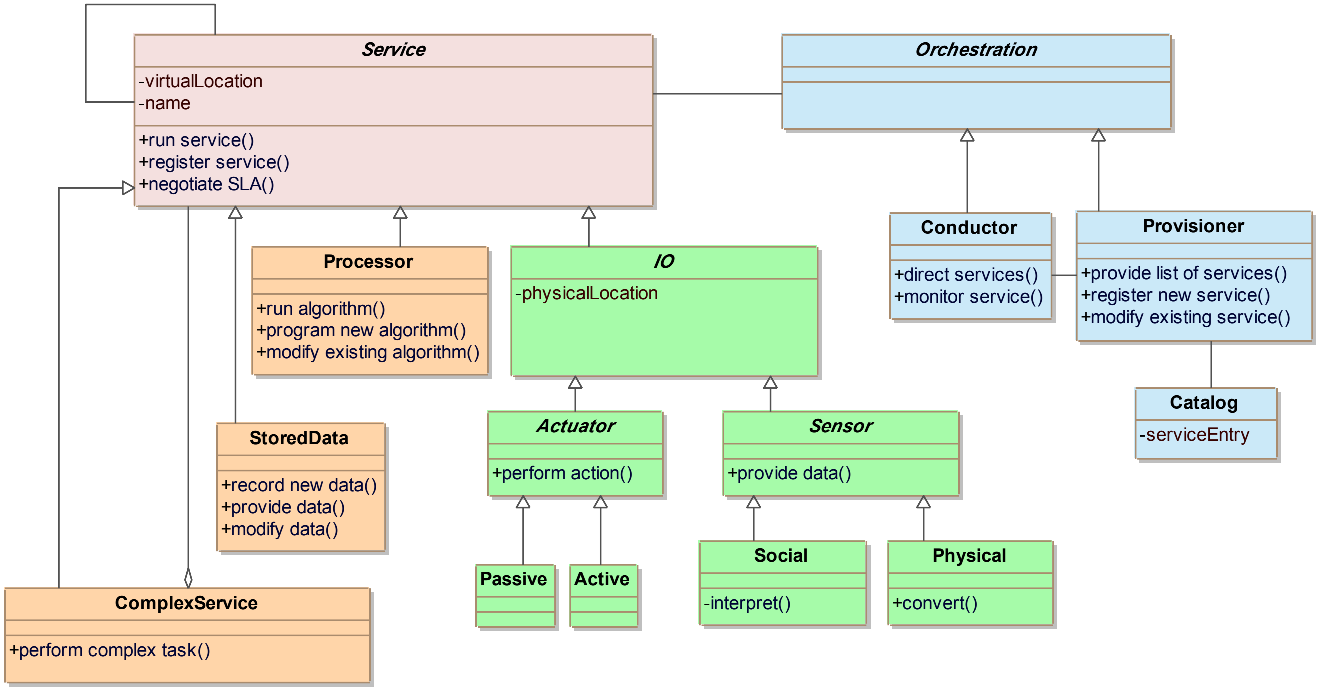


Figure 14 - Conceptual Architecture Framework for CPCC

[4] UML Version 2.0, http://www.omg.org/spec/UML/2.0; last accessed, 19 November 2012

4.2 Description of Framework Components

Three classes of components are shown in Figure 14: 1) components that convert data and information from the physical world into cyber world (sensors – shown in green), 2) components that process data and produce new information (processors – shown in orange), and 3) components that convert the information into actions in the physical world (controllers or actuators – again shown in green). There are also the components that orchestrate these individual components or building blocks into the system to achieve the goal of the system (shown in blue in Figure 14). These orchestration components can interface with non-cloud service-based components, if necessary, to perform a particular task; but these non-cloud components are not described in the architecture. The services may be realized by a single physical entity or a virtual device composed of many discrete devices. Nevertheless, from the perspective of the orchestration system and other services, it is a single service that can be provisioned within the CPCC system. The various classes depicted in Figure 14 are described below (class names are printed using a bold font).

- **IO:** The input/output (**IO**) class is an abstract class for describing the input/output devices. These types of devices provide the interfaces to connect between the physical world and the cyber (virtual) systems. **IO** devices include both sensors that measure some value and actuators that cause or control a physical change to happen based on output from cyber processing.

- **Sensors:** The **Sensor** class monitors the physical world and generates a digital representation of the measurement corresponding to physical events or phenomena. Since the measurement represents an approximation of the actual aspect being measured, uncertainties related to the measurement are required for correct interpretation of data. Additionally, location and/or timing information will be associated with the measurement allowing it to be integrated with other data more easily.

  In the architecture, two types of sensors are defined: 1) traditional types of physical sensors (such as temperature or seismic sensors), and 2) social sensors – using input from humans as sensor data. These sensors have varying levels of intelligence-depending on the type and application of the sensor -- to interpret the data and convert the data into useful information. Physical sensors are mature at this point in time and much knowledge exists on their performance. The main challenge in using these physical sensors is creating standard interfaces and providing the provisioning capabilities so that the sensor may be used as a service. All **Sensor** objects process data to some extent (even a temperature measurement needs to be converted from a voltage or current measurement using a formula or map), but the extent of the processing is left to the designers.

  Social sensors are virtual sensors that provide data gathered from human beings. These social sensors must communicate with the **Provisioner** to exchange information about the sensor availability and capabilities and to provision the sensor for use within a particular system. At runtime, the sensor communicates with the processor services (or directly with the actuator service if there is enough processing built into the sensor and actuator to perform the task) and the conductor that coordinates the system.

Once a measurement has been captured using a sensor or sensors, the data and information need to be processed to create useful knowledge. This is done in the processor by combining the sensor data with data from other sensors and *a priori* knowledge stored in archives or other sources.

- **Processor:** The **Processor** class represents the component that takes sensor input and/or input from other non-sensor data sources (either other **Processor** objects or static data) and processes it according to an algorithm (provided as part of the agreement with the **Provisioner** object for a specific system). The output of a **Processor** object may be fed directly to a controller/actuator, to another **Processor** object (allowing multiple **Processor** objects to be combined to perform a function), or to a stored data service or analysis tool.

  This new knowledge can either be an input into another process, or it can be data that is the input to an **Actuator** object, which results in an action in the physical world – either as information used by the human operators, or as a physical change in the world.

- **Actuator:** The **Actuator** class represents the devices that convert digital information into an action in the physical world. The devices may send a message to a human, or have some other physical effect on the world such as adjusting the temperature of a thermostat or sending an emergency vehicle to a particular location. An **Actuator** object may be as simple as a light switch or triggering a single solenoid, or as complex as a 3D printer or a traffic controller system. Tolerance information may be required along with the specific output value to ensure the system requirements are met when the action is performed.

  A passive **Actuator** object injects information into the physical world (for example through a smartphone application), which may not cause a direct action to occur. The system users can then use the information. An active **Actuator** object directly causes an action to occur in the physical world (a switch or thermostat are good examples of an active actuator).

- **ComplexService:** The **ComplexService** class is a combination of services to create a new type of service that may have multiple sensors, processors, stored data, and actuators. This virtual device is registered in the catalog. In the NIST cloud reference architecture [16], these services are considered cloud-based services that would be within the "Platform as a Service" layer. By providing this functionality a cyber-physical system can be reused without having to recreate the service every time it is used.

- **Provisioner:** The **Provisioner** class defines an object that interacts with the system builder to understand the system requirements and then uses the catalog to determine what sensors, processors, actuators are available and their capabilities. Finally the **Provisioner** object coordinates the agreement for services and scheduling with the services.

When a new system is needed, a system builder (human or virtual) provides a system plan to the **Provisioner** object. The **Provisioner** object then chooses from the available sensors, processors, static data sources, and actuators to build the system that best meets the system design requirements. The proposed implementation of the system is sent back to the system builder for review and approval. Once the implementation has been approved then the **Provisioner** object goes ahead and configures the required services based on the design and associated service agreements and service level agreements.

While the **Provisioner** object usually gets instructions on the design of the system from a system builder (not shown in the diagram), the ability to self-provision should be supported by the architecture. In the case of a self-provisioning system -- one that changes the configuration of the system based on real-time system input -- the **Provisioner** object may make changes based on input from **Processor** objects and/or **sensor** objects.

- **Conductor:** Once the system has been provisioned, an object instantiating the **Conductor** class orchestrates the operation of the system according to the agreements negotiated between the individual services and the **Provisioner** object to achieve the system goals. The **Conductor** object communicates with each of the devices so the components work together at runtime as described by the system design. The conductor orchestrates the system, but does not add and remove (provision/de-provision) components to the system. Like the CPCC services, the conductor may be a single entity or a virtual device composed of many discrete devices. The orchestration components can be provided as cloud services, but it is not required, although it is desirable.

Table 1. Descriptions of CPCC Components

Class	Description	Relationships
Social	A virtual sensor that uses social networking to create information useful for the cyber-physical system that can be provided to the system as a service	Outputs data to **StoredData**, **ComplexService**, **Processor** and **Actuator** Coordinates with **Provisioner** and **Conductor**
Physical	A virtual sensor based on a traditional sensor converting physical measurement(s) into a digital representation	Outputs data to **StoredData**, **ComplexService**, **Processor** and **Actuator** Coordinates with **Provisioner** and **Conductor**
Passive	A virtual device that provides information to the physical world with no direct action	Gets data from **StoredData**, **ComplexService**, **Processor**, **Sensor** Coordinates with **Provisioner** and Conductor

Active	A virtual device that causes a specific action to occur in the physical world	Gets data from **StoredData**, **ComplexService**, **Processor**, **Sensor** Coordinates with **Provisioner** and **Conductor**
Processor	A virtual device that processes data	Gets data from **Processor**, **Sensor**, **ComplexService** Outputs data to **Processor**, **StoredData**, **Actuator**, **ComplexService** Coordinates with **Provisioner** and **Conductor**
StoredData	A data archive that is provided to the system as a service	Provides data to **Processor**, **Sensor**, **Actuator**, or **ComplexService** Coordinates with **Provisioner** and **Conductor**
ComplexService	A service composed of other services to form a new type of virtual device.	Interacts with **Processor**, **Sensor**, **Actuator**, or **ComplexService** Coordinates with **Provisioner** and **Conductor**
Provisioner	Interacts with the system builder to understand the system requirements. Uses **Catalog** or polling to determine what **Sensor**, **Processor**, and **Actuator** services are available and their capabilities Coordinates agreement for services and scheduling based on the system design and available services.	Interacts with **Services**, **Provisioner**, **Catalog**
Conductor	Coordinates services during runtime (based on provisioning) Monitors services	Coordinates with **Service** Interacts with **Provisioner**
Catalog	Maintains listings of service capabilities, availabilities, and costs	Interacts with **Provisioner** Indirectly interacts with **Service** through the **Provisioner**

5. Composing the Pieces for a CPCC Application

This section describes two systems developed at NICT that can be leveraged to build CPCC components that are integral to the realization of a CPCC application. These systems perform different tasks. CPSenS is a cyber-physical-sensing information system that allows for the management of sensors by virtualizing them and enabling their composition on demand. K-L grid, is a system that allows for the composition and integration of different information assets including databases, webpages, public scientific information and so on to be available on demand to the designer and operator of a CPCC system for any given application.

5.1 CPSenS and its integration into a Cyber-Physical Cloud Computing Service

This section describes the Cyber-physical Sensor information systems (CPSenS) and its transformation into a cloud service.

5.1.1 Overview of Cyber-Physical Sensor Information System (CPSenS)

CPSenS is an Information and Communication Technology (ICT) platform for on-demand integration of real-world sensing information with cyber cloud systems. The primary goal of this system is to support a variety of cyber-physical awareness and analysis services. One example of an awareness service involves disaster situation monitoring from massive physical and social sensing data. An example of an analysis service is the huge-scale scientific analytics associated with sophisticated social and natural phenomena. As illustrated in Figure 15, the proposed system plays the central role of collecting and integrating a variety of social and natural sensing information from diverse sources. The massive natural and social sensing data intensively gathered through CPSenS will be analyzed and transformed as useful actionable information to give appropriate feedbacks to people in the physical world.

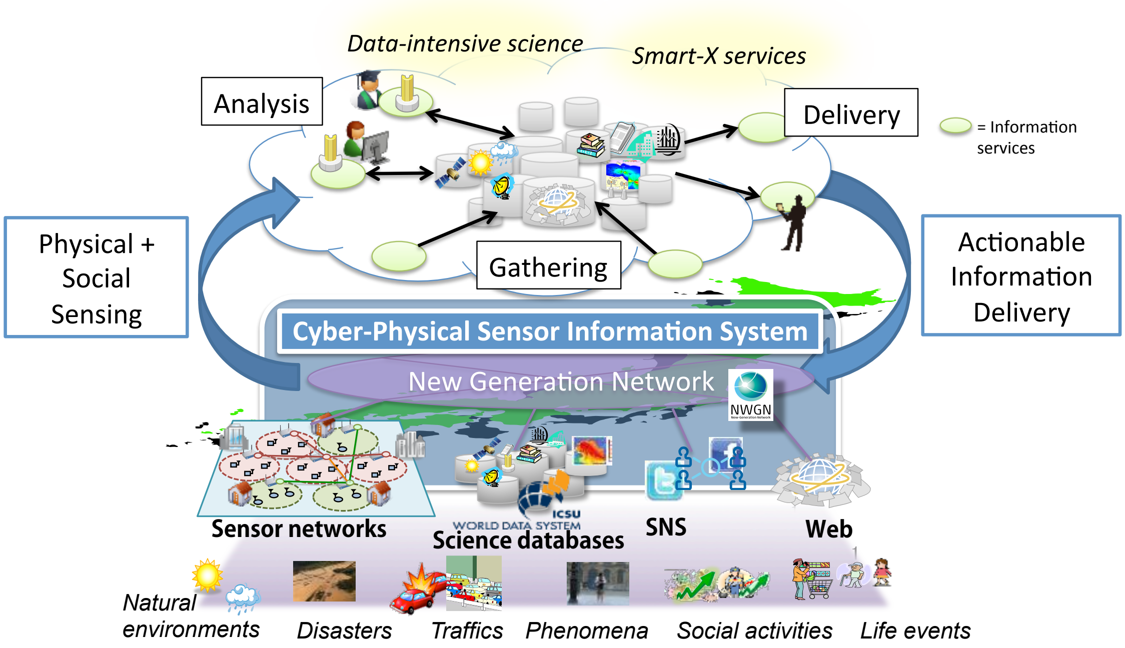


Figure 15 - Cyber-physical Sensor Information System (CPSenS)

The major technical challenge for CPSenS is to create user-defined virtual sensors by combining sensors and analysis services. The goal here is to create simple models of a variety of sensors that can be configured together for collecting a spectrum of integrated sensor data. For this purpose, we design virtual sensor models by encapsulating the very specific and intricate details of the sources. The goal of this design is to make it simple to compose and access heterogeneous social and natural sensing sources. Another important requirement is to construct a composite scalable sensor that can combine heterogeneous sensors on demand. This requirement is needed to ensure comprehensive monitoring of a disaster that will allow for the responders to assess the overall damage from a variety of information sources. This necessitates that the CPSenS system should realize the following requirements:

- *Sensor virtualization by encapsulating sensing data sources as sensor services*. In order to access and manage diverse sensing information easily, CPSenS generates a virtual sensor whose fundamental function is to basically encapsulate sensing data sources. For instance, a weather sensor service could encapsulate either atmospheric sensor devices, a weather agency web site, or weather SNS.

- *Vertical sensor integration by combining heterogeneous sensor services on demand.* In order to monitor a real-world event comprehensively, a more abstract and composite set of sensor services can be created by combining more primitive, heterogeneous, sensor services to integrate various sensing data. For instance, a "Typhoon sensor" can be composed of a weather sensor service, a disaster sensor service, and a social-response sensor service.

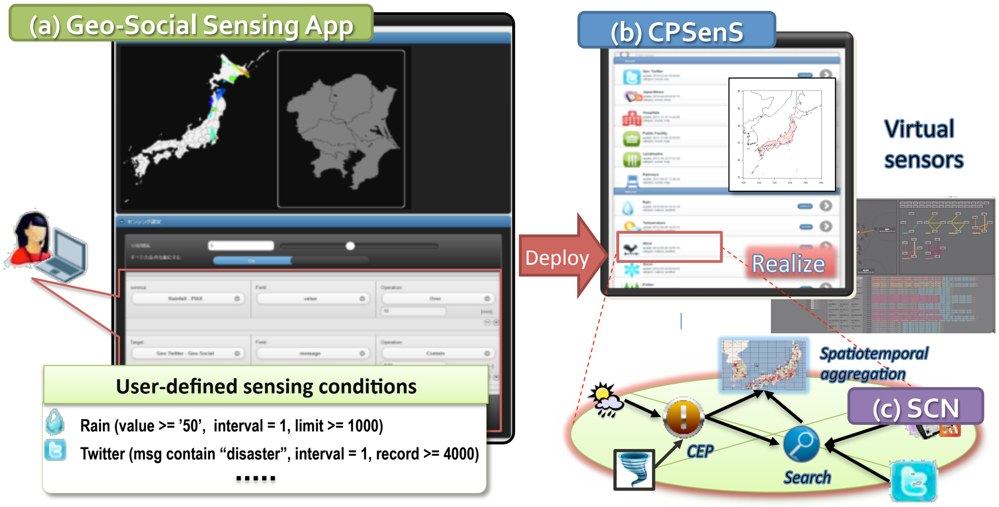
- *Horizontal sensor integration by participatory sensing with in-network data processing.* Sensor data comes from various sources at different levels of quality and reliability. Thus, it is critical to select proper sources and integrate the data from them to meet the expected quality of sensing data to the users. Sometimes, several disaster sensing sources can be faulty and useless, consequently degrading the reliability of the virtual sensors. In order to address such problems, we have to use other sensors as replaceable and complementary sensing sources. For instance, missing data will be complemented with extra sensor services from multiple sensor networks, including mobile sensors and smartphones (human-oriented sensing).

To address above requirements, CPSenS provides the following features:

- *Sensor virtualization by encapsulating external sensors and information sources.* In the CPSenS system illustrated in Figure 16, external physical sensors and a variety of information sources from the Web or other organizations are mapped as virtual sensors that are continuously being monitored and updated by CPSenS as in Figure 16(b). Based on the pool of virtual sensors managed by CPSenS, users are able to create their own large-scale, virtual, inter-connected sensors by choosing and integrating pre-existing, individual, virtual sensors.

- *Virtual/Horizontal sensor integration over a service-controlled networking (SCN).* One of noteworthy characteristics of CPSenS is its underlying realization of a network platform. In this platform, a service-controlled network (SCN), as shown in Figure 16(c), is tightly coupled with CPSenS to coordinate network resources based on the requests from applications. In particular, both primitive, virtual sensors encapsulating external sensors and composite, virtual sensors -- resulting in virtual sensor integration -- are deployed over the SCN platform.

- *Geo-Social (or Socio-Physical) Sensing App for tailoring user-customized virtual sensors.* In addition, to help users design their own virtual sensors CPSenS system provides an interactive user interface called Geo-Social (or Socio-Physical) Sensing App to enable specification of a new sensor (Figure 16a).


(a) Geo-Social Sensing App
(b) CPSenS
Virtual sensors
Deploy
Realize
Spatiotemporal aggregation
(c) SCN
CEP
Search
User-defined sensing conditions

Rain (value >= '50', interval = 1, limit >= 1000)
Twitter (msg contain "disaster", interval = 1, record >= 4000)
……

Figure 16 - Overview of CPSenS System

5.1.2 CPSenS Architecture

The details of the core of the CPSenS shown in Figure 16 are described by the conceptual architecture in Figure 17. CPSenS plays the role of a manager denoted by **Sensor Service** class (in the left upper part in Figure 17), which interacts with two other classes 'Geo-Social Sensing Application' and 'SCN' to construct the CPSenS system. Inside the CPSenS, virtual sensors can exist in two classes: **Atomic Sensor**, and **Complex Sensor**.

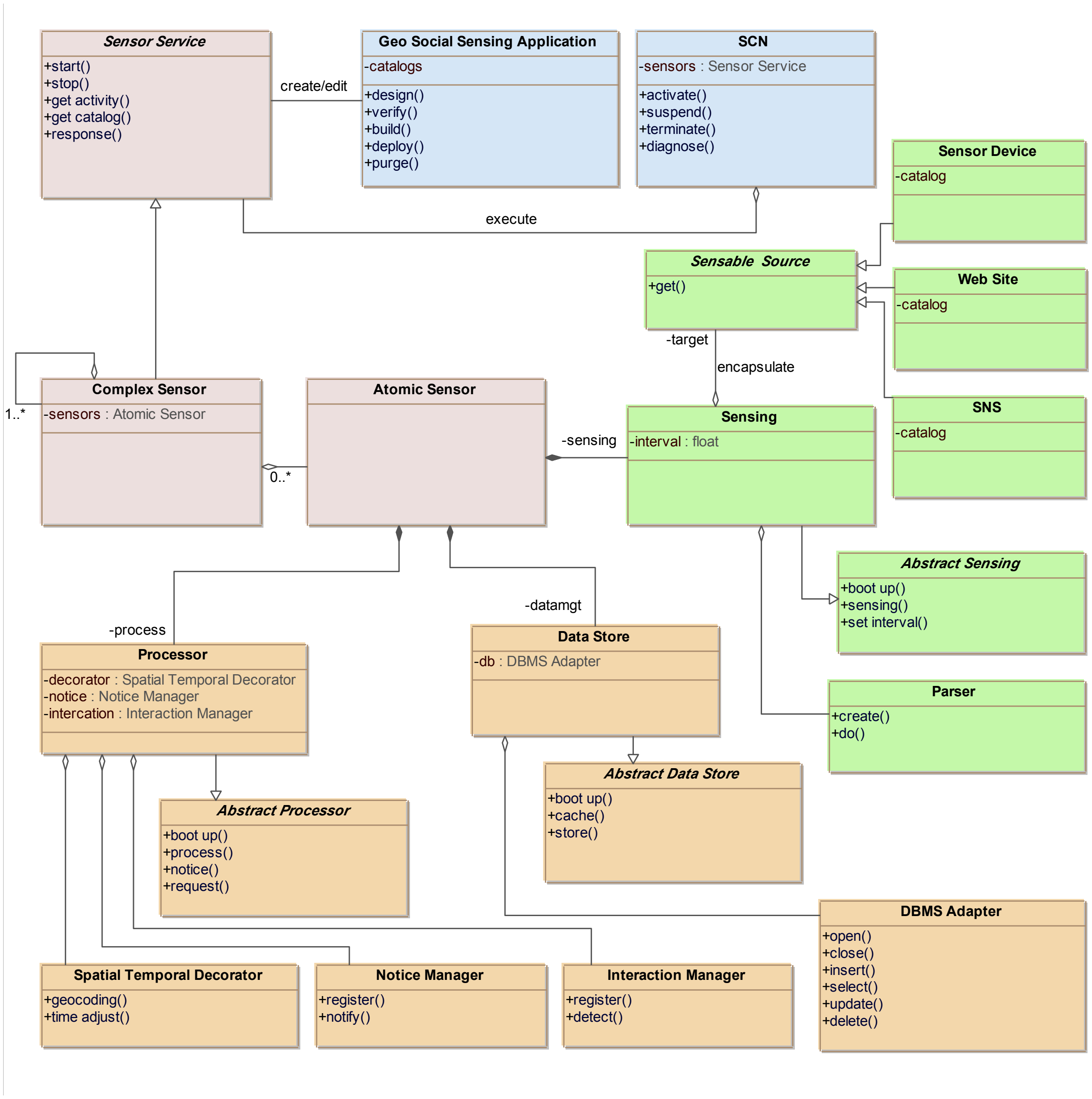


Figure 17 - CPSenS Conceptual Architecture

Various CPSenS classes are described below:

- **Atomic Sensor:** Atomic sensor is the lowest level sensor that is virtualized in CPSenS. The virtualized sensor encapsulates the details of the external physical sensor using three modular information classes: 1) **Sensing**, 2) **Processor**, and 3) **Data Store**.

- **Complex Sensor:** We achieve the vertical and horizontal sensor integration by selecting and integrating atomic sensors. Complex sensors can also be created using other complex sensors.

- **Sensing Service:** Since we are able to create a variety of, and number of, virtual sensors in the CPSenS system, it is critical to control, manage, and coordinate these virtual sensors. Using this class, two types of virtual sensors (**Atomic Sensor** and **Complex Sensor**) are created, deployed, controlled, and monitored.

- **Geo-Social Sensing Application:** In order to help users design virtual sensors, this class supports a user interface with which users can configure a virtual sensor by selecting the sensing sources and specifying the procedure for analyzing the sensed data for use.
- **Sensing:** This is the core class for sensor virtualization meant for interacting with direct sensing sources and fetching sensing data periodically. Customizing source-oriented data, update monitoring, and converting the format of heterogeneous sensing sources are achieved by the following classes: **Sensible Source** and **Parser**. This will allow for easy construction of instances of the **Sensing** class.
- **Sensible Source:** Close to the sensing sources, the role of this class is to monitor updates and fetch data from the lowest level. At the moment, three types of different sensing sources are implemented:
  - **Sensor Device**
  - **Web site**
  - **SNS**
- **Processor:** Once sensing data is imported through the **Sensing** class, it is necessary to pre-process (e.g. geo-coding, and date/time format conversion) the data for a unified data access in the CPSenS system. Further, this class notifies and interacts with the virtual sensor.
- **Data Store:** As an extension of virtual sensors in terms of data management, this class provides archiving and caching services for sensing data for supporting past data search and optimized data access.
- **SCN:** As mentioned before, the CPSenS system is constructed on a service-controlled networking called SCN with which virtual sensors can be distributed over a controlled and reliable network.

5.1.3 Enhanced Usability of Sensors by CPSenS

Figure 18 illustrates the creation of virtual sensors in CPSenS. A virtual sensor is composed of four parts: Sensing, Processing, Data management, and Query Interface. Virtual sensors with the specification of the above parts can model most of the roles of different types of sensors. The conceptual structure of a virtual sensor encapsulates the detailed specification of real-world sensors. Any type of sensor in the physical world and social sensors serving as information sources can be constructed easily in the CPSenS System. Given these individual virtual sensors, composite virtual sensors can be integrated vertically on demand to access different information sources. Each virtual sensor -- by being composed of multiple sensing information sources -- can address the horizontal sensor integration requirement.

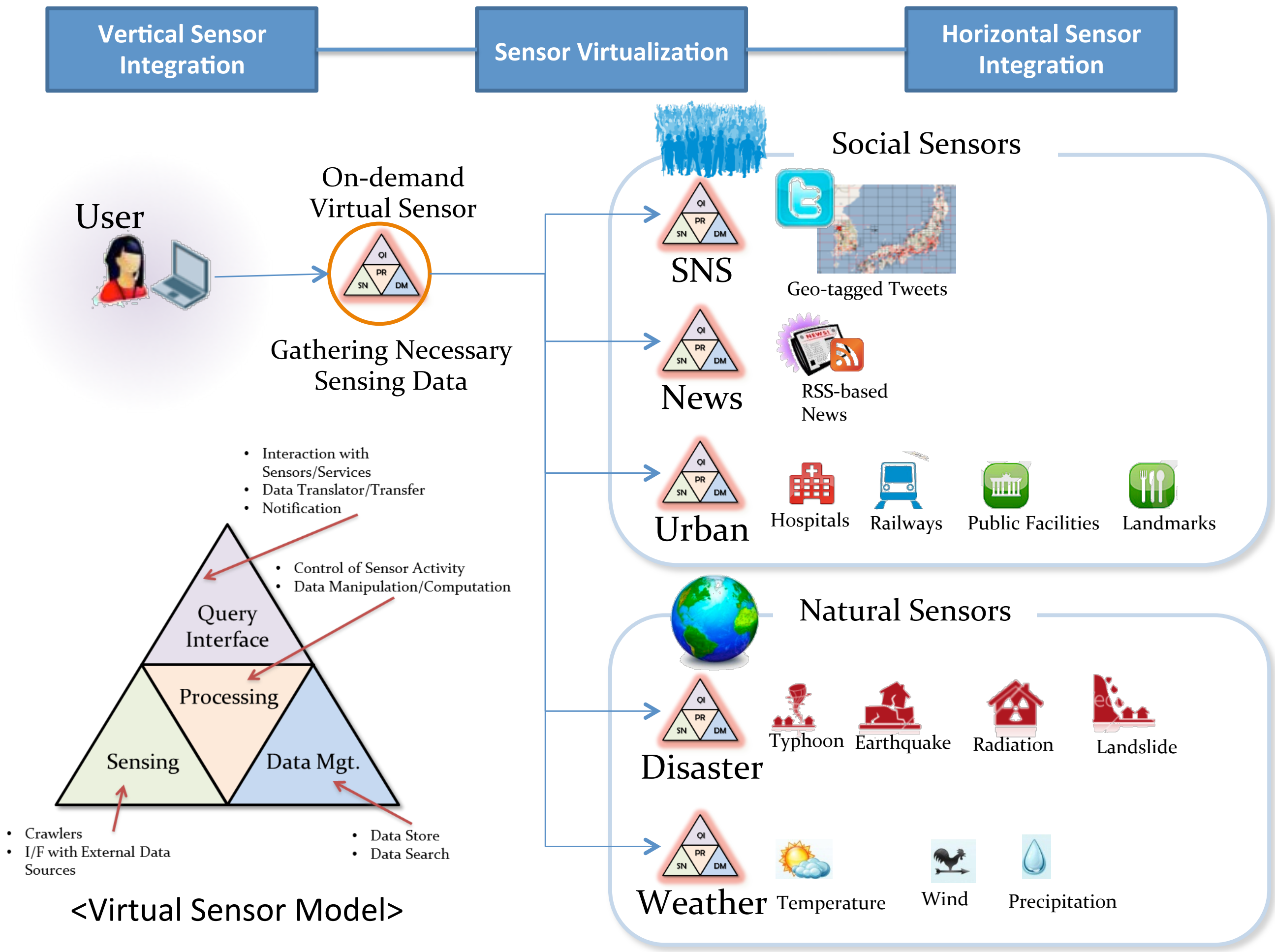


Figure 18 - Creation of Sensor Services in CPSenS

5.1.4 Migration from CPSenS to Cyber-Physical Cloud Services

In the above sections we discussed how CPSenS integrates a variety of real-world, physical, sensing data and cyber information sources to achieve comprehensive and agile situation monitoring and analyses. To make it available for wider use, it will become necessary to migrate from CPSenS to CPC services.

5.1.4.1 Cyber-Physical Sensor Services

While CPSenS was developed to provide sensor services as a standalone project, it can be modified easily to be part of the sensor class required to build the CPCC system. CPSenS is oriented towards enhancing conventional, device sensors by the providing novel characteristics of:

- Sensor virtualization with the supports of extended functions such as aggressive data collecting, pre-processing and even data storing
- Vertical/horizontal sensor integration by enabling users to combine virtual sensors for supporting large-scale cyber-physical monitoring and analysis.

Figure 19 shows the modified CPSenS class diagram. The major change is to move the orchestration components out of the sensing system and use the orchestration class in the CPCC architecture. The other major change is to remove the **Complex Sensor** class as the **ComplexService** class can perform this function for complex sensors.

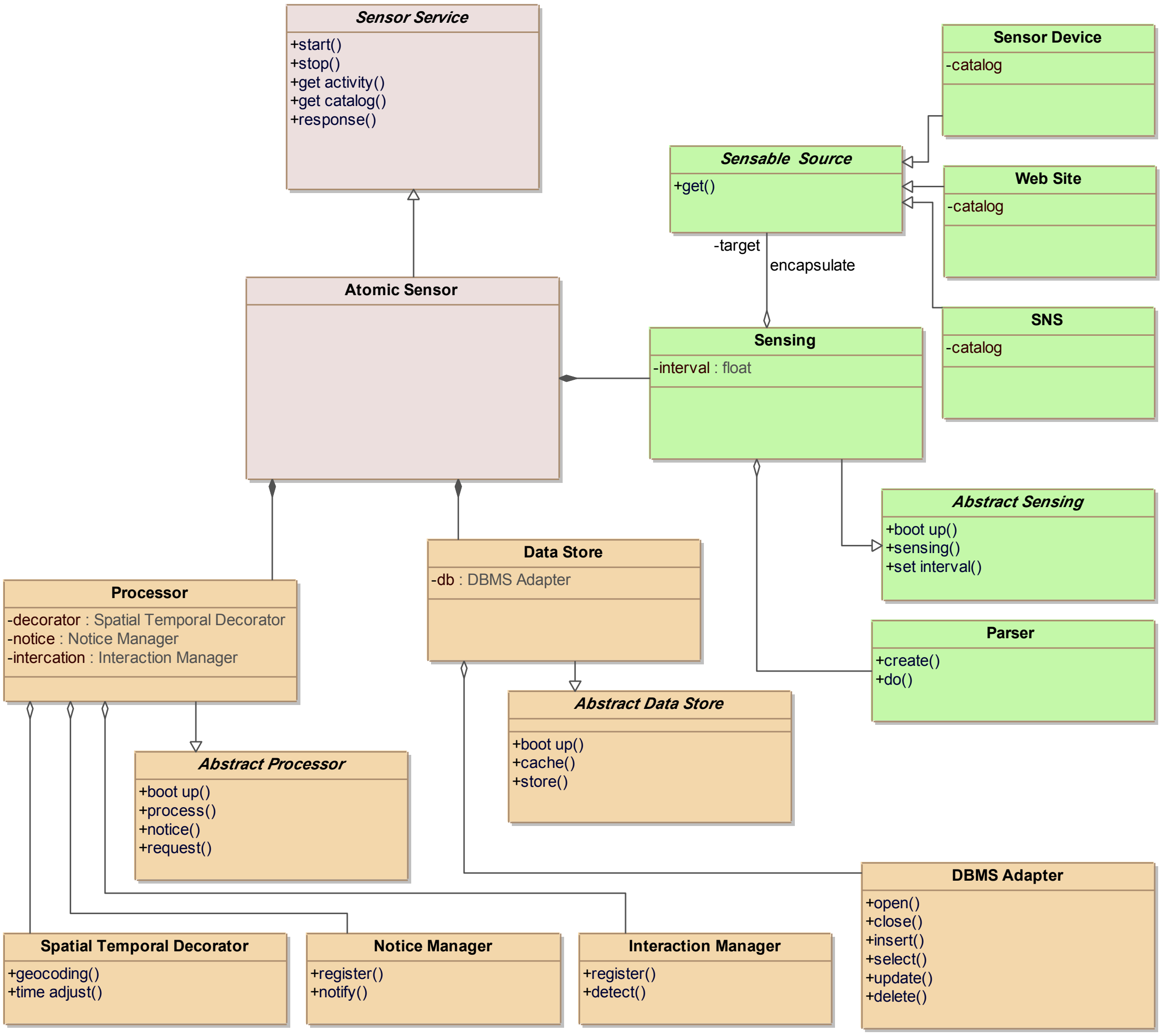


Figure 19 - Modification of CPSenS for CPCC

5.1.5 Adding Actuator Services to CPsenS

To address the need for an actuator service within the CPCC architecture, a CPActS service based on the CPSenS service can be built. The CPActS service component will initiate an action in the physical world. The new service is shown in Figure 19. Many of the components for the CPActS are identical to the ones in CPSenS and can be moved into the parent **IO** class of the CPCC architecture shown in Figure 14. In some cases, sensing and acting may be embedded in the same device; and, in other cases the actuation will be done by actuators that are independently controlled. This separation allows us to configure sensing and actuation as needed.

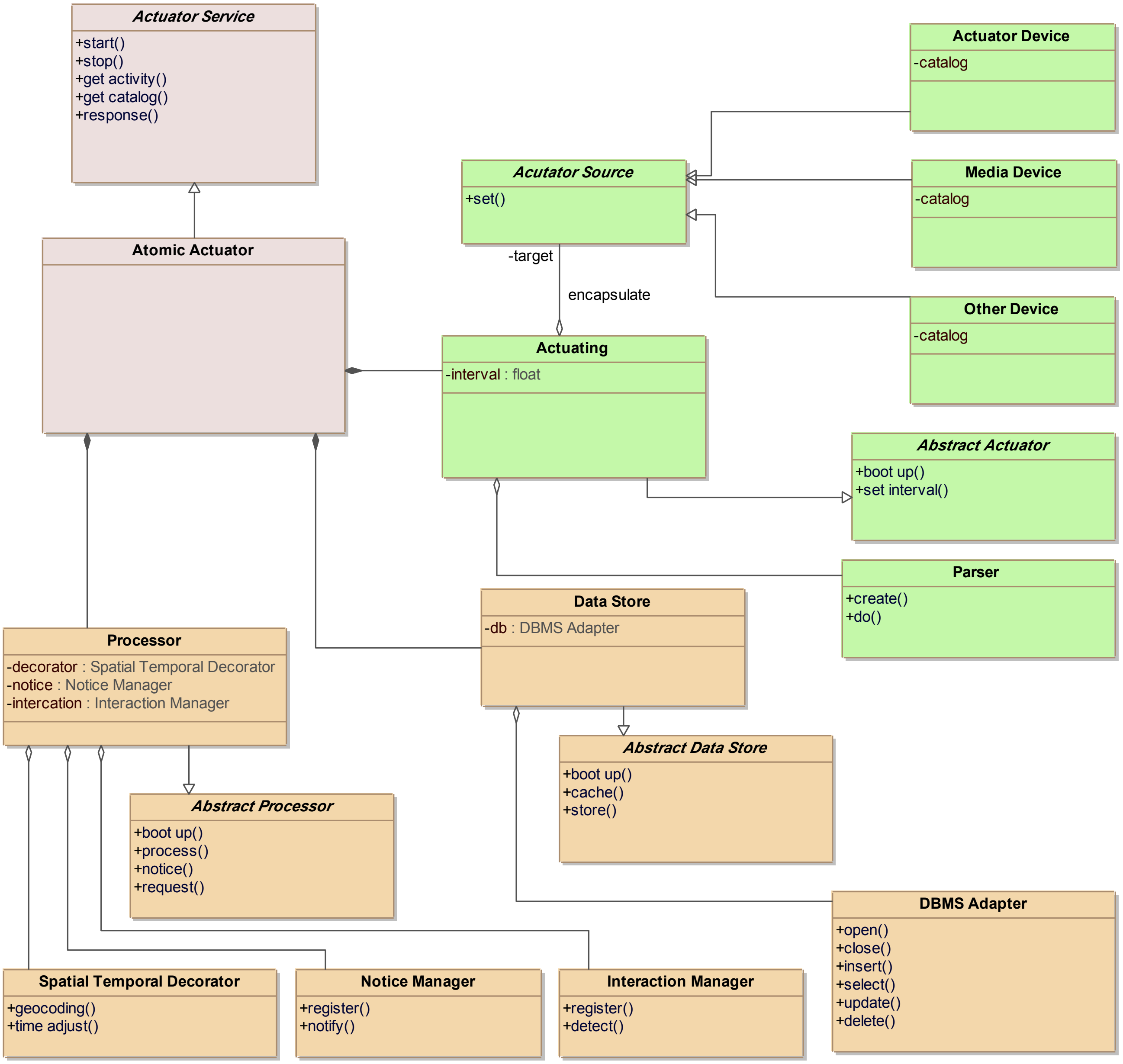


Figure 20 - Blue of CPActS for CPCC Actuator Services

5.2 Knowledge-Language Grid and its Integration into a CPCC Service

In this section, we will describe the K-L Grid, an information services platform and its transformation into a cloud service.

5.2.1 Overview of Knowledge-Language Grid (K-L Grid)

Knowledge Language Grid (hereafter K-L Grid) is an information service development platform for utilizing big data. The purpose of K-L Grid is to enable users to share various types of big data and programs as information assets, and to quickly create information services by combining those information assets. Using this platform, a command center for

Emergency Evacuation and Rescue Systems, discussed in Section 2, can quickly analyze a disaster situation by combining multiple data from different sensors and obtaining information necessary for the K-L Grid user to make a decision. In addition to the real-time data analysis, another application of K-L Grid is systematic verification of Web documents using available scientific data.

In order to implement an agile information service development, we must compose information assets by integrating input and output of data of different software applications with ease. Moreover, large-scale data processing and real-time processing are required for information-asset composition because one of the goals of K-L Grid is to leverage big data. K-L Grid also needs an ecosystem to support a variety of users by its ability to extend the available information assets. In the ecosystem, users can provide their own information assets and other users use the provided data and programs to create additional data sets for use by others. To achieve these requirements, the implementation of K-L Grid consists of three parts:

- Information service development environment
- Information service execution environment
- Information asset management

5.2.2 K-L Grid Architecture

Figure 21 shows a set of use cases of the K-L Grid. There are three stakeholders in K-L Grid: 1) System Administrator, for operating the K-L Grid system, 2) Provider, for generating information assets, and 3) Consumer, for creating information services by using the information assets.

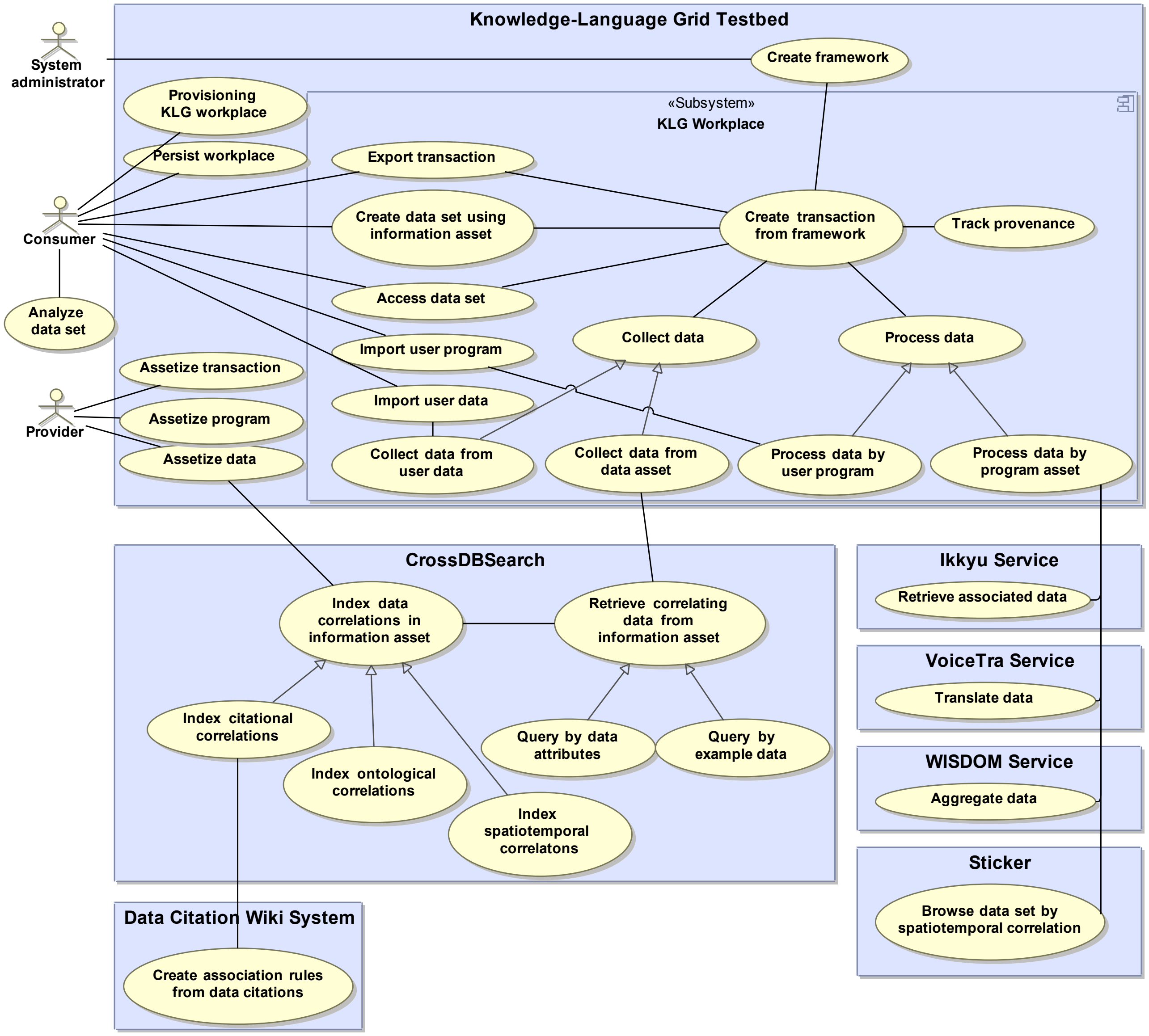

Knowledge-Language Grid Testbed
System administrator
Create framework
Provisioning KLG workplace
«Subsystem»
KLG Workplace
Persist workplace
Export transaction
Consumer
Create data set using information asset
Create transaction from framework
Track provenance
Access data set
Analyze data set
Collect data
Process data
Import user program
Assetize transaction
Import user data
Provider
Assetize program
Assetize data
Collect data from user data
Collect data from data asset
Process data by user program
Process data by program asset
CrossDBSearch
Index data correlations in information asset
Retrieve correlating data from information asset
Index citational correlations
Index ontological correlations
Index spatiotemporal correlatons
Query by data attributes
Query by example data
Ikkyu Service
Retrieve associated data
VoiceTra Service
Translate data
WISDOM Service
Aggregate data
Sticker
Browse data set by spatiotemporal correlation
Data Citation Wiki System
Create association rules from data citations

Figure 21 - Use Case of Knowledge-language Grid

K-L Grid provisions a K-L Grid Workplace as an information-service-development environment for a Consumer. On to the K-L Grid Workplace, consumers can import their own data and programs. They can also create a new information service by combining them with available information assets on K-L Grid, and obtain new data sets from the information service. Using the transaction, Consumer collects data from data assets and user data, and processes them using existing program assets and user programs to create new datasets.

Data is collected from data assets using the Cross Data Base (Cross-DB) search engine. The search engine enables users to query by data attributes and query by examples.

To implement these use cases (see Figure 21), we have designed a conceptual architecture of K-L Grid shown in Figure 22. This architecture consists of two types of components: assets (shown in orange), and components to coordinate assets (shown in blue). The architecture also has three interfaces between these components (shown in purple).

The components to coordinate information assets can be further divided into components to design the system (provisioning) and components to realize the coordination of the system (conducting). The provisioning is implemented by **WorkplaceFactory**, **Workplace**, **AssetRepository**, and **CrossDBSearchEngine**, and the conducting is achieved by Transaction. **WorkplaceFactory**, a manager of the development environment, provisions **Workplace**, an information service development environment, based on **Framework** that defines a coordination pattern of information assets. **Workplace** enables **Consumer** to search information on available information assets using **CrossDBSearchEngine** and **AssetRepository**, and build **Transaction** that describes a specification to coordinate information assets. According to this specification, **Transaction** combines information assets, and provenance information.

In each design and implementation phase of a **Transaction**, it is necessary to access information assets or an information asset profile to search, combine, and record provenance of information assets. To realize this, we introduce three types of interfaces among information assets and components to coordinate the information assets: 1) the Searchable interface is used to search information asset profile, 2) the **Bindable** interface binds information assets to **Transaction**, and 3) the Provenance interface tracks provenance of information assets.

Meanwhile, data and programs coordinated by K-L Grid include not only **DataAsset** and **ProgramAsset**, shared among users as information assets, but also **UserData** and **UserProgram**, imported into **Workplace** for personal use. Currently, **DataAsset** contains **WebArchiveData**, an archive of billions of Web documents; **WDSData**, a science database; **GeoSocialSensingData**, an archive of sensing data; **ProgramAsset** contains **IkkyuService**, a question and answering system; **WISDOMService**, sentiment extraction system; and, **VoiceTraService**, speech recognition/translation/synthesis system. In addition to data and program, **Transaction** can be also shared among users as an information asset by attaching **Searchable** and **Provenancable** interfaces.

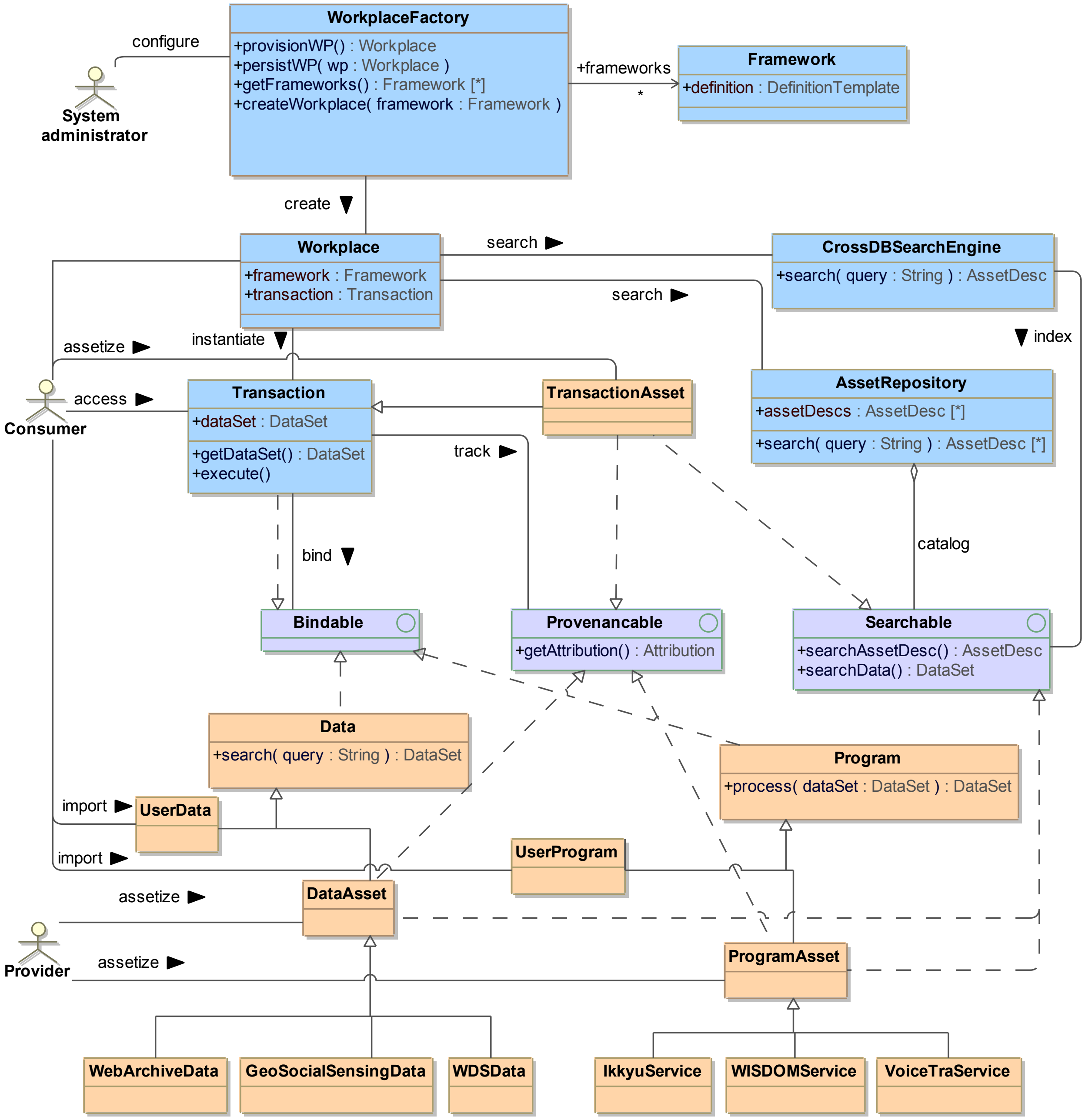


Figure 22 - Conceptual Architecture for Knowledge-Language Grid

5.2.3 Migration of K-L Grid to a CPCC Platform

In this section, we discuss the potential role of K-L Grid as a CPCC platform by orchestrating various cloud services. As shown in Figure 23, we mapped the corresponding components between K-L Grid and CPCC by the same colours. Information assets including transaction assets, data assets, and program assets (shown in orange) in K-L Grid correspond to services such as complex service, stored data, and processors in the CPCC architecture because information assets are also building blocks, same as cloud services in CPCC.

On the other hand, the components to coordinate information assets correspond to **Provisioner**, **Catalog**, and **Conductor** in the CPCC architecture. Specifically, **Provisioner, which** constructs a CPCC system by selecting suitable sensors, processors, data, and actuators is mapped to a combination of three components in K-L Grid: **Workplace**, **WorkplaceFactory**, and **Framework**. This is because the components have a common responsibility of constructing CPCC systems or information services, but not executing them. Next, the **Catalog** for managing available cloud services is mapped to **AssetRepository** and **CrossDBSearchEngine**. To implement information services on the cloud, the **Catalog** stores service profiles and discovers information assets satisfying a composition specification. Lastly, to explicitly map the **Conductor** to a component in the K-L Grid, we divide **Transaction** into a composition specification called **Conductor** and an executor called **TransactionConductor**. **Conductor** and **TransactionConductor** have a common responsibility to run the constructed systems and services using the building blocks.

In this way we can easily map architectures between K-L Grid and CPCC. However, it is necessary to specify the relation between **Orchestration** and **Service** in implementing the conceptual architecture of CPCC. How **Orchestration** utilizes **Service** depends on the goal of the implementation system.

In implementing K-L Grid, we introduced three functionalities: **Searchable** interface to search information assets, **Bindable** interface to combine information assets, and **Provenancable** interface to manage provenance of information assets. The first two functionalities are useful for CPCC systems, in general, to create **Catalog** and enable **Conductor** to execute cloud services. To search and combine various information assets, we need to standardize the metadata that **Searchable** interface returns and data types to define **Bindable** interface. In K-L Grid, users can take on two roles: consumer and provider. Users register their own information assets and other users create new datasets by combining the registered information assets. The created datasets are also registered as information assets. As this cycle is repeated, the dependency between data and programs become too complicated. Therefore, K-L Grid needs to provide a provenance interface so that users can track where the data originated. The provenance information allows users to validate the created data and identify which data should be modified.

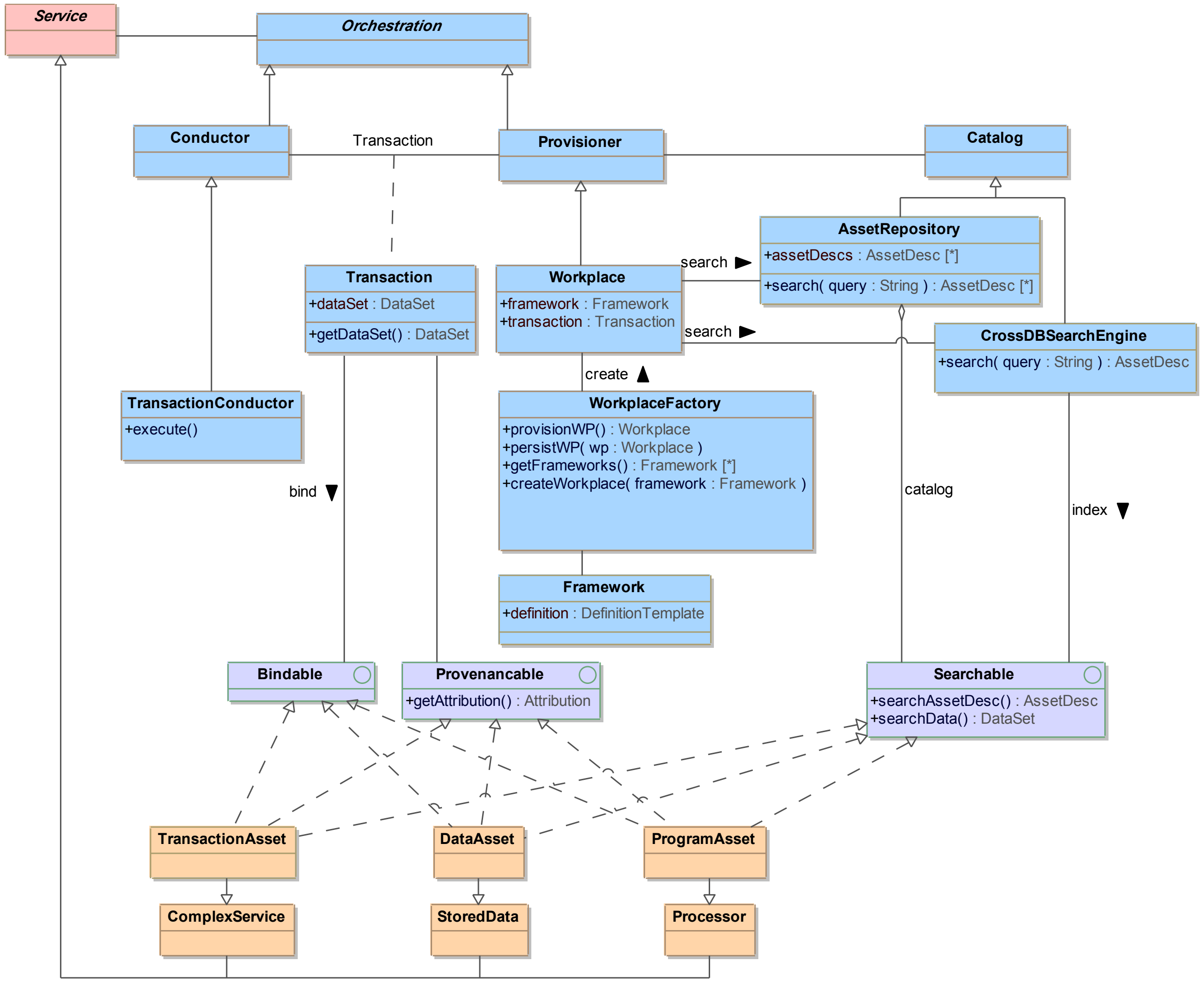


Figure 23 - Cloud Service Orchestration for CPCC

6. Issues in Creating and Managing CPCC

To implement a CPCC architecture such as that described in Section 4 that meets the CPCC requirements described in Section 3 the following issues must be addressed.

6.1 Virtualized Sensors/Actuators

The development of virtualized, service-based sensors and actuators is necessary for the realization of the CPCC services. Presently, there are several efforts to develop systems and related standards to provide sensors as services. However, there is inadequate effort to develop actuators as services.

Some of the requirements for the sensor/actuator services that need to be addressed are listed below:

- Support for heterogeneous sensors
- Ability to parse different types of sensing data
- Caching for improving the performance (within the timing requirements)
- Catalog/directory service for virtual sensors and actuators
- Tools assisting users in the design of large-scale virtual sensors/actuators
- Orchestration for virtual actuators
- Ability to define service capabilities for the catalog
- Defined uncertainties and tolerances
    - Sensing devices need the ability to report uncertainties in a standardized and understandable fashion
    - Actuators need to know what tolerances are associated with a particular action (for example: timing and magnitude)
    - Need to assign uncertainties/tolerances that can be compared across devices
- Social sensors
    - Clear interpretation of social data
    - Methodology for assigning uncertainties for social data
    - Semantics and data formats for virtualized sensors

6.2 Interconnectivity

The ability for the services and orchestration system to operate across a broad network in real-time is necessary for fulfilling the goals of the CPCC system. The eco-system must support the connection of diverse components together into a single virtual system. The heterogeneous nature of the resulting CPCC system requires a high level of service interoperability and data portability between components.

Some of the areas that need to be researched include:

- Network-level
    - Service-controlled networking/software defined networking
    - Real-time operation with different levels of support

- Developing interfaces between services and between services and orchestration that:
    - Provide Representational State Transfer (REST)-based API of data and application
    - Standardize data types to define bindable interfaces
    - Describe composition specification based on dependency between input and output of data and programs
    - Allow composition of sensor services/information asset

6.3 Data Integration (Semantics and Formats)

Not only do the services need to interoperate with each other, but also data must be portable between services and systems. The large number of heterogeneous cyber-physical cloud services within the systems will increase the amount and variety of data that is gathered, combined, and interpreted for targeting actions and creating new knowledge. It is not only a problem of data volume but also of data form and meaning. There are many different types of data with different units, uncertainties, structures, and semantics in a cyber-physical cloud system.

The following data management issues must be considered:

- Improvement of quality (accuracy) of data in real time -- sampling and filtering
- Unified data representation and processing models to accommodate heterogeneous or new types of data
- Intelligent data interpretation and semantic interoperability
- Inter-situation analysis and prediction
- Knowledge creation and reasoning
- Short-term and long-term storage

6.4 Resource Provisioning

To achieve the successful deployment of the CPCC systems, the individual services need to be coordinated and organized into a virtual system. The following provisioning aspects are needed:

- CPCC Service Level Agreement (SLA) taxonomy to provide a common language describing the CPCC system
- A SLA negotiation framework to allow the **Provisioner** to negotiate with the services what will be provided and at what cost
- A CPCC service catalog framework providing
    - Standardized description of services including functional and non-functional (performance, uncertainty, security) abilities
    - Compatibility with existing Service Oriented Architecture (SOA) description standards
    - Service detection

6.5 Security & Privacy

In any complex, heterogeneous, multi-stakeholder system there is an important need to secure the system and protect the privacy of the users and the users' data. While much of the existing security/privacy knowledge, systems, and standards will be applicable to any CPCC System, a full analysis of security/privacy requirements needs to be completed and guidelines developed for implementation. With regards to data and resource sharing, privacy and security policies and issues will impact the management of access to provenance information to record ownership and process history of data across and within services.

6.6 Performance

CPCC will be used for a variety of applications. Each application will have specific performance requirements that need to be met to achieve the application goals. Timing (including time synchronization) and accuracy are two priority characteristics that need to be well characterized. Timing performance (i.e., response time, delay, and jitter) is required to meet the specific real-time needs of a control system. Time clock synchronization is required so that time-stamped sensor data from different sensors can be combined into a meaningful dataset. Time synchronization can also be used for advanced control systems that use real-time clocks instead of timers or sequences to trigger an action. Likewise, sensor-measurement accuracy (uncertainty) should be well characterized.

Since not all applications have the same performance requirements, a **Provisioner** needs to know what each service can provide before the service is used in a particular application. Therefore, the services must be aware of their service performance characteristics and have the ability to communicate these to the provisioning system (either at the request of the **Provisioner** or when the service is added or updated within the service **Catalog**). The characteristics should be defined in a manner that is unambiguous, consistent, and measurable. Challenges include designing the CPCC architecture and protocols that support the reliability requirements along with metrics and methodologies for testing.

6.7 Reliability

As with performance, each application will have specific reliability requirements that must be met. Aspects of reliability that are of concern include: service uptime, redundancy, recovery, and outage reporting. As with performance requirements, not only does a service need the ability to meet the reliability requirements, it needs the ability to report this information to the orchestration component. Challenges include designing the CPCC architecture and protocols that support the reliability requirements along with metrics and methodologies for testing.

6.8 Measurements

It is necessary to be able to measure and test the system for the various characteristics mentioned above (such as performance, reliability, security, etc.). Measurements, metrics, and test methods need to be available to test and characterize various aspects of the systems. In cases where it is possible for measurements to be made, an understanding of the uncertainties associated with the measurements also needs to be developed. Since a measurement is only an approximation, without an understanding of the uncertainty associated with a measurement, the measurement itself has little meaning. Each individual service has different parameters that must be measured. A clear, reproducible, methodology is needed.

6.9 Standards for CPCC

To provide the flexibility necessary to construct these CPCC systems, standards are needed for semantic understanding, data formats, and interfaces between system components, service descriptions, and security/privacy models. Without standards the CPCC services cannot be combined into a system. Without standards a **Provisioner** cannot know whether a given service will meet the system needs. Without standards it will not be clear whether a service has fulfilled it service obligations. While badly designed standards or standards that provide too little or too much flexibility will hinder the ability to connect devices and build the CPCC systems, well-designed standards will enhance the ability to create these systems. Once the initial design requirements for the CPCC system are complete, a set of standards requirements can be developed.

Although CPCC systems have not yet been realized, there are many existing standards supporting sensor measurements, cyber-physical systems, service oriented architectures, and cloud computing. Once the initial evaluation of standards requirements is complete, a comparison against existing standards should be done. Suitable standards should be reused and standards gaps developed based on this comparison.

The quality of the standards should be considered when choosing and developing standards for the CPCC system. Due to the extremely complex and heterogeneous nature of CPCC systems it is imperative to use standards that meet a clearly defined set of goals, are clear and unambiguous, and can be implemented with ease. Important characteristics of standards for CPCC include: semantically unambiguous, machine readable, query-able, and testable.

While it may be ideal to have only one standard within each standards space, it is likely that there will be overlapping standards within a given space. Therefore, mapping and translation approaches that can accurately convert from one standard to another will be critical.

Specific CPCC standards priority actions include:

- Finalizing a set of functional and non-functional goals and high level use cases for the CPCC system
- Specifying the design of the CPCC system
- Specifying the description of each service and its capabilities; this area can use previous work for cloud computing and for sensor services
- Specifying the interfaces between services and between services and the orchestration components; this area can leverage previous work for SOA, cloud computing and other service based architectures
- Specifying the semantics for CPCC systems; this area can leverage existing semantic standards
- Specifying service Level Agreements for CPCC systems; this area can leverage existing cloud computing SLA and add support for the specific sensor and actuator requirements such as uncertainty and tolerance
- Specifying the semantics, data format and uncertainties for social data

7. Research Areas for CPCC

To realize the CPCC system architecture the aforementioned issues need to be resolved. These issues can be mapped into six major research areas:

1. *CPCC semantics and ontology:* understanding the concepts involved and the different languages used by the various stakeholders
2. *CPCC architecture:* understanding the basic components of the system and investigating different possible implementations
3. *Knowledge generation from social sensors:* interpreting human language data from single and multiple sources
4. *Service discovery, provisioning and conducting:* understanding how the orchestration component functions
5. *Interoperability, portability and related standards:* developing needed standards
6. *System security and privacy:* developing both technology and standards

Specific next steps for the joint NICT/NIST program include:

- Develop business goals and business use cases for the CPCC system along with a proposed CPCC system including CPCC principles.
- Create an architecture for the implementation of cyber-physical cloud systems. The architecture will provide the basic building blocks that can be implemented by different stakeholders to implement specific systems as desired. This includes:
  - Analyze and decompose the goals of such a meta-system
  - Develop subsystem use cases
  - Generate a set of requirements based on the use cases and goals.
- Review state of the art for distributed service-based CPS, including existing, systems, expert papers, and standards to determine what can be leveraged.
- Identify and solve implementation gaps based on the set of requirements compared to the existing state of the art.
- Ensure standards exist for CPCC systems and infrastructure. Work with Standards Development Organizations to modify or develop the required standards for interoperability, portability, security standards and metrology in CPCC systems. Where multiple standards overlap, develop harmonization methodologies to allow multiple standards in the same or overlapping space to interoperate.

The above research areas are restricted to the cyber-physical cloud system infrastructure and potential application scenarios. The application scenarios have been simplified to emphasize the relationship to the cloud system, The scenarios such as Smart Disaster Management interface with a number of human organizations such as the police, fire department, homeland security, emergency management agency and other including the general pubic that provides information through social media. Integration of the cyber-physical cloud system with organizational and social informational flows for decision making at different levels of the organizations will take this work into the next step where SNSS function as mutually dependent parts of the larger smart society.

References

1. Weiser, M., "The Computer for the 21st Century," *Scientific American*, Vol. 265, No. 9, pp. 66-75, 1991.

2. Lee, E.A., "Cyber-Physical Systems - Are Computing Foundations Adequate?," *Position Paper for NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*, Austin, Texas, October 16-17, 2006.

3. "Cyber-Physical Systems: Draft Situation Analysis of Current Trends, Technologies, and Challenges," *NIST Foundations for Innovation for Cyber-Physical Systems Workshop*, June 18, 2012.

4. Conti, M., Das, S.K., Bisdikian, C. et al., "Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber-physical convergence," *Pervasive and Mobile Computing*, Vol. 8, No. 1, pp. 2-21, 2012.

5. Murakami, K.J., "Enabling Cyber-Physical-Social Systems (CPSS) for a Better Future." JEC-ECC 2012: 2012 *Japan-Egypt Conference on Electronics, Communications and Computers*. Available at http://soc.ait.kyushuu.ac.jp/presentation/murakami_20120307.pdf. Last accessed, November 5, 2012

6. Cook, D., Das, S.K., *Smart Environments: Technology, Protocols, and Applications*, John Wiley, New York, 2004.

7. (n.d.) "Winning the Future with Science and Technology for 21st Century Smart Systems," Available at: http://events.energetics.com/NISTCPSWorkshop/pdfs/CPS_OSTP_Response_WhitePaper.pdf. Last accessed, November 2, 2012.

8. Abowd, G.D., et al., "Towards a Better Understanding of Context and Context-Awareness," In *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pp.304-307, 1999.

9. Black, J. P., et al., "Pervasive Computing in Health Care: Smart Spaces and Enterprise Information Systems," In *Proceedings of the Second International Conference on Mobile Systems, Applications and Services (MobiSYS2004)*, June 2004, Boston, MA.

10. Mori, N., T. Takahashi, T. Yasuda, and H. Yanagisawa, "Survey of 2011 Tohoku earthquake tsunami inundation and run-up," *Geophys. Res. Lett.*, 38, L00G14, 2011.

11. "MultiService Forum: Multivendor testing in global LTE & IMS Networks," *MSF VoLTE Interoperability Event 2011*. Available at http://www.msforum.org/interoperability/MSF_VoLTE%20_2011_WhitePaper.pdf. Last accessed, November 2, 2012.

12. ITU report No. M.2033, *Radio communication objectives and requirements for public protection and disaster relief*, 2003. Available at http://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2033-2003-PDF-E.pdf. Last accessed on November 2, 2012.

13. Cardenas, A. et al., "Challenges for Securing Cyber Physical Systems," *Workshop on Future Directions in Cyber-physical Systems Security*, DHS, July 18, 2009.

14. Craciunas, S.S., et al., "Information-acquisition-as-a-service for cyber-physical cloud computing," in *Proc. of the 2nd USENIX conference on Hot Topics in Cloud Computing (HotCloud)*, 2010.

15. Unified Modeling Language version 2.0, http://www.omg.org/spec/UML/2.0/. Last accessed, November 2, 2012.

16. Fang, L. et al. NIST Cloud Computing Reference Architecture (Draft) *NIST Special Publication 500-292*, pp: 1-35, 2011.

Appendix A: Glossary

Actuator A device causing a change to occur in the physical world based on data output from a processor

Conductor An orchestration component that coordinates a service based system during run-time

Cloud Computing System (also referred to as a Cloud System) A system, having the following five characteristics; on demand self-service, broadband network access, resource pooling, rapid elasticity and measured service

Cloud Infrastructure The collection of hardware and software that enables the five essential characteristics of cloud computing [NIST SP 800-145]

Cloud Service A cloud computing system provided as a service.

Cyber-physical Cloud Computing Service A cyber-physical system delivered as a cloud service

Cyber-physical Cloud Computing System A cyber-physical system composed of services and exhibiting the five characteristics of cloud computing

Cyber-physical System A system interlinking the cyber and physical worlds through sensors and actuators

Functional Requirements Requirements to achieve the functions a system is designed to perform

Non-functional Requirements Requirements to achieve aspects of the system not directly related to the system functions

Orchestration The automated arrangement, coordination, and management of components within a complex system

Processor A component that takes data inputs, applies some logic to them and outputs the results

Provisioner An orchestration component that prepares, schedules and configures components within a complex system

Sensor A device that measures a physical quantity and converts it into a digital representation

Service A set of functionalities grouped together that interface with other services as well as the orchestration component

Appendix B: Acronyms

CCTV	Closed-circuit television
CPActS	Cyber-physical Actuator System
CPCC	Cyber-physical Cloud Computing
CPS	Cyber-physical System(s)*
CPSenS	Cyber-physical Sensor systems
DAS	Disaster Assessment Systems
DRS	Disaster Response System
EERS	Emergency Evacuation and Rescue Systems
EHS	Emergency Healthcare Systems
ETS	Emergency Telecommunication Systems
ICT	Information and Communication Technology
IO	Input/Output
IoT	Internet of Things
K-L Grid	Knowledge-Language Grid
NIST	National Institute of Standards and Technology
PFS	Person Find Systems
REST	Representational State Transfer
SCN	Software Controlled Network
SLA	Service Level Agreement
SOA	Service Oriented Architecture
SNS	Social Network Service
SNSS	Smart Networked System and Society(s)*
UML	Unified Modeling language

*These acronyms are used to represent both the singular and plural forms