

NISTIR 7198

Shop Data Model and Interface Specification

Charles McLean
Y. Tina Lee
Guodong Shao
Frank Riddick

NIST

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

NISTIR 7198

Shop Data Model and Interface Specification

Charles McLean

Y. Tina Lee

Guodong Shao

Frank Riddick

Second Author

*Manufacturing Systems Integration Division
Manufacturing Engineering Laboratory*

January 2005



U.S. DEPARTMENT OF COMMERCE

Donald L. Evans, Secretary

TECHNOLOGY ADMINISTRATION

Phillip J. Bond, Under Secretary of Commerce for Technology

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

Hratch G. Semerjian, Acting Director

PREFACE

The High Performance Computing and Communication (HPCC) program was formally established by the High Performance Computing Act of 1991 (Public Law 102-194). The goal of this program is to accelerate the development of future generations of high performance computers and networks and the use of these resources in the government and throughout the U.S. economy. The National Institute of Standards and Technology's (NIST) System Integration of Manufacturing Applications (SIMA) program is the agency's coordinating focus for its HPCC activities, see [NIST 2003]. SIMA is addressing the information interface needs of the U.S. manufacturing community. The SIMA program focuses on:

- Defining, testing, and promoting standards for interoperability solutions, and
- Facilitating remote access to scientific and engineering data

This document presents an information model that provides neutral data interfaces for integrating machine shop software applications with simulation. The model is presented using the Unified Modeling Language (UML) and the eXtensible Markup Language (XML).

This draft information model is being transferred to the Core Manufacturing Simulation Data (CMSD) Product Development Group (PDG) of the Simulation Interoperability Standards Organization (SISO) [SISO 2005]. The model will be the strawman of the first product of the CMSD PDG. SISO focuses on facilitating simulation interoperability and component reuse across the Department of Defense, other government, and non-government applications. SISO originated over ten years ago with a small conference held 1989, called, "Interactive Networked Simulation for Training." One of the outcomes from the workshop of 60 participants was a concern that there was activity occurring in networked simulation, but that it was occurring in isolation. The group believed that if there were a means to exchange information between companies and groups that the technology would advance more rapidly. The group also believed that once the technology begins to stabilize then there would also be a need for standardization. The technology and the consensus of the community would be captured in the standards as networking or simulation technology matured.

The conferences soon developed into the Distributed Interactive Simulation (DIS) Workshops and in late 1996, in light of the development of the High Level Architecture (HLA), the DIS organization transformed itself into a more functional organization called SISO. Subsequently, on 19 November 2003, the IEEE Computer Society Standards Activities Board voted unanimously to grant the SISO Standards Activities Committee (SAC) status as a recognized IEEE Sponsor Committee. The ultimate result is a single body for creating and promoting interoperability standards in the modeling and simulation community.

Work described in this paper was sponsored by the NIST Systems Integration for Manufacturing Applications (SIMA) and the Software Engineering Institute's (SEI) Technology Insertion, Demonstration, and Evaluation (TIDE) Program. SIMA supports NIST projects applying information technologies and standards-based approaches to manufacturing software integration problems. TIDE is sponsored by the Department of Defense and Software Engineering Institute. No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. The work described was funded by the United States Government and is not subject to copyright.

1	INTRODUCTION.....	7
1.1	Concept for the Information Model.....	7
1.2	Presentation of the Document.....	10
2	COMMON DATA DEFINITIONS.....	11
2.1	Data Element Keys.....	11
	a) Data Element with “-key” Suffix.....	11
	b) Data Element with “-multi-key” Suffix.....	12
	c) Data Element with “-keys” Suffix.....	13
2.2	Commonly-used Data Elements.....	15
	a) Commonly-used Basic Data Elements.....	15
	b) Common-used Data Elements with Prefixes/Suffixes.....	15
2.3	Commonly-Used Complex Data Elements.....	16
2.3.1	Addresses.....	17
2.3.2	Batch Sizes.....	19
2.3.3	Boundary Element.....	20
2.3.4	Child Work Items.....	20
2.3.5	Contacts.....	21
2.3.6	Cost Categories.....	22
2.3.7	Cost Summary.....	23
2.3.8	Customers.....	24
2.3.9	Due Dates.....	25
2.3.10	Duration Distributions.....	25
2.3.11	Estimated Durations.....	26
2.3.12	Events.....	28
2.3.13	From Reference Frame.....	28
2.3.14	Organizational-contact.....	30
2.3.15	Parameters.....	30
2.3.16	Part Quantities.....	31
2.3.17	Plan Definition.....	32
2.3.18	Precedent Constraints.....	34
2.3.19	Reliability Statistics.....	36
2.3.20	Resources Assigned.....	37
2.3.21	Resources Required.....	40
2.3.22	Resources Used.....	42
2.3.23	Revisions.....	44
2.3.24	Setup Components.....	45
2.3.25	Setup Configurations.....	46
2.3.26	Status Code.....	47
2.3.27	Stock Level Quantities.....	47
2.3.28	Suppliers.....	48
2.3.29	Units of Measurement.....	49
2.3.30	Work Actual Progress.....	50
2.3.31	Work Assignments.....	51
2.3.32	Work Scheduled Progress.....	52
3	MACHINE SHOP DATA STRUCTURE.....	54
3.1	Organizations.....	56
	a) Unique Data Elements.....	57
	b) Data Element Keys, Multi-keys, and Key Lists.....	57
	c) Common Data Elements.....	57
3.2	Departments.....	57
	a) Unique Data Elements.....	58
	b) Data Element Keys, Multi-keys, and Key Lists.....	58
	c) Common Data Elements.....	59
3.3	Calendars.....	59
	a) Unique Data Elements.....	60

b) Data Element Keys, Multi-keys, and Key Lists	61
c) Common Data Elements.....	61
3.4 Resources.....	61
a) Unique Data Elements	62
b) Data Element Keys, Multi-keys, and Key Lists	62
c) Common Data Elements.....	62
3.5 Machines	62
a) Unique Data Elements	64
b) Data Element Keys, Multi-keys, and Key Lists	65
c) Common Data Elements.....	65
3.6 Stations	65
a) Unique Data Elements	67
b) Data Element Keys, Multi-keys, and Key Lists	67
c) Common Data Elements.....	67
3.7 Cranes.....	67
a) Unique Data Elements	68
b) Data Element Keys, Multi-keys, and Key Lists	69
c) Common Data Elements.....	69
3.8 Employees	69
a) Unique Data Elements	71
b) Data Element Keys, Multi-keys, and Key Lists	71
c) Common Data Elements.....	71
3.9 Tool Catalog.....	71
a) Unique Data Elements	72
b) Data Element Keys, Multi-keys, and Key Lists	72
c) Common Data Elements.....	73
3.10 Fixture Catalog	73
a) Unique Data Elements	74
b) Data Element Keys, Multi-keys, and Key Lists	74
c) Common Data Elements.....	74
3.11 Layout.....	74
a) Unique Data Elements	77
b) Data Element Keys, Multi-keys, and Key Lists	77
c) Common Data Elements.....	77
3.12 Skill Definitions.....	77
a) Unique Data Elements	78
b) Data Element Keys, Multi-keys, and Key Lists	78
c) Common Data Elements.....	78
3.13 Setup Definitions	78
a) Unique Data Elements	79
b) Data Element Keys, Multi-keys, and Key Lists	80
c) Common Data Elements.....	80
3.14 Operation Definitions	80
a) Unique Data Elements	81
b) Data Element Keys, Multi-keys, and Key Lists	81
c) Common Data Elements.....	81
3.15 Maintenance Definitions.....	81
a) Unique Data Elements	82
b) Data Element Keys, Multi-keys, and Key Lists	82
c) Common Data Elements.....	82
3.16 Parts	82
a) Unique Data Elements	84
b) Data Element Keys, Multi-keys, and Key Lists	84
c) Common Data Elements.....	84
3.17 Bills-of-Materials	84
a) Unique Data Elements	85

b) Data Element Keys, Multi-keys, and Key Lists	86
c) Common Data Elements.....	86
3.18 Inventory	86
a) Unique Data Elements	86
b) Data Element Keys, Multi-keys, and Key Lists	87
c) Common Data Elements.....	87
d) Example Part	87
3.19 Part Inventory	87
a) Unique Data Elements	88
b) Data Element Keys, Multi-keys, and Key Lists	88
c) Common Data Elements.....	88
3.20 Materials Inventory.....	88
a) Unique Data Elements	90
b) Data Element Keys, Multi-keys, and Key Lists	90
c) Common Data Elements.....	90
3.21 Tool Inventory	90
a) Unique Data Elements	92
b) Data Element Keys, Multi-keys, and Key Lists	92
c) Common Data Elements.....	92
3.22 Fixture Inventory	92
a) Unique Data Elements	94
b) Data Element Keys, Multi-keys, and Key Lists	94
c) Common Data Elements.....	94
3.23 Procurements	94
a) Unique Data Elements	96
b) Data Element Keys, Multi-keys, and Key Lists	96
c) Common Data Elements.....	96
3.24 Process Plans	96
a) Unique Data Elements	97
b) Data Element Keys, Multi-keys, and Key Lists	98
c) Common Data Elements.....	98
3.25 Routing Sheets.....	98
a) Unique Data Element	98
b) Data Element Keys, Multi-keys, and Key Lists	99
c) Common Data Elements.....	99
3.26 Operation Sheets.....	99
a) Unique Data Element	99
b) Data Element Keys, Multi-keys, and Key Lists	100
c) Common Data Elements.....	100
3.27 Machine Programs.....	100
a) Unique Data Element	101
b) Data Element Keys, Multi-keys, and Key Lists	101
c) Common Data Elements.....	101
3.28 Work.....	101
a) Unique Data Elements	102
b) Data Element Keys, Multi-keys, and Key Lists	102
c) Common Data Elements.....	102
3.29 Orders	102
a) Unique Data Elements	104
b) Data Element Keys, Multi-keys, and Key Lists	104
c) Common Data Elements.....	104
3.30 Order Items.....	104
a) Unique Data Elements	106
b) Data Element Keys, Multi-keys, and Key Lists	106
c) Common Data Elements.....	106
3.31 Jobs.....	106

a) Unique Data Elements	108
b) Data Element Keys, Multi-keys, and Key Lists	108
c) Common Data Elements.....	108
3.32 Tasks.....	108
a) Unique Data Elements	109
b) Data Element Keys, Multi-keys, and Key Lists	109
c) Common Data Elements.....	110
3.33 Maintenance Orders.....	110
a) Unique Data Elements	111
b) Data Element Keys, Multi-keys, and Key Lists	111
c) Common Data Elements.....	111
3.34 Pick Orders	111
a) Unique Data Elements	113
b) Data Element Keys, Multi-keys, and Key Lists	113
c) Common Data Elements.....	113
3.35 Tool Orders.....	113
a) Unique Data Elements	115
b) Data Element Keys, Multi-keys, and Key Lists	115
c) Common Data Elements.....	115
3.36 Schedules.....	115
a) Unique Data Elements	118
b) Data Element Keys, Multi-keys, and Key Lists	119
c) Common Data Elements.....	119
3.37 Time Sheets	119
a) Unique Data Elements	120
b) Data Element Keys, Multi-keys, and Key Lists	121
c) Common Data Elements.....	121
3.38 References	121
a) Unique Data Elements	122
b) Data Element Keys, Multi-keys, and Key Lists	123
c) Common Data Elements.....	123
3.39 Probability Distributions.....	123
a) Unique Data Elements	124
b) Data Element Keys, Multi-keys, and Key Lists	124
c) Common Data Elements.....	124
4 REFERENCES	125

1 INTRODUCTION

A Machine Shop Information model has been developing at the National Institute of Standards and Technology (NIST) as a part of efforts that support the development of standard data interfaces. This work is being carried out as part of the Software Engineering Institute's Technology, Insertion, Development, and Evaluation (TIDE) Program. The information model is intended to be used for representing and exchanging machine shop data, initially between manufacturing execution, scheduling, and simulation systems. The model is presented in two formats: the Unified Modeling Language (UML) [OMG 2003] and the Extensible Markup Language (XML) [W3C 2000]. The information model, when completed, must satisfy the following needs:

- support data requirements for the entire simulation manufacturing life cycle,
- enable data exchange between simulation and other manufacturing software for machine shops,
- provide for the construction of machine shop simulators, and
- support testing and evaluation of machine shops' manufacturing software.

The Machine Shop Information model contains five major supporting data structures and fifteen major manufacturing data structures. Supporting data structures include:

- time sheets,
- probability distributions,
- revisions,
- references, and
- units of measurement.

Manufacturing data structures include:

- organizations,
- calendars,
- resources (machines, machines, cranes, employees, tool catalog, and fixture catalog),
- skill definitions,
- setup definitions,
- operation definitions,
- maintenance definitions,
- layout,
- parts,
- bills of materials,
- inventory,
- procurements,
- process plans (routing sheets, operation sheets, and machine programs),
- work (orders, order items, jobs, tasks, maintenance orders, pick orders, and tool orders), and
- schedule.

1.1 Concept for the Information Model

In this section, the concept of the shop information model from the user perspective is described. The primary objective was to develop a structure for exchanging shop data between various manufacturing software applications, including simulation. The idea was to use the same data structures for managing actual production operations and simulating the machine shop. The rationale was that if one structure can serve both purposes, the need for translation and abstraction of the real data would be minimized when simulations are constructed. The mapping of real world data into simulation abstractions is not, for the most part, addressed in the current information model.

It is also recognized that maintaining data integrity and minimizing the duplication of data were important requirements. For this reason, each unique piece of information appears in only one place in the model. Cross-reference links are used to avoid the creation of redundant copies of data.

The machine shop information model contains twenty major elements. Each of the major data elements are italicized in the discussion that follows. The data elements are called: *organizations*, *calendars*, *resources*, *skill-definitions*, *setup-definitions*, *operation-definitions*, *maintenance-definitions*, *layout*, *parts*, *bills-of-materials*, *inventory*, *procurements*, *process-plans*, *work*, *schedules*, *revisions*, *time-sheets*, *probability-distributions*, *references*, and *units-of-measurement*. Figure 1 illustrates some of the major elements of the conceptual information model and their relationships to each other.

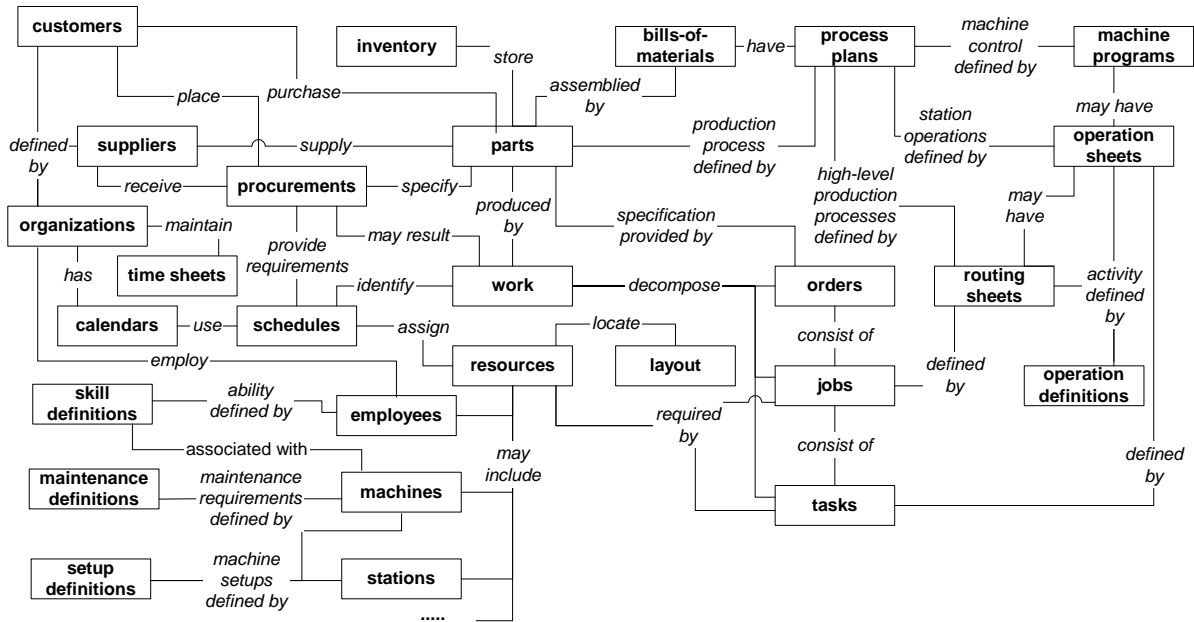


Figure 1. Concept for the Machine Shop Information Model

Perhaps a good place to start the discussion of the information model is with the customer. Machine shops are businesses. They typically produce machined parts for either internal or external customers. Data elements are needed to maintain information on customers. The types of organizational information that is needed about customers is very similar to the data needed about suppliers that provide materials to the shop. The same types of organizational data are also needed about the machine shop itself. For this reason, an *organizations* element was created to maintain organizational and contact information on the shop, its customers, and its suppliers.

Organizations can be thought of as both a phone book and an organization chart. The element provides sub-elements for identifying departments, their relationships to each other, individuals within departments, and their contact information. Various other types of information needs to be cross-referenced to organizations and contacts within structure, e.g., customer orders, parts, and procurements to suppliers.

The operation of the machine shop revolves around the production of parts, i.e., the fabrication of parts from raw materials such as metal or plastic. The raw materials typically come in the form of blocks, bars, sheets, forgings, or castings. These materials are themselves parts that are procured from suppliers. The *parts* data element was created to maintain the broad range of information that is needed about each part that is handled by the machine shop. Part data includes an identifying part number, name, description, size, weight, material composition, unit-of-issue, cost, group technology classification codes, and revision (change) data. Cross-reference links are needed to the customers that buy the parts from the shop and/or the suppliers that provide them as raw materials. Links are also needed to other data elements, documents, and files that are related to the production of parts including: part specification documents, geometric models, drawings, bills-of-materials, and process plans.

The *bills-of-materials* element is basically a collection of hierarchically-structured parts lists. It is used to define the parts and subassemblies that make up higher-level part assemblies. A bill-of-materials identifies, by a part number reference link, the component or subassembly required at each level of assembly. The quantity required for each part is also indicated. Cross-references links are needed between parts that are assemblies and their associated bill-of-materials.

The *parts* and *bills-of-materials* elements establish the basic definition of parts produced or used by the shop. Another element, *inventory*, is used to identify quantity of part instances at each location within the facility. *Inventory* data elements are provided for parts, tools, fixtures, and materials. Materials are defined as various types of stock that may be partially consumed in production, e.g., sheets, bars, and rolls. Structures are provided within inventory to keep track of various stock levels (e.g., reorder point level) and the specific instances of parts that are used in assemblies.

The *procurements* element identifies the internal and external purchase orders that have been created to satisfy order or part-inventory requirements. Cross-reference links are defined to *Parts* to identify the specific parts that are being procured and to *work* to indicate which work items they will be used to satisfy.

The *work* data element is used to specify a hierarchical collection of work items that define orders, production and support activities within the shop. Support activities include maintenance, inventory picking, and fixture/tool preparation. *Work* is broken down hierarchically into orders, jobs, and tasks.

Orders may be either customer orders for products or internally-generated orders to satisfy part requirements within the company, e.g., maintenance of inventory levels of stock items sold through a catalog. Orders contain both definition and status information. Definition information specifies whom the order is for (i.e., customer cross-references), its relative priority, critical due dates, what output products are required (a list of order items, i.e., part references and quantities required), special resource requirements, precedence relationships on the processing of order items, and a summary of estimated and actual costs. Order items are also cross-referenced to jobs and tasks that decompose the orders into individual process steps performed at workstations within the shop. Status information includes data about scheduled and actual progress towards completing the order.

Jobs typically define complex production work items that involve activities at multiple stations and ultimately produce parts. Tasks are lower-level work items that are typically performed at a single workstation or area within the shop.

The *process-plans* element contains the process specifications that describe how production and support work is to be performed in the shop. Major elements contained within *process-plans* include routing sheets, operation sheets, and equipment programs. Routing and operation sheets are the plans used to define job- and task-level work items, respectively, in the work hierarchy. These process plans define the steps, precedence constraints between steps, and resources required to produce parts and perform support activities. Precedence constraints defined in a process plan are used to establish precedence relationships between jobs and tasks. Equipment program elements establish cross reference links to files that contain computer programs that are used to run machine tools and other programmable equipment that process specific parts. Each part in the *parts* element contains cross-reference links to the process plans that define how to make that part. Jobs and tasks contain links back to the process plans that defined them.

The *resources* element is used to define production and support resources that may be assigned to jobs or tasks in the shop, their status, and scheduled assignments to specific work items. The resource types available in the machine shop environment include: stations and machines, cranes, employees, tool and tool sets, fixtures and fixture sets.

The *skill-definitions*, *setup-definitions*, *operations-definitions*, *maintenance-definitions*, and *time-sheets* elements provide additional supporting information associated with resources. *Skill-definitions* lists the skills that an employee may possess and the levels of proficiency associated with these skills. Skills are referenced in employee resource requirements contained in process plans. *Setup-definitions* typically specifies tool or fixture setups on a machine. Tool setups are typically the tools that are required in the tool

magazine. Fixture setups are work-holding devices mounted on the machine. Setups may also apply to cranes or stations. *Operation-definitions* specifies the types of operations that may be performed at a particular station or group of stations within the shop. *Maintenance-definitions* specifies preventive or corrective maintenance to be done on machines or other maintained resources. *Time-sheets* is used to log individual employee's work hours, leave hours, overtime hours, and so on.

The *layout* element defines the physical locations of resource objects and part instances within the shop. It also defines reference points, area boundaries, and paths.. It contains references to external files that are used to further define resource and part objects using appropriate graphics standards. Cross-references links are also provided between layout objects and the actual resources that they represent.

Schedules and *calendars* data elements deal with time. *Schedules* provides two views of the planned assignment of work and resources. Work items (orders, jobs, and tasks) are mapped to resources, and conversely, resources are mapped to work items. The planned time events associated with those mappings are also identified, e.g., scheduled start times and end times. *Calendars* identifies scheduled work days for the shop, the shift schedules that are in effect for periods of time, planned breaks, and holiday periods.

The four remaining major data elements are *revisions*, *references*, *probability-distributions*, and *units-of-measurement*. The *revisions* element is used repeatedly throughout many levels of the information model. It provides a mechanism for identifying versions of subsets of the data, revision dates, and the creator of the data. The *references* element identifies external digital files and paper documents that support and further define the data elements contained within the shop data structure. It provides a mechanism for linking to outside files that conform to various other format specifications or standards, e.g., STEP (or the International Organization for Standardization 10303, informally known as the STandard for the Exchange of Product model data, STEP) part design files [ISO 1994]. The *probability-distributions* element defines probability distributions that are used to vary processing times, breakdown and repair times, availability of resources among others. Distributions may be cross-referenced from elsewhere in the model, e.g., equipment resources maintenance data. *Units-of-measurement* specifies the units used in the file for various quantities such as length, weight, currency, and speed.

1.2 Presentation of the Document

The document provides definitions to all data elements presented in the Machine Shop Information Model. Data elements within the model are grouped into three categories: data element keys, commonly-used data elements, and unique manufacturing shop data elements. Section 2 provides definitions to the data elements that are found repeatedly at many levels of the data structures within the information model, in this document these data elements are called data elements keys and commonly-used data elements. Section 3 provides definitions to data elements uniquely specified in the data structures of the information model. Appendix A lists the enumeration items for each enumeration data element. Appendix B presents the data format for each selected data element.

2 COMMON DATA DEFINITIONS

There are data elements that are found repeatedly at many levels of the data structures within the information model, they are topic of this section. In this section, data elements keys and commonly-used data elements are identified and defined. Data element keys are data elements that serve as individual pointers or a collection of pointers to the information model. The commonly-used data elements are presented according to the elements' data types, including basic data elements, data elements with prefixes/suffixes, and complex data elements. A data element may be a basic data element, which cannot be further subdivided into elements, or a complex data element, which contains more than one basic data elements.

2.1 Data Element Keys

This subsection presents a set of data element keys that are used to avoid of redefining the same data elements. The data is grouped into three categories based on each data element's name suffix: "-key", "-multi-key" or "-keys".

a) Data Element with "-key" Suffix

The data element served as a pointer to the model to avoid of redefining the same data element. The name of this type of data element clearly identifies its source data element. Each source data element can be found at somewhere within the information model. Each of these "-key" suffix data elements is presented with two attributes: identifier and number. Here lists data elements of this type that present in the information model:

```
<bill-of-materials-key />
<bom-element-key />
<calendar-key />
<contact-key />
<crane-key />
<department-key />
<employee-key />
<fixture-definition-key />
<fixture-instance-key />
<fixtureset-definition-key />
<job-key />
<level-definition-key />
<machine-group-key />
<machine-key />
<machine-program-key />
<maintenance-definition-key />
<maintenance-order-key />
<operation-definition-key />
<operation-sheet-key />
<order-item-key />
<order-key />
<organization-key />
<part-key />
<part-instance-key />
<part-inventory-item-key />
<pick-order-key />
<plan-step-key />
<position-description-key />
<position-key />
<precedent-plan-step-key />
<probability-distribution-key />
```

```

<process-plan-key />
<procurement-key />
<procurement-item-key />
<reference-frame-key />
<reference-key />
<resource-key />
<routing-sheet-key />
<setup-definition-key />
<shift-key />
<shift-schedule-key />
<skill-definition-key />
<station-group-key />
<station-key />
<task-key />
<tool-definition-key />
<tool-instance-key />
<tool-order-key />
<toolset-definition-key />
<work-item-group-key>

```

b) Data Element with “-multi-key” Suffix

Each data element groups two or more “key” suffix data elements to represent a compound data element. Here lists this group’s data elements that present in the information model:

```

<calendar-multi-key>
  <calendar-key />
  <shift-schedule-key />
  <shift-key />
</calendar-multi-key>

<child-department-multi-key>
  <department-key />
</child-department-multi-key>

<component-part-multi-key>
  <part-key />
  <part-inventory-item-key />
  <part-instance-key />
</component-part-multi-key>

<order-multi-key>
  <order-key />
  <order-item-key />
</order-multi-key>

<position-multi-key>
  <organization-key />
  <department-key />
  <position-key />
</position-multi-key>

<process-plan-multi-key>
  <process-plan-key />
  <routing-sheet-key />
  <operation-sheet-key />

```

```

    <machine-program-key />
  </process-plan-multi-key>

  <procurement-multi-key>
    <procurement-key />
    <procurement-item-key />
  </procurement-multi-key>

  <skill-multi-key>
    <skill-definition-key />
    <level-definition-key />
  </skill-multi-key>

```

c) Data Element with “-keys” Suffix

Each data element is a collection of “key” suffix data elements for convenient purposes. Here lists this group’s data elements that present in the information model:

```

  <child-setup-keys>
    <setup-definition-key />
  </child-setup-keys>

  <component-part-multi-keys>
    <component-part-multi-key />
  </component-part-multi-keys>

  <crane-keys>
    <crane-key />
  </crane-keys>

  <employee-keys>
    <employee-key />
  </employee-keys>

  <fixture-instance-keys>
    <fixture-instance-key />
  </fixture-instance-keys>

  <fixtureset-member-keys>
    <fixture-definition-key />
    <fixtureset-definition-key />
  </fixtureset-member-keys>

  <locations-served-keys>
    <station-key />
    <machine-key />
  </locations-served-keys>

  <machine-keys>
    <machine-key />
    <machine-group-key />
  </machine-keys>

  <maintained-resource-keys>
    <machine-key />
    <crane-key />

```

```

    <station-key />
</maintained-resource-keys>

<parent-setup-keys>
  <setup-definition-key />
</parent-setup-keys>

<position-multi-keys>
  <position-multi-key />
</position-multi-keys>

<process-plan-multi-keys>
  <process-plan-multi-key />
</process-plan-multi-keys>

<procurement-multi-keys>
  <procurement-multi-key />
</procurement-multi-keys>

<reference-keys>
  <reference-key />
</reference-keys>

<setup-definition-keys>
  <setup-definition-key />
</setup-definition-keys>

< setup-resource-keys>
  <machine-key />
  <machine-group-key />
  <station-key />
  <station-group-key />
  <crane-key />
</setup-resource-keys>

<skill-multi-keys>
  <skill-multi-key />
</skill-multi-keys>

<station-keys>
  <station-key />
  <station-group-key />
</station-keys>

<tool-instance-keys>
  <tool-instance-key />
</tool-instance-keys>

<toolset-member-keys>
  <tool-definition-key />
  <toolset-definition-key />
</toolset-member-keys>

<work-keys>
  <order-multi-key />
  <job-key />

```



```
<task-key />
<procurement-multi-key />
<pick-order-key />
<tool-order-key />
<maintenance-order-key />
</work-keys>
```

2.2 Commonly-used Data Elements

This subsection presents a set of data that is found repeatedly at many levels of the data structure within the model. The data is grouped into three categories: basic data elements, data elements with prefixes/suffixes, and complex data elements. Basic data elements can not be further subdivided into elements in the shop information model. Data elements with prefixes/suffixes use data elements with unique modifiers, either in prefixing or suffixing, for similar but unique purposes; the definitions of these data elements are generally self-explained. Complex data are constructed in a structure form with two or more basic data elements.

a) Commonly-used Basic Data Elements

Commonly-used basic data elements are listed and defined in this subsection:

comment: a remark or an annotation.

description: text that relates to the nature of the subject.

hourly-rate: a charge or payment calculated by hour.

identifier: a symbol or code number used to uniquely identify the object internally within the system. An *identifier* is generated automatically by the system when the objects are created.

logical-relationship: a mechanism for representing a logical relationship (i.e., AND, OR) between two objects.

name: a word or group of words used to identify the object by the user or user organization. A *name* element is provided for readability by the user or user organization.

nominal-duration: a stated duration time as opposed to the real duration time.

number: a symbol or code number used to identify the object to the user or the user organization. It should be a unique key for identifying the object either when taken alone or possibly together with the object type.

priority-rating: a rank value for representing order of urgency or importance associated with the work item initiated by a department, e.g., production jobs.

repetition-count: the number of times the production activities need to be repeated to satisfy the work requirement. For example: if the part quantities required is 50 and the batch size for the routing sheet is 10, then the repetition count would be 5.

quantity: a specified number or amount of a particular item.

type: a classification code used to categorize the object.

b) Common-used Data Elements with Prefixes/Suffixes

Commonly-used data elements with prefixes or suffixes are listed and defined in this subsection:

code: a system of symbols, letters, or words that representing certain object. Prefixes to *code* include: *color-code*, *cost-code*, *function-code*, *operation-code*, *shape-code*, *status-code*, *weight-code*, etc.

coefficient: a numerical factor used with an equation or expression. Prefixes to *coefficient* include: *unit X-coefficient*, *y-coefficient*, *z-coefficient*, *alpha-coefficient*, *beta-coefficient*, *gamma-coefficient*, etc.

cost: an amount paid or to be paid for a purchase or a service. Prefixes to *cost* include: *unit-cost*, *material-cost*, *labor-cost*, *equipment-usage-cost*, *total-cost*, etc.

date: the date when the activity effective. Prefixes to *date* include: *dates*, *start-date*, *end-date*, *effective-start-date*, *effective-end-date*, *release-date*, *request-earliest-ship-date*, *request-latest-ship-date*, *material-availability-date*, etc. The format for *date* is *yyyy-mm-dd* where *yyyy* is for representing year, *mm* for month, and *dd* for the day of the month.

duration: the period of time during which some activity performs or lasts. Prefixes to *duration* include: *estimated-duration*, *nominal-duration*, etc.

location: a place where something is or might be located. Prefixes to *location* include *current-location*, etc.

maximum: the maximum required number of a particular object. Suffixes to *maximum* include: *maximum-stations*, *maximum-machines*, *maximum-employees*, etc.

minimum: the minimum required number of a particular object. Suffixes to *minimum* include: *minimum-stations*, *minimum-machines*, *minimum-employees*, etc.

number: a system of symbols, letters, or words that representing certain object. Prefixes to *number* include: *day-number*, *maximum-level-number*, *minimum-level-number*, *model-number*, *serial-number*, etc.

number-of: a total number of a particular object. Suffixes to *number-of* include: *number-of-stations*, *number-of-machines*, *number-of-employees*, *number-of-cranes*, etc.

rate: a quantity measured in terms of another measured quantity. Prefixes to *rate* include: *hourly-rate*, *overtime-rate*, *annual-absentee-rate*, etc.

state: a stage or condition of development. Prefixes to *state* include: *operational-state*, *maintenance-state*, etc.

time: the time of the date when the activity effective. Prefixes to *time* include: *start-time*, *end-time*, etc. The format for *time* is *hh:mm:ss* where *hh* is for representing hour ranging between 0 and 24, *mm* is for representing minute ranging between 0 and 60, and *ss* is operational and is for representing minute ranging between 0 and 60.

type: a classification code used to categorize the object. Prefixes to *type* include: *controller-type*, *operation-type*, *operation-subtype*, *material-type*, *salary-type*, etc.

units describes the units for a particular measurement. Prefixes to *units* include: *angular-units*, *currency-units*, *distance-units*, *length-units*, *mass-units*, *speed-units*, *time-duration-units*, and *volume-units*.

value: a numerical measure of the subject. Prefixes to *value* include: *value1*, *value2*, *value3*, etc. *value1*, *value2*, *value3* are used to specify coefficients for a probability distribution.

2.3 Commonly-Used Complex Data Elements

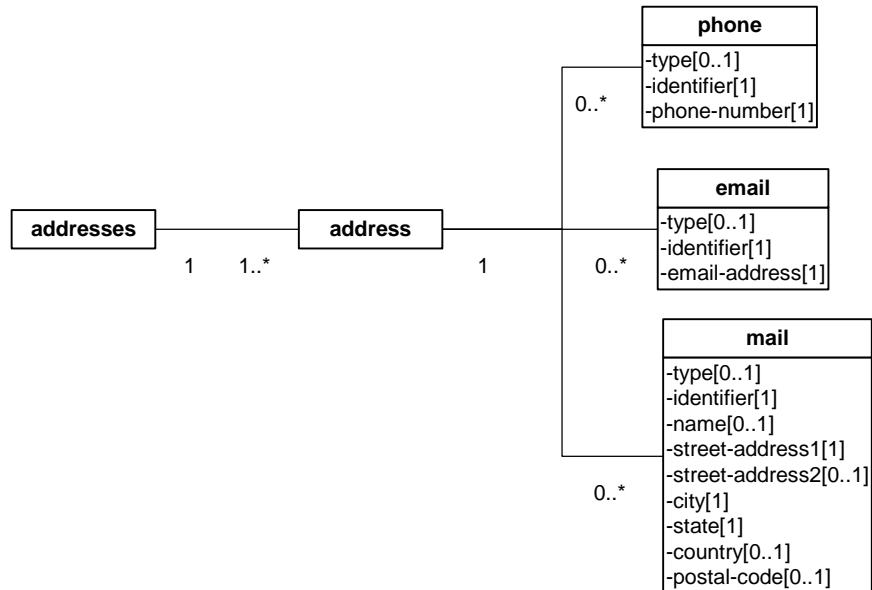
Commonly-used complex data elements are defined in the following subsections:

- [2.3.1 addresses](#)
- [2.3.2 batch-sizes](#)
- [2.3.3 boundary-elements](#)
- [2.3.4 child-work-items](#)
- [2.3.5 contacts](#)
- [2.3.6 cost-categories](#)
- [2.3.7 cost-summary](#)
- [2.3.8 customers](#)
- [2.3.9 due-dates](#)
- [2.3.10 duration-distributions](#)
- [2.3.11 estimated-durations](#)
- [2.3.12 events](#)
- [2.3.13 from-reference-frame](#)
- [2.3.144 organizational-contact](#)
- [2.3.155 parameters](#)
- [2.3.16 part-quantities](#)
- [2.3.17 plan-definition](#)
- [2.3.18 precedent-constraints](#)
- [2.3.19 reliability-statistics](#)
- [2.3.20 resources-assigned](#)
- [2.3.21 resources-required](#)
- [2.3.22 resources-used](#)
- [2.3.23 revisions](#)
- [2.3.24 setup-components](#)
- [2.3.25 setup-configurations](#)
- [2.3.26 status-code](#)
- [2.3.27 stock-level-quantities](#)
- [2.3.28 suppliers](#)
- [2.3.29 units-of-measurement](#)
- [2.3.30 work-actual-progress](#)
- [2.3.31 work-assignments](#)
- [2.3.32 work-scheduled-progress](#)

2.3.1 Addresses

addresses specifies contact information of an organization, an organization unit, or an individual. It is represented by a set of *address* data elements.

The UML model for the *addresses* element is shown below:



The XML structure for the *addresses* element is shown below:

```

<addresses>
  <address>
    <phone type="" identifier="">
      <phone-number />
    </phone>
    <email type="" identifier="">
      <email-address />
    </email>
    <mail type="" identifier="">
      <name />
      <street-address1 />
      <street-address2 />
      <city />
      <state />
      <country />
      <postal-code />
    </mail>
  </address>
</addresses>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

address: information about phone number, email addresses and mailing addresses. It is represented by the following data elements: *phone*, *email*, and *mail*.

addresses: a collection of individual addresses.

city: the city the address belongs.

country: the country the address belongs.

email: information about the email address. It is represented by the following data elements: *type*, *identifier*, and *email-address*.

mail: information about the mailing address. It is represented by the following data elements: *type*, *identifier*, *name*, *street-address1*, *street-address2*, *city*, *state*, *country*, and *postal-code*.

phone: information about the phone number. It is represented by the following data elements: *type*, *identifier*, and *number*.

state: the state the address belongs.

street-address1: the first part of the street address of the mailing address.

street-address2: the second part of the street address of the mailing address.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

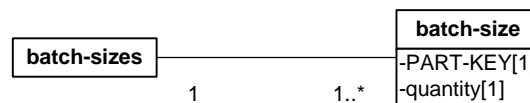
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *identifier*, *name*, *number*, *postal-code*, and *type*.

2.3.2 Batch Sizes

batch-sizes identifies the type and total quantity of each part. It is represented by a set of *batch-size* data elements.

The UML model for the *batch-sizes* element is shown below:



The XML structure for the *batch-sizes* element is shown below:

```
<batch-sizes>
  <batch-size>
    <part-key />
    <quantity />
  </batch-size>
</batch-sizes>
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

batch-size: the type and quantity of a particular part. It is represented by the following data elements: *part-key* and *quantity*.

batch-sizes: a collection of individual batch-sizes.

b) Data Element Keys, Multi-keys, and Key Lists

part-key is used in this element.

c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *quantity*.

2.3.3 Boundary Element

boundary-element specifies information about a boundary reference point. It is used to define *area-boundary*, *path-route*, *object-boundary*, and *object-boundary*.

The UML model for the *boundary-element* element is shown below:

boundary-element
-type[0..1]
-identifier[1]
-number[1]
-name[0..1]

The XML structure for the *batch-sizes* element is shown below:

```
<boundary-element type="" identifier="" number="">  
  < name />  
</boundary-element>
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

boundary-element: information about a reference point of an area, a path, or an object.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

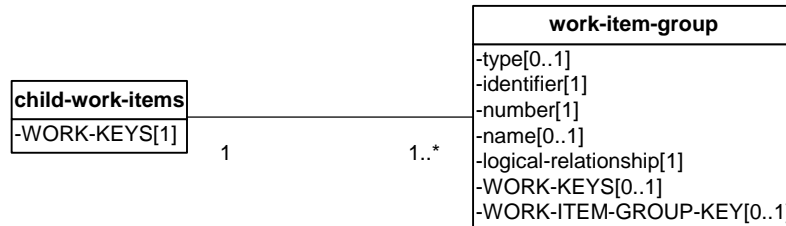
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *identifier*, *name*, *number*, and *type*.

2.3.4 Child Work Items

child-work-items specifies a collection of subordinate (up to one level down) work items. A sample of a subordinate work item is that a task can be a child work item of a job. *child-work-items* is represented by the following data elements: *work-keys* and *work-item-group*.

The UML model for the *child-work-items* element is shown below:



The XML structure for the *child-work-items* element is shown below:

```

<child-work-items>
  <work- keys />
  <work-item-group type="" identifier="" number="">
    <name />
    <logical-relationship />
    <work-keys />
    <work-item-group-key />
  </work-item-group>
</child-work-items>
  
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

child-work-items: a collection of subordinate (up to one level down) work items.

work-item-group: information that represents a set of work items and the logical relationships among these work items. It is represented by the following data elements: *type*, *identifier*, *number*, *name*, *logical-relationship*, *work-keys* and *work-item-group-key*.

b) Data Element Keys, Multi-keys, and Key Lists

work-item-group-key and *work-keys* are used in this element.

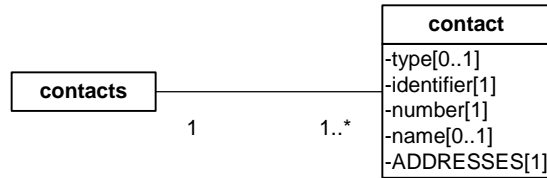
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *type*, *identifier*, *logical-relationship*, *name*, *number*, and *work-keys*.

2.3.5 Contacts

contacts is a collection of contact information, including phone numbers, email addresses, and mailing addresses, of an organization or an organization unit. It is represented by a set of *contact* data elements.

The UML model for the *contacts* element is shown below:



The XML structure for the *contacts* element is shown below:

```

<contacts>
  <contact type="" identifier="" number="">
    <name />
    <addresses />
  </contact>
</contacts>
  
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

contact: contact information of an organization or an organization unit. The information includes phone numbers, email addresses, and mailing addresses. It is represented by the following data elements: *type*, *identifier*, *number*, *name*, and *addresses*.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

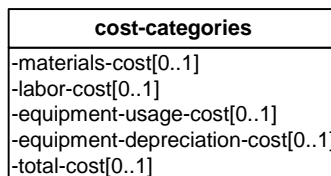
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *addresses*, *identifier*, *name*, *number*, and *type*.

2.3.6 Cost Categories

cost-categories specifies the cost figures in the following categories: materials, labor, equipment usage, equipment depreciation cost, and total. It is represented by the following data elements: *materials-cost*, *labor-cost*, *equipment-usage-cost*, and *total-cost*.

The UML model for the *cost-categories* element is shown below:



The XML structure for the *cost-categories* element is shown below:

```

<cost-categories>
  <materials-cost />
  
```



```

    <labor-cost />
    <equipment-usage-cost />
    <equipment-depreciation-cost />
    <total-cost />
  </cost-categories>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

cost-categories: information about costs in the categories of materials, labor, equipment usage, equipment depreciation, and total.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

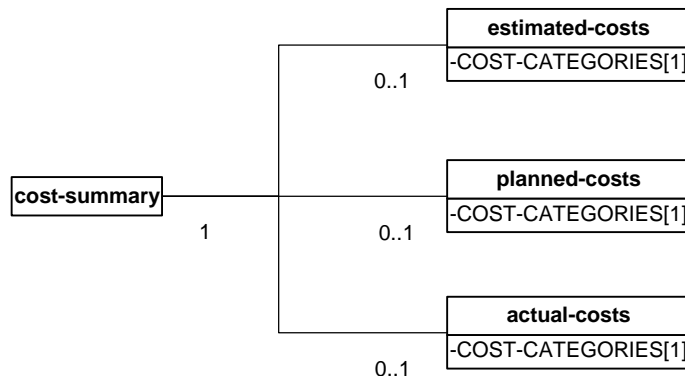
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *equipment-usage-cost*, *labor-cost*, *materials-cost*, and *total-cost*.

2.3.7 Cost Summary

cost-summary specifies a summary of costs at different manufacturing stages: estimation, planning and actual expenses. It is represented by the following data elements: *estimated-costs*, *planned-costs*, and *actual-costs*.

The UML model for the *cost-summary* element is shown below:



The XML structure for the *cost-summary* element is shown below:

```

<cost-summary>
  <estimated-costs>
    <cost-categories />
  </estimated-costs>
  <planned-costs>
    <cost-categories />
  </planned-costs>
  <actual-costs>

```

```

    <cost-categories />
  </actual-costs>
</cost-summary>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

actual-costs: information about actual expenses, in several spending categories, such as materials, labor, etc. It is represented by the *cost-categories* data element.

cost-summary: information about costs at different manufacturing stages including estimation, planning and final stages.

estimated-costs: information about the costs, in several categories, that are calculated by estimation. It is represented by the *cost-categories* data element.

planned-costs: information about costs, in several categories, that are calculated when a planning is performed. It is represented by the *cost-categories* data element.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

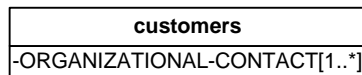
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *cost-categories*.

2.3.8 Customers

customers specifies contact information about an organization's customers. It is represented by a set of *organizational-contact* data elements.

The UML model for the *customers* element is shown below:



The XML structure for the *customers* element is shown below:

```

<customers>
  <organizational-contact />
</customers>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

customers: a collection of individual customers.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *organizational-contact*.

2.3.9 Due Dates

due-dates specifies a set of dates that are requested or estimated. These dates are normally associated with a work item. It is represented by the following data elements: *release-date*, *requested-earliest-ship-date*, and *requested-latest-ship-date*.

The UML model for the *due-dates* element is shown below:

due-dates
-release-date[0..1]
-requested-earliest-ship-date[0..1]
-requested-latest-ship-date[0..1]

The XML structure for the *due-dates* element is shown below:

```
<due-dates>
  <release-date />
  <requested-earliest-ship-date />
  <requested-latest-ship-date />
</due-dates>
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

due-dates: a collection of different types of due dates including release due date, requested earliest shipping date, and requested latest shipping date.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

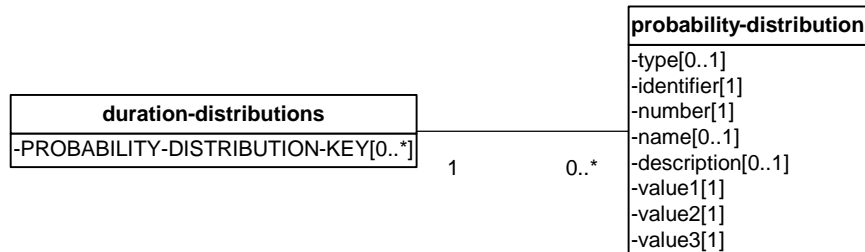
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *release-date*, *requested-earliest-ship-date*, and *requested-latest-ship-date*.

2.3.10 Duration Distributions

duration-distributions specifies distributions that are used to vary time duration, such as processing times, breakdown and repair times, etc. It is represented by the following data elements: *probability-distribution-key* and *probability-distribution*.

The UML model for the *duration-distributions* element is shown below:



The XML structure for the *duration-distributions* element is shown below:

```

<duration-distributions>
  <probability-distribution-key>
    <probability-distribution type="" identifier="" number="">
      <name />
      <description />
      <value1 />
      <value2 />
      <value3 />
    </probability-distribution>
</duration-distributions>
  
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

duration-distributions: a collections of various probability distributions.

probability-distribution: a distribution that is used to vary processing times, breakdown and repair times, etc. It is represented by the following data elements: *type*, *identifier*, *number*, *name*, *description*, *value1*, *value2*, and *value3* where *value1*, *value2* and *value3* are distribution coefficients.

b) Data Element Keys, Multi-keys, and Key Lists

probability-distribution-key is used in this element.

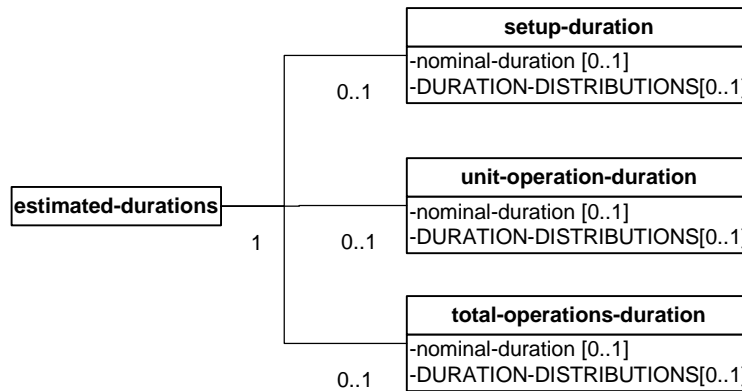
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *description*, *identifier*, *name*, *number*, *type*, *value1*, *value2*, and *value3*.

2.3.11 Estimated Durations

estimated-durations specifies distributions that are used to vary time duration for setup, unit operation, and total operations during a plan step. It is represented by the following data elements: *setup-duration*, *unit-operation-duration*, and *total-operations-durations*.

The UML model for the *estimated-durations* element is shown below:



The XML structure for the *estimated-duration* element is shown below:

```

<estimated-durations>
  <setup-duration>
    <nominal-duration />
    <duration-distributions />
  </setup-duration>
  <unit-operation-duration>
    <nominal-duration />
    <duration-distributions />
  </unit-operation-duration>
  <total-operations-duration>
    <nominal-duration />
    <duration-distributions />
  </total-operations-duration>
</estimated-durations>
  
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

estimated-durations: a collection of time duration estimations for setup duration, unit operation duration, and total operations.

setup-duration: an estimated time required to perform machine setups. It is represented by the following data elements: *nominal-duration* and *duration-distributions*.

total-operations-duration: an estimated time required to perform a complete set of operations. It is represented by the following data elements: *nominal-duration* and *duration-distributions*.

unit-operation-duration: an estimated time required to perform an unit of a particular operation. It is represented by the following data elements: *nominal-duration* and *duration-distributions*.

b) Data Element Keys, Multi-keys, and Key Lists

No data element keys are used in this element.

c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *duration-distributions* and *nominal-duration*.

2.3.12 Events

events specifies a set of time tags. *events* is represented by a set of *event* data elements.

The UML model for the *events* element is shown below:



The XML structure for the *events* element is shown below:

```
<events>
  <event type="" identifier="" number="">
    <name />
    <date />
    <time />
  </event>
</events>
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

event: information about a time tag. It is presented by the following data elements: *type*, *identifier*, *number*, *name*, *date*, and *time*.

events: a collection of individual events.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

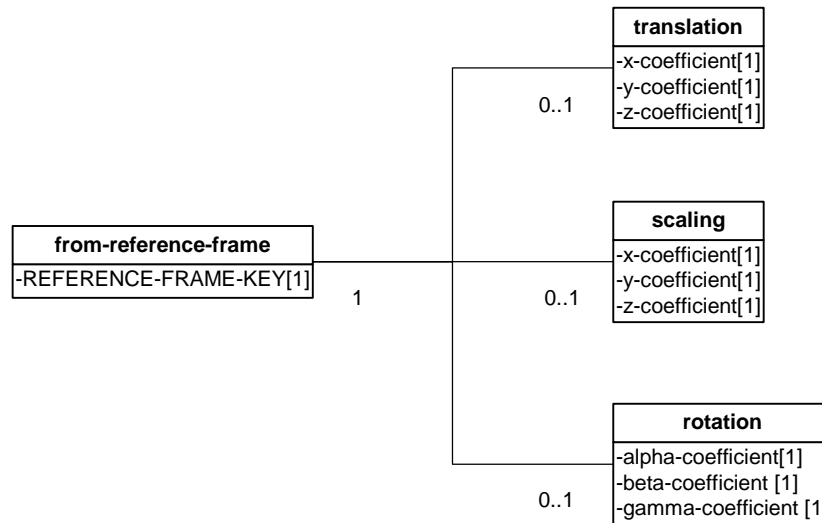
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *date*, *identifier*, *name*, *number*, *time* and *type*.

2.3.13 From Reference Frame

from-reference-frame specifies information about coordinate transformations, including translation, scaling, and rotation, from a reference frame. It is represented by the following data elements: *reference-frame-key*, *translation*, *scaling*, and *rotation*.

The UML model for the *from-reference-frame* element is shown below:



The XML structure for the *from-reference-frame* element is shown below:

```

<from-reference-frame>
  <reference-frame-key />
  <translation>
    <x-coefficient />
    <y-coefficient />
    <z-coefficient />
  </translation>
  <scaling>
    <x-coefficient />
    <y-coefficient />
    <z-coefficient />
  </scaling>
  <rotation>
    <alpha-coefficient />
    <beta-coefficient />
    <gamma-coefficient />
  </rotation>
</ from-reference-frame >
  
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

from-reference-frame: information about a coordinate transformation related to rotation, scaling, or translation.

rotation: information about a rotation transformation performed onto a coordinate system. It is represented by the following data elements: *x-coefficient*, *y-coefficient*, and *z-coefficient*.

scaling: information about a scaling transformation performed onto a coordinate system. It is represented by the following data elements: *x-coefficient*, *y-coefficient*, and *z-coefficient*.

translation: information about a translation transformation performed onto a coordinate system. It is represented by the following data elements: *x-coefficient*, *y-coefficient*, and *z-coefficient*.

b) Data Element Keys, Multi-keys, and Key Lists

reference-frame-key is used in this element.

c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *alpha-coefficient*, *beta-coefficient*, *gamma-coefficient*, *x-coefficient*, *y-coefficient*, and *z-coefficient*.

2.3.14 Organizational-contact

organizational-contact specifies the contact information of a particular organization and/or department. The contact information includes phone numbers, email addresses, and mailing addresses. It is represented by the following data elements: *organization-key*, *department-key*, and *contact-key*.

The UML model for the *organizational-contact* element is shown below:

organizational-contact
-ORGANIZATION-KEY[1]
-DEPARTMENT-KEY[0..1]
-CONTACT-KEY[0..1]

The XML structure for the *organizational-contact* element is shown below:

```
<organizational-contact >  
  <organization-key />  
  <department-key />  
  <contact-key />  
</organizational-contact>
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

organizational-contact: contact information of a particular organization and/or department.

b) Data Element Keys, Multi-keys, and Key Lists

contact-key, *department-key*, and *organization-key* are used in this element.

c) Common Data Elements

No common data elements are used in this element.

2.3.15 Parameters

parameters specifies a set of variables that appearing in a mathematical expression. The data associated with parameters are *type*, *identifier*, *number*, *name*, *value*, and *default*. *value* is a given value for the parameter, and *default* is a pre-assigned value for the parameter.

The UML model for the *parameters* element is shown below:



The XML structure for the *parameters* element is shown below:

```

<parameters>
  <parameter type="" identifier="" number="">
    <name />
    <value />
    <default />
  </parameter>
</parameters>
  
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

default: a pre-assigned value that is used when the parameter value is not specified.

parameters: a collection of individual parameters.

b) Data Element Keys and Multi-keys

No keys are used in this element.

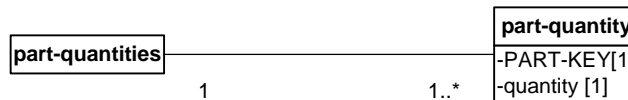
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *identifier*, *name*, *number*, *type*, and *value*.

2.3.16 Part Quantities

part-quantities specifies the type and quantity of each part. It is represented by a set of *part-quantity* data elements.

The UML model for the *part-quantities* element is shown below:



The XML structure for the *part-quantities* element is shown below:

```
<part-quantities>
  <part-quantity>
    <part-key />
    <quantity />
  </part-quantity>
</part-quantities>
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

part-quantity: the type and quantity number of a particular part. It is represented by the following data elements: *part-key*, and *quantity*.

b) Data Element Keys, Multi-keys, and Key Lists

part-key is used in this element.

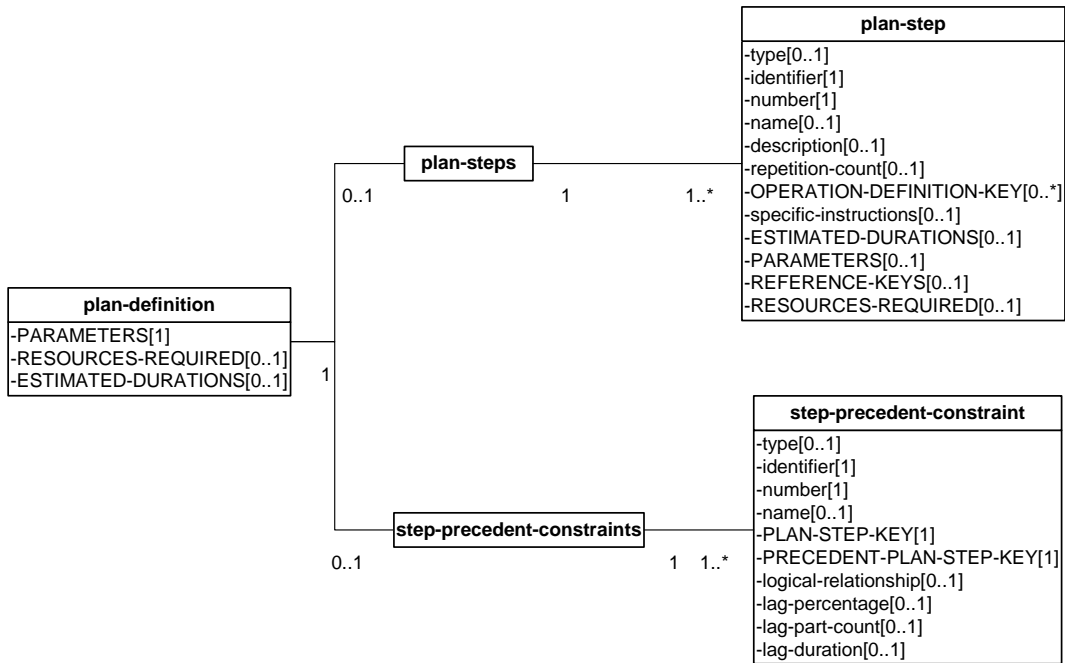
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *quantity*.

2.3.17 Plan Definition

plan-definition specifies information about a particular operation defined within an operation sheet or a routing sheet that supports a particular process plan. The information contains all required resources, operation steps, and each operation step's precedent constraints. It is represented by the following data elements: *parameters*, *resources-required*, *plan-steps*, and *step-precedent-constraints*.

The UML model for the *plan-definition* element is shown below:



The XML structure for the *plan-definition* element is shown below:

```

<plan-definition>
  <parameters />
  <resources-required />
  <plan-steps>
    <plan-step type="" identifier="" number="">
      <name />
      <description />
      <repetition-count />
      <operation-definition-key />
      <specific-instructions />
      <estimated-durations />
      <parameters />
      <reference-keys />
      <resources-required />
    </plan-step>
  </plan-steps>
  <step-precedent-constraints>
    <step-precedent-constraint type="" identifier="" number="">
      <name />
      <plan-step-key />
      <precedent-plan-step-key />
      <logical-relationship />
      <lag-percentage />
      <lag-part-count />
      <lag-duration />
    </step-precedent-constraint>
  </step-precedent-constraints>
  <estimated-durations />
</plan-definition>
  
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

lag-duration: the amount of time that must elapse after a constraining work-item starts (SS or SF) or finishes (FS or FF) and before a constrained job may start (SS or FS) or finish (SF or FF), where FS indicates the constraining work-item must finish before the constrained work-item may start, FF indicates the constraining work-item must finish before the constrained work-item may finish, SS indicates the constraining work-item must start before the constrained work-item may start, and SF indicates the constraining work-item must start before the constrained work-item may finish.

lag-part-count: the number of parts that must be completed after a constraining work-item starts (SS or SF) or finishes (FS or FF) and before a constrained job may start (SS or FS) or finish (SF or FF).

lag-percentage: the percentage of the estimated work-item completion time that must elapse after a constraining work-item starts (SS or SF) or finishes (FS or FF) and before a constrained job may start (SS or FS) or finish (SF or FF).

logical-relationship: the relationship between when a constraining work-item starts or finishes (always stated first) and when a constrained work-item starts or finishes (always stated last). The relationship is stated as one of “FS”, “FF”, “SS”, or “SF”.

plan-definition: information about a particular operation defined within an operation sheet or a routing sheet that supports a particular process plan.

plan-step: information about an operation step that supports a manufacturing operation defined in the process plan. The information includes the identification, time requirements, resource requirements, and references about the operation step. It is represented by the following data elements: *type*, *identifier*, *number*, *name*, *description*, *repetition-count*, *operation-definition-key*, *specific-instructions*, *plan-durations*, *parameters*, *reference-keys*, *step-durations*, and *resources-required*.

plan-steps: a collection of operation steps. It is represented by a set of *plan-step* data elements.

specific-instructions: statements related to a particular operation.

step-precedent-constraint: a set of precedent constraints applied to a particular operation step specified in the process plan. It is represented by the following data elements: *type*, *identifier*, *number*, *name*, *plan-step-key*, *precedent-plan-step-key*, *logical-relationship*, *lag-percentage*, *lag-part-count* and *lag-duration*.

step-precedent-constraints: a collection of precedent constraints for a particular manufacturing operation. It is represented by a set of *step-precedent-constraint* data elements.

b) Data Element Keys, Multi-keys, and Key Lists

operation-definition-key, *plan-step-key*, *reference-keys*, and *precedent-plan-step-key* are used in this element.

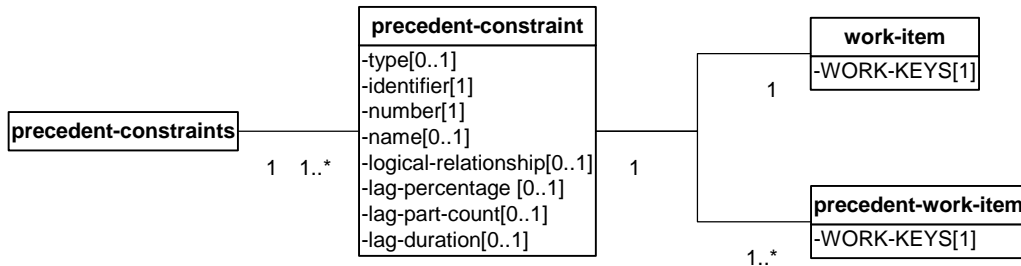
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *description*, *estimated-durations*, *identifier*, *logical-relationship*, *name*, *number*, *parameters*, *plan-durations*, *repetition-count*, *resources-required*, and *type*.

2.3.18 Precedent Constraints

precedent-constraints describes all precedent constraints for a particular work item and its children work items. A precedent constraint is a prerequisite for starting the performance of the work item, such prerequisite is based on the completion of a particular work item or a set of work items. A precedent constraint may be about procurement order, production item, pick up order, or maintenance order. It is represented by a set of *precedent-constraint* data elements.

The UML model for the *precedent-constraints* element is shown below:



The XML structure for the *precedent-constraints* element is shown below:

```

<precedent-constraints>
  <precedent-constraint type="" identifier="" number="">
    <name />
    <logical-relationship />
    <lag-percentage />
    <lag-part-count />
    <lag-duration />
    <work-item>
      <work-keys />
    </work-item>
    <precedent-work-item>
      <work-keys />
    </precedent-work-item>
  </precedent-constraint>
</precedent-constraints>
  
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

lag-duration: (refer to Section 2.3.17).

lag-part-count: (refer to Section 2.3.17).

lag-percentage: (refer to Section 2.3.17).

logical-relationship: (refer to Section 2.3.17).

precedent-constraint: a precedent constraint is a prerequisite for starting the performance of a particular work item, such prerequisite is based on the completion of a particular work item or a set of work items. A precedent constraint may be about procurement order, production item, pick up order, or maintenance

order. It is represented by the following data elements: *type*, *identifier*, *number*, *name*, *logical-relationship*, *lag-percentage*, *lag-duration*, *part-lag-count*, *precedent-work-item*, and *work-item*,

precedent-constraints: a collection of individual precedent constraints.

precedent-work-item: a collection of precedent work items to a particular work item.

work-item: a particular work item with whom the precedent work items are associated.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

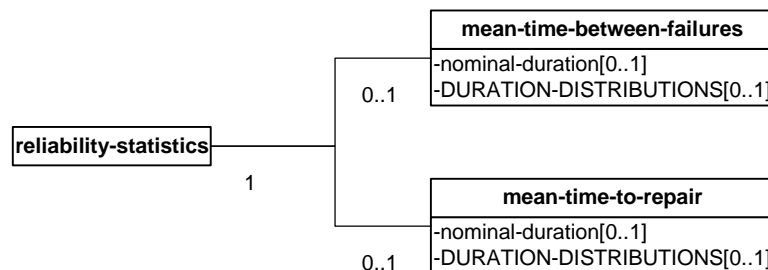
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *identifier*, *logical-relationship*, *name*, *number*, *type*, and *work-keys*.

2.3.19 Reliability Statistics

reliability-statistics specifies statistics data about a particular machine's mean time between failures and mean time to repair. It is represented by the following data elements: *mean-time-between-failure* and *mean-time-to-repair*.

The UML model for the *reliability-statistics* element is shown below:



The XML structure for the *reliability-statistics* element is shown below:

```

<reliability-statistics>
  <mean-time-between-failures>
    <nominal-duration />
    <duration-distributions />
  </mean-time-between-failures>
  <mean-time-to-repair>
    <nominal-duration />
    <duration-distributions />
  </mean-time-to-repair>
</reliability-statistics>
  
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

mean-time-between-failure: an average statistic value about time duration between two failures of a particular machine.

mean-time-to-repair: an average statistic value about time used to repair a particular machine.

reliability-statistics: statistics data about a particular machine's mean time between failures and mean time between repair.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

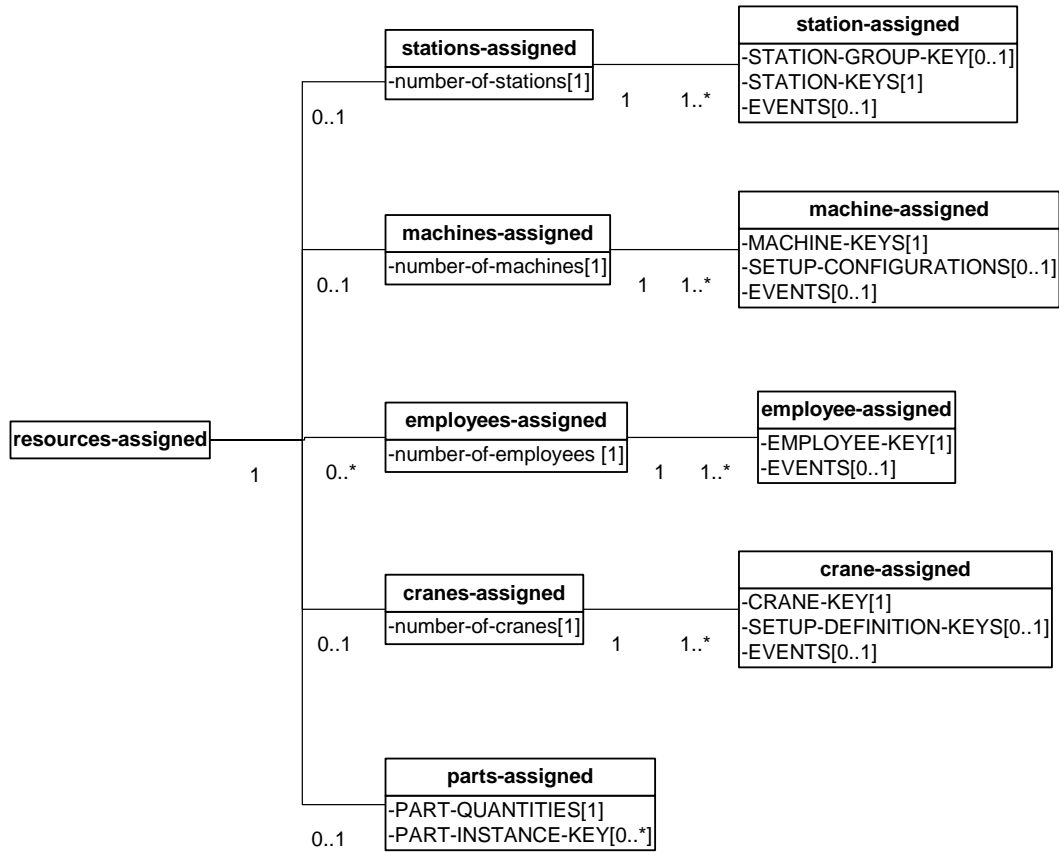
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *duration-distributions and nominal-duration*.

2.3.20 Resources Assigned

resources-assigned specifies the resources that assigned to a particular work item. The resources may include stations, machines, employees, cranes, and parts. It is represented by the following data elements: *stations-assigned, machines-assigned, employees-assigned, cranes-assigned, and parts-assigned*.

The UML model for the *resources-assigned* element is shown below:



The XML structure for the *resources-assigned* element is shown below:

```

<resources-assigned>
  <stations-assigned>
    <number-of-stations />
    <station-assigned>
      <station-keys />
      <events />
    </station-assigned>
  </stations-assigned>
  <machines-assigned>
    <number-of-machines />
    <machine-assigned>
      <machine-keys />
      <setup-configurations />
      <events />
    </machine-assigned>
  </machines-assigned>
  <employees-assigned>
    <number-of-employees />
    <employee-assigned>
      <employee-key />
      <events />
    </employee-assigned>
  </employees-assigned>
  <cranes-assigned>
    <number-of-cranes />

```



```

    <crane-assigned>
      <crane-key />
      <setup-definition-keys />
      <events />
    </crane-assigned>
  </cranes-assigned>
</parts-assigned>
  <part-quantities />
  <part-instance-key />
</parts-assigned>
</resources-assigned>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

crane-assigned: the crane that is assigned to a particular work item. It is represented by the following data elements: *crane-key*, *setup-definition-keys*, and *events*.

cranes-assigned: the cranes that are assigned to a particular work item. It is represented by the following data elements: *number-of-cranes* and *crane-assigned*.

employee-assigned: the employee that is assigned to a particular work item. It is represented by the following data elements: *employee-key* and *events*.

employees-assigned: the employees that are assigned to a particular work item. It is represented by the following data elements: *number-of-employees* and *employee-assigned*.

machine-assigned: the machine that is assigned to a particular work item. It is represented by the following data elements: *machine-keys*, *setup-configurations*, and *events*.

machines-assigned: the machines that are assigned to a particular work item. It is represented by the following data elements: *number-of-machines* and *machine-assigned*.

parts-assigned: parts that are assigned to a particular work item. It is represented by the following data elements: *part-quantities* and *part-instance-key*.

resources-assigned: information about the resources that are assigned to a particular work item.

station-assigned: the station that is assigned to a particular work item. It is represented by the following data elements: *station-group-key*, *station-key*, and *events*.

stations-assigned: the stations that are assigned to a particular work item. It is represented by the following data elements: *number-of-stations* and *station-assigned*.

b) Data Element Keys, Multi-keys, and Key Lists

crane-key, *employee-key*, *machine-keys*, *part-instance-key*, *setup-definition-keys*, and *station-keys* are used in this element.

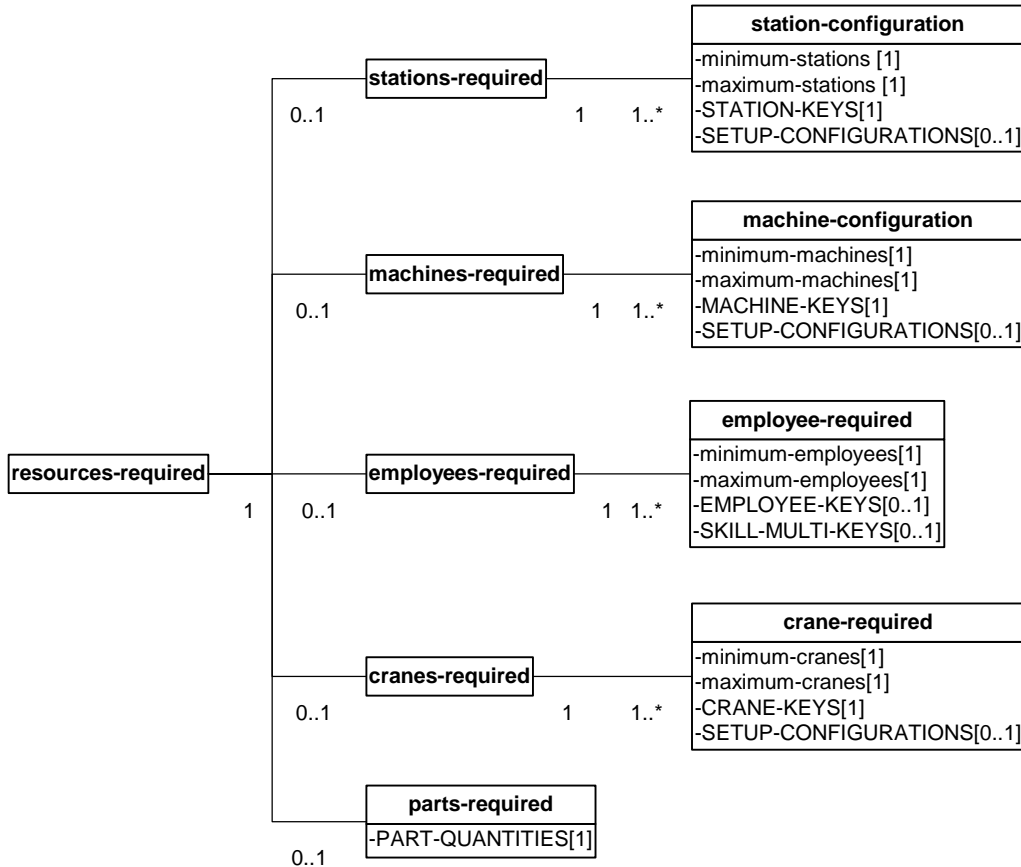
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *events*, *number-of-cranes*, *number-of-employees*, *number-of-machines*, *number-of-stations*, *part-quantities*, and *setup-configurations*.

2.3.21 Resources Required

resources-required specifies the resources required to perform a particular work item. The resources may include stations, machines, employees, cranes, and parts. It is represented by the following data elements: *stations-required*, *machines-required*, *employees-required*, *cranes-required*, and *parts-required*.

The UML model for the *resources-required* element is shown below:



The XML structure for the *resources-required* element is shown below:

```

<resources-required>
  <stations-required>
    <station-configuration>
      <minimum-stations />
      <maximum-stations />
      <station-keys />
      <setup-configurations />
    </station-configuration>
  </stations-required>
  <machines-required>
    <machine-configuration>
      <minimum-machines />
      <maximum-machines />
      <machine-keys />
      <setup-configurations />
    </machine-configuration>
  </machines-required>
  <employees-required>
    <employee-required>
      <minimum-employees />
      <maximum-employees />
      <employee-keys />
      <skill-multi-keys />
    </employee-required>
  </employees-required>
  <cranes-required>
    <crane-required>
      <minimum-cranes />
      <maximum-cranes />
      <crane-keys />
      <setup-configurations />
    </crane-required>
  </cranes-required>
  <parts-required>
    <part-quantities />
  </parts-required>
</resources-required>

```

```

    </machine-configuration>
  </machines-required>
  <employees-required>
    <employee-required>
      <minimum-employees />
      <maximum-employees />
      <employee-keys />
      <skill-multi-keys />
    </employee-required>
  </employees-required>
  <cranes-required>
    <crane-required>
      <minimum-cranes />
      <maximum-cranes />
      <crane-keys />
      <setup-configurations />
    </crane-required>
  </cranes-required>
  <parts-required>
    <part-quantities />
  </parts-required>
</resources-required>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

crane-required: the crane that is required to a particular work item. It is represented by the following data elements: *crane-key* and *setup-definition-keys*.

cranes-required: the cranes that are required to a particular work item. It is represented by a set of *crane-required* data elements.

employee-required: the employee that is required to a particular work item. It is represented by the following data elements: *minimum-employees*, *maximum-employees*, *employee-keys*, and *skill-multi-keys*.

employees-required: the employees that are required to a particular work item. It is represented by a set of *employee-required* data elements.

machine-configuration: the required machines and their setups. It is represented by the following data elements: *minimum-machines*, *maximum-machines*, *machine-keys*, and *setup-configurations*.

machines-required: the machines that are required to a particular work item. It is represented by the following data elements: *minimum-machines*, *maximum-machines*, and *machine-configuration*.

parts-required: parts that are required to a particular work item. It is represented by the following data elements: *part-quantities* and *part-instance-key*.

resources-required: information about the resources that are required to perform a particular work item.

stations-required: the stations that are required to a particular work item. It is represented by the following data elements: *minimum-stations*, *maximum-stations*, *station-group-key*, and *station-key*.

station-configuration: the required stations and their setups. It is represented by the following data elements: *minimum-stations*, *maximum-stations*, *station-key*, and *setup-configurations*.

b) Data Element Keys, Multi-keys, and Key Lists

crane-keys, *employee-keys*, *machine-keys*, *skill-multi-keys*, and *station-keys* are used in this element.

c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *maximum-employees*, *maximum-machines*, *maximum-stations*, *minimum-employees*, *minimum-machines*, *minimum-station*, *part-quantities*, and *setup-configurations*.

2.3.22 Resources Used

resources-used specifies the resources used by a particular work item. The resources may include stations, machines, employees, cranes, and parts. It is represented by the following data elements: *stations-used*, *machines-used*, *employees-used*, *cranes-used*, and *parts-used*.

The UML model for the *resources-used* element is shown below:



The XML structure for the *resources-used* element is shown below:

```

<resources-used>
  <stations-used>

```

```

    <number-of-stations />
    <station-used >
      <station-keys />
      <events/>
    </station-used>
  </stations-used>
  <machines-used>
    <number-of-machines />
    <machine-used>
      <machine-keys />
      <setup-configurations />
      <events/>
    </machine-used>
  </machines-used>
  <employees-used>
    <number-of-employees />
    <employee-used >
      <employee-key />
      <events/>
    </employee-used>
  </employees-used>
  <cranes-used>
    <number-of-cranes />
    <crane-used>
      <crane-key />
      <setup-definition-keys />
      <events/>
    </crane-used>
  </cranes-used>
  <parts-used>
    <part-quantities />
    <part-instance-key />
  </parts-used>
</resources-used>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

crane-used: the crane that is used to a particular work item. It is represented by the following data elements: *crane-key*, *setup-definition-keys*, and *events*.

cranes-used: the cranes that are used to a particular work item. It is represented by the following data elements: *number-of-cranes* and *crane-used*.

employee-used: the employee that is used to a particular work item. It is represented by the following data elements: *number-of-employees*, *employee-key*, and *events*.

employees-used: the employees that are used to a particular work item. It is represented by the following data elements: *number-of-employees* and *employee-used*.

machine-used: the machine that is used to a particular work item. It is represented by the following data elements: *machine-keys*, *setup-configurations*, and *events*.

machines-used: the machines that are used to a particular work item. It is represented by the following data elements: *number-of-machines* and *machine-used*.

parts-used: parts that are used to a particular work item. It is represented by the following data elements: *part-quantities* and *part-instance-key*.

resources-used: information about the resources that are used by a particular work item.

station-used: the station that is used to a particular work item. It is represented by the following data elements: *station-keys*, and *events*.

stations-used: the stations that is used to a particular work item. It is represented by the following data elements: *number-of-stations* and *station-used*.

b) Data Element Keys, Multi-keys, and Key Lists

crane-key, *employee-key*, *machine-keys*, *machine-group-key*, *part-instance-key*, *setup-definition-keys*, *station-group-key*, and *station-keys* are used in this element.

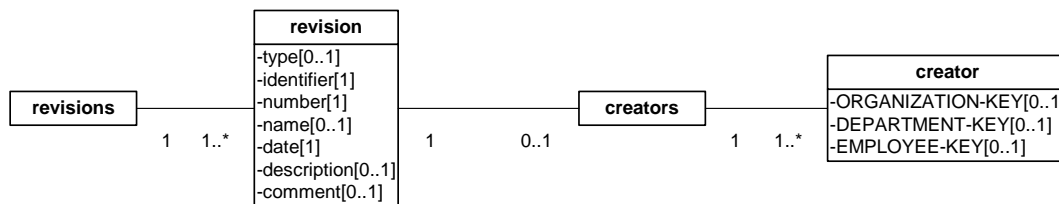
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *events*, *number-of-cranes*, *number-of-employees*, *number-of-machines*, *number-of-stations*, *part-quantities*, and *setup-configurations*.

2.3.23 Revisions

revisions specifies information about a set of revisions of the subjects. Information included in the element are each revision's identification, description, date, creators, etc.

The UML model for the *revisions* element is shown below:



The XML structure for the *revisions* element is shown below:

```

<revisions>
  <revision type="" identifier="" number="">
    <name>
    <date />
    <description />
    <comment />
    <creators>
      <creator>
        <organization-key />
        <department-key />
        <employee-key />
      </creator>
    </creators>
  </revision>
</revisions>
  
```

```
</revision>
</revisions>
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

creator: the person/system who drafts the version. It is represented by the following data elements: *organization-key*, *department-key*, and *employee-key*.

creators: the persons/systems who draft the version. It is represented by a set of *creator* data elements.

revision: information about a particular subject's revision. It is represented by the following data elements: *type*, *identifier*, *number*, *name*, *date*, *description*, *comment*, and *creators*.

revisions: a collection of individual revisions.

b) Data Element Keys, Multi-keys, and Key Lists

department-key, *employee-key*, and *organization-key* are used in this element.

c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *comment*, *date*, *description*, *identifier*, *name*, *number*, and *type*.

2.3.24 Setup Components

setup-components specifies the components items that are required to support a particular machine setup. The components items may include particular fixture(s), fixtureset(s), too(s), and toolset(s). It is represented by the following data elements: *fixture-definition-key*, *fixtureset-definition-key*, *tool-definition-key*, *toolset-definition-key*, and *logical-structure*.

The UML model for the *setup-components* element is shown below:

setup-components
-FIXTURE-DEFINITION-KEY[0..*]
-FIXTURESET-DEFINITION-KEY[0..*]
-TOOL-DEFINITION-KEY[0..*]
-TOOLSET-DEFINITION-KEY[0..*]

The XML structure for the *setup-components* element is shown below:

```
<setup-components>
  <fixture-definition-key />
  <fixtureset-definition-key />
  <tool-definition-key />
  <toolset-definition-key />
</setup-components>
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

setup-components: information about the components items that required to support a particular machine setup.

b) Data Element Keys, Multi-keys, and Key Lists

fixture-definition-key, *fixtureset-definition-key*, *tool-definition-key*, and *toolset-definition-key* are used in this element.

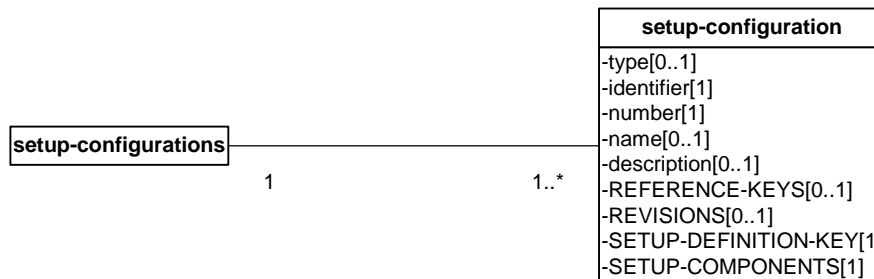
c) Common Data Elements

None.

2.3.25 Setup Configurations

setup-configurations specifies a collection of configurations that are used to support particular machines' setups. It identifies each machine setup's source definition and the tools/fixtures/toolsets/fixturesets that are required for this machine setup. It is represented by a set of *setup-configuration* data elements.

The UML model for the *setup-configurations* element is shown below:



The XML element for the *setup-configurations* element is shown below:

```
<setup-configurations>
  <setup-configuration type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <setup-definition-key />
    <setup-components />
  </setup-configuration>
</setup-configurations>
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

setup-configuration: a configuration about a particular machine setup. The configuration describes the type of setup and changeover, and the component items required to do the setup. *setup-configuration* is represented by the following data elements: *type*, *identifier*, *number*, *name*, *description*, *reference-keys*, *revisions*, *setup-definition-key*, and *setup-components*.

setup-configurations: a collection of individual setup configurations.

b) Data Element Keys, Multi-keys, and Key Lists

reference-keys and *setup-definition-key* are used in this element.

c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *description*, *identifier*, *name*, *number*, *revisions*, *setup-components*, and *type*.

2.3.26 Status Code

status-code specifies the state of a particular action/function. It is represented by the following data elements: *type*, *identifier*, *number*, *name*, *value*, and *default*.

The UML model for the *status-code* element is shown below:

status-code
-type[0..1]
-identifier[1]
-number[1]
-name[0..1]
-value[0..1]
-default[0..1]

The XML structure for the *status-code* element is shown below:

```
<status-code type="" identifier="" number="">  
  <name />  
  <value />  
  <default />  
</status-code>
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

default: a pre-assigned value that is used when the value of the status code is not specified.

status-code: information about the state of a particular action/function.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

c) Common Data Elements

See either Section 2.2 for commonly-used basic data definitions or the list at the beginning of Section 0 for the sections that define these common data attribute(s) and/or element(s): *identifier*, *name*, *number*, *type*, and *value*.

2.3.27 Stock Level Quantities

stock-level-quantities specifies the quantities of tools, fixtures, materials, or parts for various purposes. It is represented by the following data elements: *allocated-quantity*, *back-order-quantity*, *on-hand-quantity*, *on-order-quantity*, *required-quantity*, *safety-stock-quantity*, *replenishment-volume-quantity*, and *work-in-progress-quantity*.

The UML model for the *stock-level-quantities* element is shown below:

stock-level-quantities
-on-hand-quantity[1]
-allocated-quantity[0..1]
-safety-stock-quantity[0..1]
-replenishment-volume-quantity[0..1]
-required-quantity[0..1]
-on-order-quantity[0..1]
-back-order-quantity[0..1]
-work-in-process-quantity[0..1]

The XML structure for the *stock-level-quantities* element is shown below:

```
<stock-level-quantities>
  <on-hand-quantity />
  <allocated-quantity />
  <safety-stock-quantity />
  <replenishment-volume-quantity>
  <required-quantity />
  <on-order-quantity />
  <back-order-quantity />
  <work-in-process-quantity />
</ stock-level-quantities >
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

stock-level-quantities: information about the quantities of tools, fixtures, materials, or parts for various purposes.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *allocated-quantity*, *back-order-quantity*, *on-hand-quantity*, *on-order-quantity*, *replenishment-volume-quantity*, *required-quantity*, *safety-stock-quantity*, and *work-in-progress-quantity*.

2.3.28 Suppliers

suppliers specifies contact information about an organization's suppliers. It is represented by a set of *organizational-contact* data elements.

The UML model for the *suppliers* element is shown below:

suppliers
-ORGANIZATIONAL-CONTACT[1..*]

The XML structure for the *suppliers* element is shown below:

```
<suppliers>
  <organizational-contact />
</supplier>
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

suppliers: contact information about an organization’s suppliers.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *organizational-contact*.

2.3.29 Units of Measurement

units-of-measurement describes various measurement units used in the file, for example, the measurement may be for distance, speed, mass, time duration, or currency. The data associated with *units-of-measurements* are the following: *type*, *identifier*, *number*, *name*, *description*, *reference-keys*, *angular-units*, *currency-units*, *distance-units*, *length-units*, *mass-units*, *speed-units*, *time-duration-units*, and *volume-units*.

The UML model for the *units-of-measurement* element is shown below:

units-of-measurement
-type[0..1]
-identifier[1]
-number[1]
-name[0..1]
-description[0..1]
-REFERENCE-KEYS[0..1]
-angular-units[0..1]
-currency-units[0..1]
-distance-units[0..1]
-length-units[0..1]
-mass-units[0..1]
-speed-units[0..1]
-time-duration-units[0..1]
-volume-units[0..1]

The XML structure for the *units-of-measurement* element is shown below:

```

<units-of-measurement type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <angular-units />
  <currency-units />
  <distance-units />
  <length-units />
  <mass-units />
  <speed-units />
  <time-duration-units />
  <volume-units />
</units-of-measurement>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

units-of-measurement: information about various measurements units used in the file.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

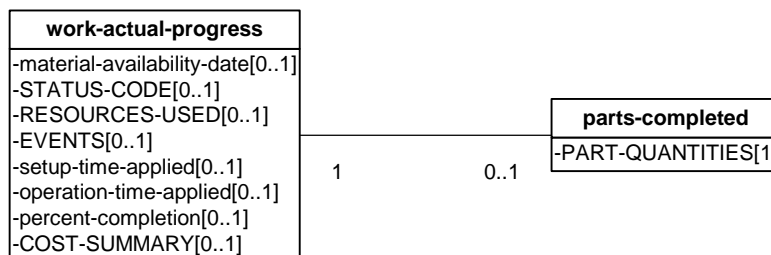
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *angular-units*, *currency-units*, *description*, *distance-units*, *identifier*, *length-units*, *mass-units*, *name*, *number*, *reference-keys*, *speed-units*, *time-duration-units*, *type*, and *volume-units*.

2.3.30 Work Actual Progress

work-actual-progress specifies a particular work item's actual working progress, including the date of material availability, a code of work status, all resources used, all events took placed, total setup time used, total operation time used, work completion percentage, number of part(s) completed, and a cost summary. It is represented by the following data elements: *material-availability-date*, *status-code*, *resources-used*, *events*, *setup-time-applied*, *operation-time-applied*, *percent-completion*, *parts-completed*, and *cost-summary*.

The UML model for the *work-actual-progress* element is shown below:



The XML structure for the *work-actual-progress* element is shown below:

```
<work-actual-progress>
```

```

    <material-availability-date />
    <status-code />
    <resources-used />
    <events / >
    <setup-time-applied />
    <operation-time-applied />
    <percent-completion />
    <parts-completed>
      <part-quantities />
    </parts-completed>
    <cost-summary / >
  </work-actual-progress>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

operation-time-applied: time spent on performing operation.

parts-completed: the part type and quantity of each part that resulted from the work item. It is represented by a collection of *part-quantities* data elements.

percent-completion: a completion percentage of the part(s).

setup-time-applied: time spent on setting up machine(s).

work-actual-progress: information about a particular work item's actual working progress.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

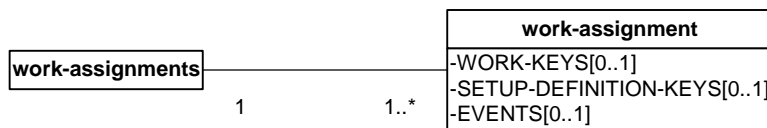
c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *cost-summary*, *events*, *material-availability-date*, *part-quantity*, *resources-used*, and *status-code*.

2.3.31 Work Assignments

work-assignments specifies information about a particular resource's work assignments. It includes work items, required setup, and all events resulted from each assignment. It is represented by a set of *work-assignment* data elements.

The UML model for the *work-assignments* element is shown below:



The XML structure for the *work-assignments* element is shown below:

```

<work-assignments>
  <work-assignment>
    <work-keys />
    <setup-definition-keys />
    <events />
  </work-assignment>
</work-assignments>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

work-assignment: information about a particular resource's work assignment. It is represented by the following data elements: *work-keys*, *setup-definition-keys*, and *events*.

work-assignments: a collection of individual work assignments.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

c) Common Data Elements

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *events*.

2.3.32 Work Scheduled Progress

work-scheduled-progress specifies a particular work item's scheduled progress, including the date of material availability, a code of work status, all resources assigned, and all events took place. It is represented by the following data elements: *material-availability-date*, *status-code*, and *events*.

The UML model for the *work-scheduled-progress* element is shown below:

work-scheduled-progress
-material-availability-date[0..1]
-STATUS-CODE[0..1]
-RESOURCES-ASSIGNED[0..1]
-EVENTS[0..1]
-COST-SUMMARY[0..1]

The XML structure for the *work-scheduled-progress* element is shown below:

```

<work-scheduled-progress>
  <material-availability-date />
  <status-code />
  <resources-assigned />
  <events />
  <cost-summary />
</work-scheduled-progress>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

work-scheduled-progress: information about a particular work item's scheduled progress.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this element.

c) Common Data Element

See Section 2.2 for commonly-used basic data definitions that define these common data attribute(s) and/or element(s): *cost-summary*, *events*, *material-availability-date*, *resource-assigned*, and *status-code*.

3 MACHINE SHOP DATA STRUCTURE

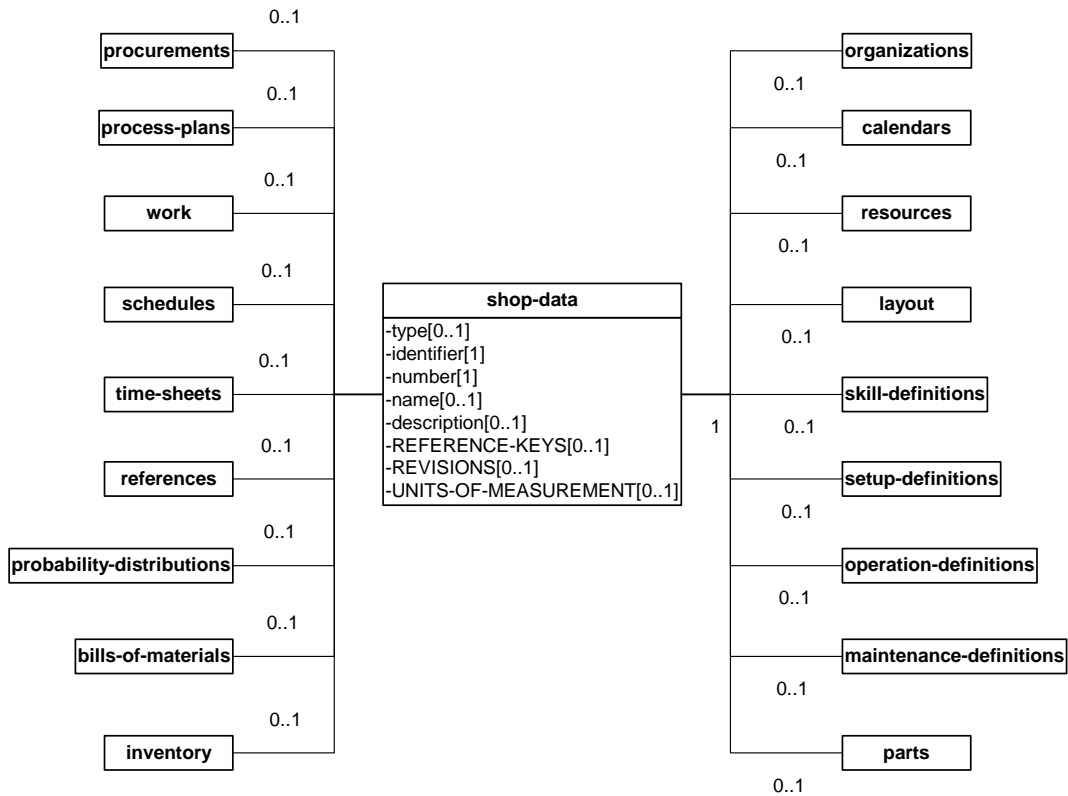
The manufacturing shop data elements are presented in the following format:

- summary description of the element
- a Unified Modeling Language (UML) information model for the element
- an eXtensible Markup Language (XML) template for coding the data of the element
- unique data elements: data elements that are uniquely defined within this particular element
- data element keys, multi-keys, and key lists: key or pointer elements that are used to reference objects defined elsewhere in the model (refer to Section 2.1).
- common data elements: data elements that are used repeatedly by other elements within the information model (refer to Section 2.2 and 2.3)

This section presents major elements of machine shop data structure. The elements are further defined in the following subsections:

- 3.1 organizations
- 3.2 departments
- 3.3 calendars
- 3.4 resources
 - 3.5 machines
 - 3.6 stations
 - 3.7 cranes
 - 3.8 employees
 - 3.9 tool-catalog
 - 3.10 fixture-catalog
- 3.11 layout
- 3.12 skill-definitions
- 3.13 setup-definitions
- 3.14 operation-definitions
- 3.15 maintenance-definitions
- 3.16 parts
- 3.17 bills-of-materials
- 3.18 inventory
 - 3.19 part-inventory
 - 3.20 materials-inventory
 - 3.21 tool-inventory
 - 3.22 fixture-inventory
- 3.23 procurements
- 3.24 process-plans
 - 3.25 routing-sheets
 - 3.26 operation-sheets
 - 3.27 machine-programs
- 3.28 work
 - 3.29 orders
 - 3.30 order-items
 - 3.31 jobs
 - 3.32 tasks
 - 3.33 maintenance-orders
 - 3.34 pick-orders
 - 3.35 tool-orders
- 3.36 schedules
- 3.37 time-sheets
- 3.38 references
- 3.39 probability-distributions

The UML information model for the overall shop-data element is shown below:



The overall XML structure for the *shop-data* element is shown below:

```

<shop-data type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <units-of-measurement />
  <organizations />
  <calendars />
  <resources />
  <layout />
  <skill-definitions />
  <setup-definitions />
  <operation-definitions />
  <maintenance-definitions />
  <parts />
  <bills-of-materials />
  <inventory />
  <procurements />
  <process-plans />
  <work />
  <schedules />

```

```

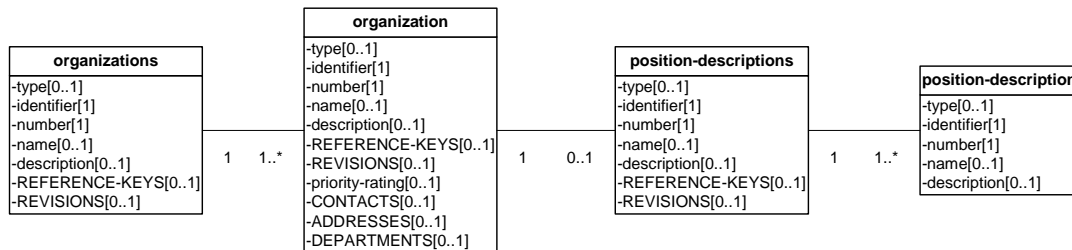
    <time-sheets />
    <references />
    <probability-distributions />
  </shop-data>

```

3.1 Organizations

organizations is used to maintain organizational structure, contacts and addresses information for the manufacturing organization, its customers and suppliers. The departmental structure of an organization may be defined using the *departments* element, see section 3.2. The *position-descriptions* element may be used to specify a *position-description* for each *position* in the *organization*. The *organizations* element is illustrated below:

The UML information model for the *organizations* element is shown below:



The XML structure for the *organizations* element is shown below:

```

<organizations type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <organization type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <priority-rating />
    <contacts />
    <addresses />
    <departments />
    <position-descriptions type="" identifier="" number="">
      <name />
      <description />
      <reference-keys />
      <revisions />
      <position-description type="" identifier="" number="">
        <name />
        <description />
      </position-description>
    </position-descriptions>
  </organization>
</organizations >

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

departments: the departments that contained within an organization, see the departments element in Section 3.2. Use of the *departments* element is optional, since the departmental structure of external organizations may not be known or relevant.

position-description: a particular job position within the organization.

position-descriptions: a list of position descriptions for the organization.

organization: the organization that is being modeled and external organizations with which it interacts, e.g., customers and suppliers.

organizations: a collection of organizational information such as points of contact and addresses for internal or external organizations, e.g., suppliers, customers.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this element: *reference-keys*. Elements within *resources*, *parts*, *procurements*, *revisions*, and *work* use the *organization-key* and *contact-key* to reference the data elements in the *organizations* element.

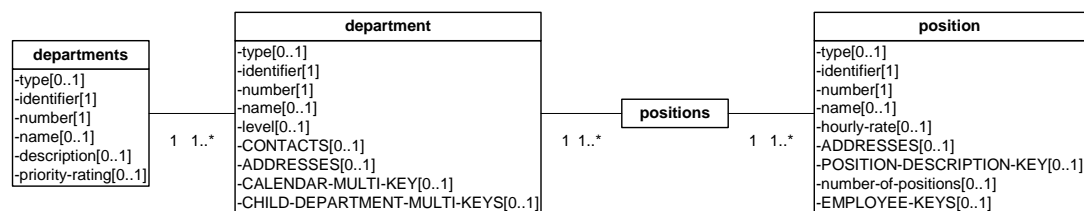
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *addresses*, *contacts*, *description*, *identifier*, *name*, *number*, *priority-rating*, *revisions*, and *type*.

3.2 Departments

departments defines the departments within the organization, the child departments of the department, the positions within the department, and the employees occupying those positions. The *position* elements within *department* is used to identify a *position-description* contained within *position-descriptions*. The *departments* element is illustrated below:

The UML information model for the *departments* element is shown below:



The XML structure for the *departments* element is shown below:

```

<departments type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <priority-rating />

```

```

<department type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <level />
  <contacts />
  <addresses />
  <calendar-multi-key />
  <child-department-multi-key />
  <positions>
    <position type="" identifier="" number="">
      <name />
      <hourly-rate />
      <addresses />
      <position-description-key />
      <number-of-positions />
      <employee-keys />
    </position>
  </positions>
</department>
</departments>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

department: an element that identifies a particular organizational element and its relationships to other organization elements, department-wide shift schedules used, the positions and employees in the department.

departments: a collection of individual departments.

level: the hierarchical level within the organization at which the department is resided. The convention is that the highest level in the organization is given the number “1”. Child departments are organized in a hierarchical tree structure and have increasingly higher level numbers. A child department does not necessarily have to be at the next sequentially higher number. For example, a level “3” department may be the child department of a level “1” department. (Note: Departmental level definitions and keys will be added in a later version of this specification.)

position: a specific departmental role held by an employee. An employee may hold multiple positions in a department, in different departments, and different organizations.

positions: a list of positions contained within a department. *positions* is defined by one or more *position* data elements.

b) Data Element Keys, Multi-keys, and Key Lists

calendar-multi-key, *child-department-multi-key*, *employee-keys*, and *position-description-key* are used in this element. *calendar-multi-key* is used to identify the effective shift schedule and non-working days associated with a particular department. *child-department-multi-keys* is used to specify a list of one or more departments that are subordinate to the specified department. The multi-key provides keys to identify both the *organization* and *department* elements for the child department. *employee-keys* is used to specify the list of one or more employee keys for the employees that occupy a position in the department. *position-*

description-key is used to specify a specific position in a department. Elements *employees* and *revisions* use a *department-key* and *position-multi-keys* to reference the data elements in the *departments*.

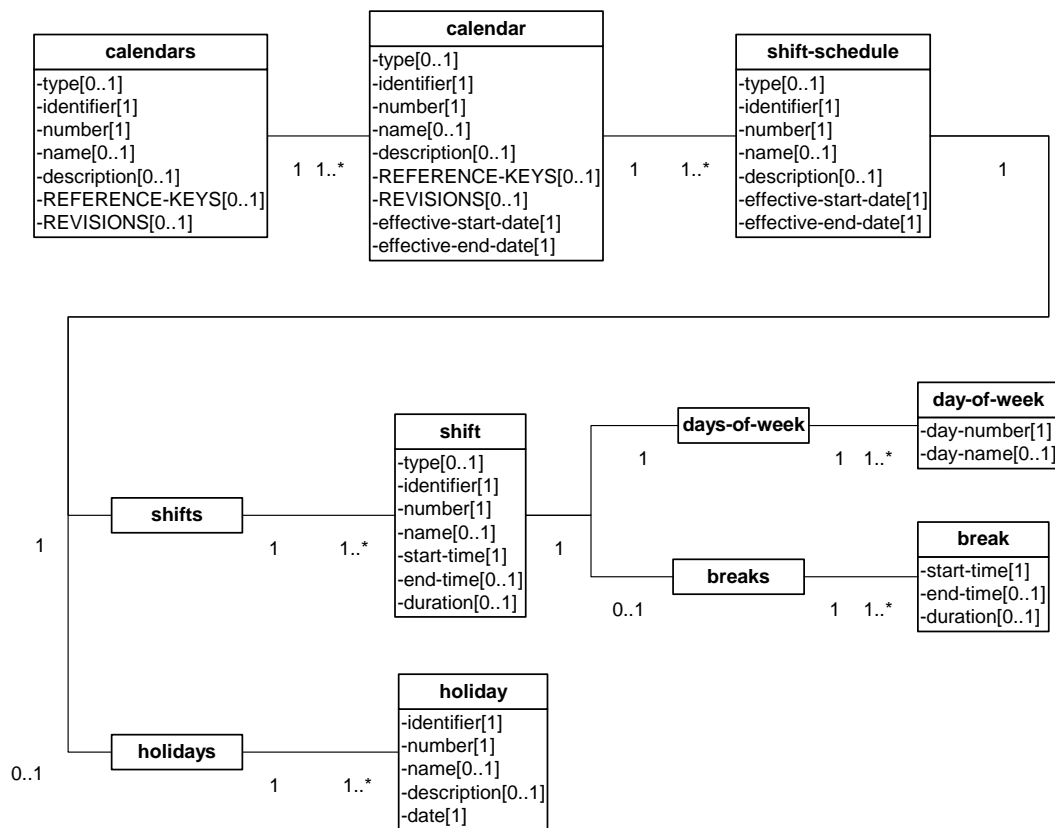
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *addresses*, *contacts*, *description*, *hourly-rate*, *identifier*, *name*, *number*, *number-of-positions*, *priority-rating*, and *type*.

3.3 Calendars

calendars identifies the shift schedules that are in effect for a period of time, breaks, holidays. The *calendars* element is illustrated below:

The UML information model for the *calendars* element is shown below:



The XML structure for the *calendars* element is shown below:

```

<calendars type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <calendar type="" identifier="" number="">
    <name />
    <description />

```

```

<reference-keys />
<revisions />
<effective-start-date />
<effective-end-date />
<shift-schedule type="" identifier="" number="">
  <name />
  <description />
  <effective-start-date />
  <effective-end-date />
  <shifts>
    <shift type="" identifier="" number="">
      <name />
      <days-of-week>
        <day-of-week>
          <day-number />
          <day-name />
        </day-of-week>
      </days-of-week>
      <start-time />
      <end-time />
      <duration />
      <breaks>
        <break>
          <start-time />
          <end-time />
          <duration />
        </break>
      </breaks>
    </shift>
  </shifts>
  <holidays>
    <holiday identifier="" number="">
      <name />
      <description />
      <date />
    </holiday>
  </holidays>
</shift-schedule>
</calendar>
</calendars>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

break: a scheduled free time, i.e., non-working time, during a particular work shift.

breaks: a collection of individual *break* definitions.

calendar: information about the shift schedules that are in effect for a period of time, breaks, and holidays.

calendars: a collection of individual *calendar* definitions.

day-name: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, or Saturday.

day-number: 0, 1, 2, 3, 4, 5, or 6, (representing Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, or Saturday.)

days-of-week: the working days within the week. *days-of-week* is defined by a set of *day-number* and *day-name* data elements.

holiday: a non-work day specified by the organization.

holidays: a collection of individual holidays.

shift: a mechanism for representing a particular work shift.

shifts: a list of one or more *shift* data elements.

shift-schedule: a mechanism for presenting all work shifts and holidays for a range of dates.

b) Data Element Keys, Multi-keys, and Key Lists

reference-keys are used in this element. Elements within *resources* and *departments* use the *calendar-multi-key* to reference the data elements in the *calendars* element.

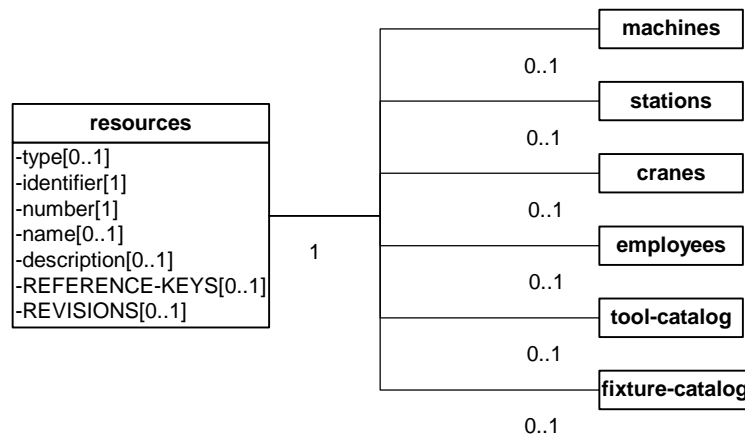
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *date*, *description*, *duration*, *effective-end-date*, *effective-start-date*, *end-time*, *identifier*, *name*, *number*, *revisions*, *start-time*, and *type*.

3.4 Resources

resources describes all the resources that may be assigned to tasks in the shop. The resource types available in the machine shop environment include: stations and machines, cranes, employees, tool and fixture catalog items.

The UML information model for the *resources* element is shown below:



The XML structure for the *resources* element is shown below:

```

<resources type="" identifier="" number="">
  <name />

```

```

    <description />
    <reference-keys />
    <revisions />
    <machines />
    <stations />
    <cranes />
    <employees />
    <tool-catalog />
    <fixture-catalog />
</resources>

```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

cranes: an element for specifying a collection of crane-type resources (refer to Section 3.7)

employees: an element for specifying a collection of employee-type resources (refer to Section 3.8)

machines: an element for specifying a collection of machine-type or machine-group-type resources (refer to Section 3.5)

fixture-catalog: an element for specifying a collection of fixture-type resources (refer to Section 3.10)

resources: a collection of shop resource records, e.g., employees, machines, and stations.

stations: an element for specifying a collection of station-type or station-group-type resources (refer to Section 3.6)

tool-catalog: an element for specifying a collection of tool-type resources (refer to Section 3.9)

b) Data Element Keys, Multi-keys, and Key Lists

reference-keys is used in this structure. Keys within *resources-assigned*, *resources-required*, and *resources-used* point at elements in this structure.

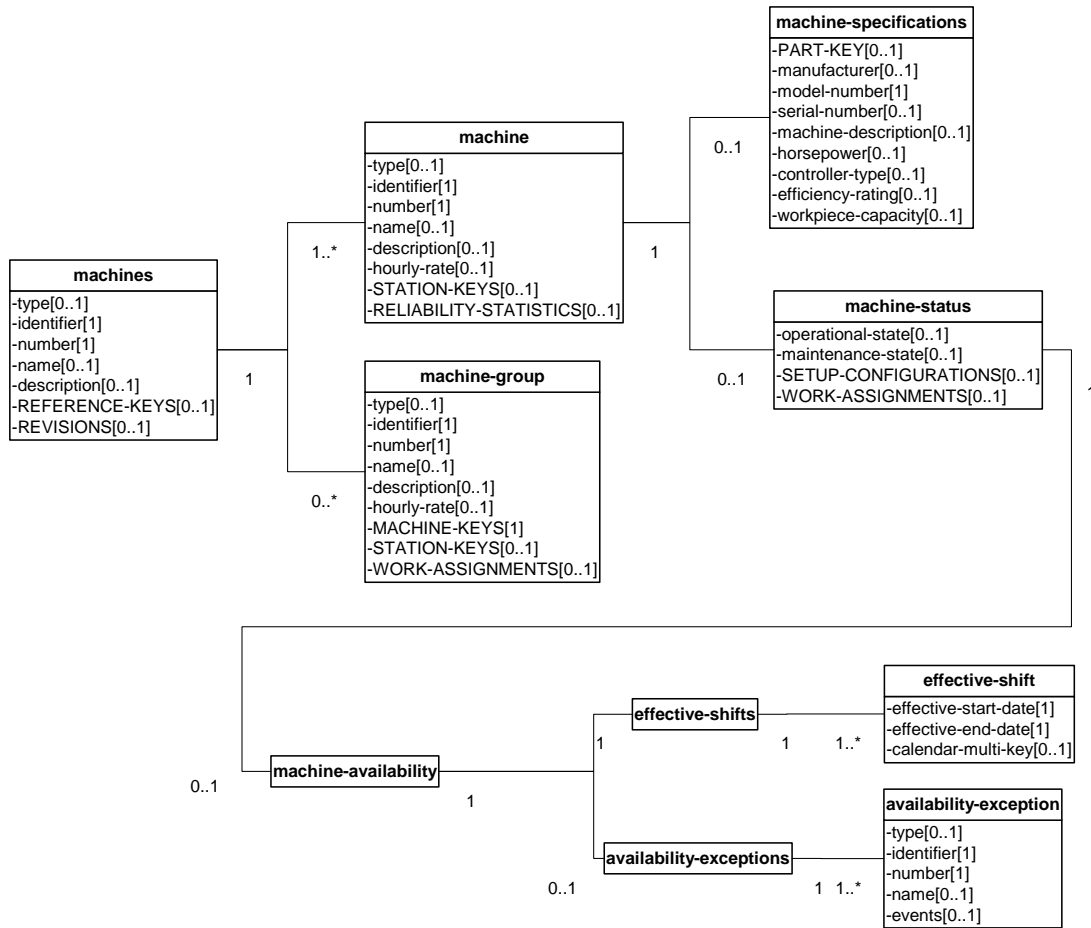
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *identifier*, *name*, *number*, *revisions*, and *type*.

3.5 Machines

machines lists each machine that exists within the shop. The *machine* element provides descriptive information, hourly rate, technical specifications, status, station associations, and reliability data for a machine.

The UML information model for the *machines* element is shown below:



The XML structure for the *machines* element is shown below:

```

<machines type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <machine type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <hourly-rate />
    <machine-specifications>
      <part-key />
      <manufacturer />
      <model-number />
      <serial-number />
      <machine-description/>
      <horsepower />
      <controller-type />
      <efficiency-rating />
      <workpiece-capacity />
    </machine-specifications>
  </machine>
  <machine-availability />
  <effective-shifts />
  <availability-exceptions />
</machines>
  
```

```

    <machine-status>
      <machine-availability>
        <effective-shifts>
          <effective-shift>
            <effective-start-date />
            <effective-end-date />
            <calendar-multi-key />
          </effective-shift>
        </effective-shifts>
        <availability-exceptions>
          <availability-exception type="" identifier="" number="">
            <name />
            <events />
          </availability-exception>
        </availability-exceptions>
      </machine-availability>
      <operational-state />
      <maintenance-state />
      <setup-configurations />
      <work-assignments />
    </machine-status>
    <station-keys />
    <reliability-statistics />
  </machine>
  <machine-group type="" identifier="" number="">
    <name />
    <description />
    <hourly-rate />
    <machine-keys />
    <station-keys />
    <work-assignments />
  </machine-group>
</machines>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

controller-type: the type of controller on the machine.

efficiency-rating: the relative efficiency of a machine as compared to other machines capable of performing similar processes. The nominal efficiency rating is 1.0. A machine with an efficiency rating of 2.0 would be %100 more efficient. Presumably the machine with the rating 2.0 would perform the same process in half the time of a machine with an efficiency rating of 1.0.

horsepower: the horsepower rating of the machine.

machine: information about a machine.

machines: a collection of individual machines.

machine-group: a group of machines that have similar characteristics for scheduling purposes.

machine-specifications: information about the machine's specifications. It is represented by the following data elements: *manufacturer*, *model-number*, *serial-number*, *description*, *horsepower*, *controller-type*, and *efficiency-rating*.

machine-status: the operational state, maintenance state, current setup, and work assignments associated with a machine.

manufacturer: the name of the manufacturer of the machine.

workpiece-capacity: the maximum number of workpieces that can be processed by the machine at the same time.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this element: *machine-keys*, *part-key*, *reference-keys*, and *station-keys*. *machine-keys* identifies a set of similar machines that are contained within a machine group. *station-keys* identifies one or more stations where employees may be assigned that are associated with a machine. Elements within *machine-programs*, *resources-assigned*, *resources-required*, *resources-used*, *schedules*, and *stations* use the *machine-key* to reference the data elements in the *machines* element.

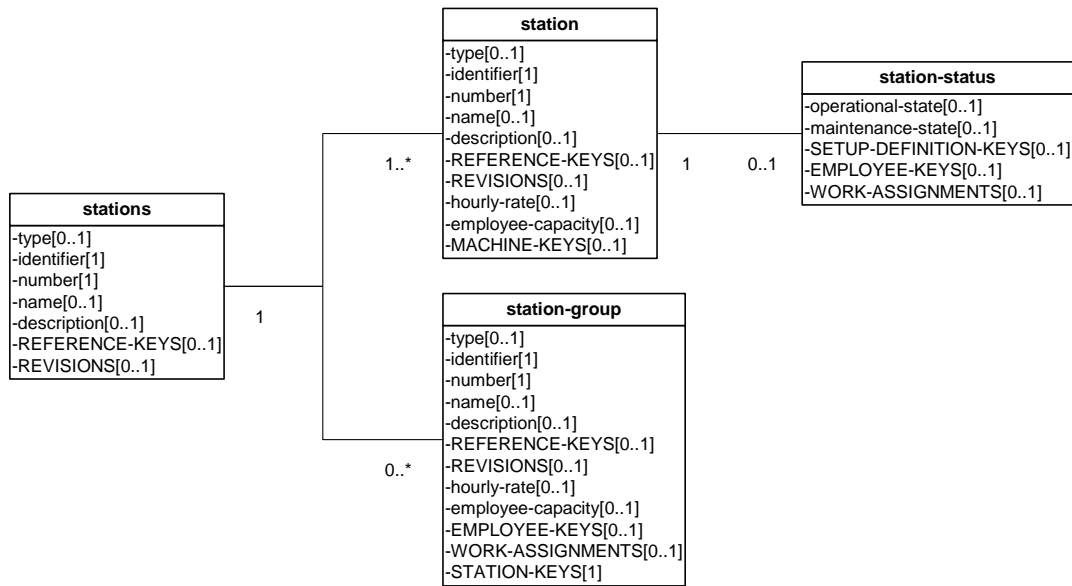
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *hourly-rate*, *identifier*, *maintenance-state*, *model-number*, *name*, *number*, *operational-state*, *reliability-statistics*, *revisions*, *serial-number*, *setup-configurations*, *type*, and *work-assignments*.

3.6 Stations

stations specifies all the places where employees may perform work in the shop. The element provides elements for indicating station status, associated machines, assigned employees and work assignments. *station-group* specifies a set of stations that have the same characteristics. *station-keys* are used to identify the stations that make up the group.

The UML information model for the *stations* element is shown below:



The XML structure for the *stations* element is shown below:

```

<stations type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <station type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <hourly-rate />
    <employee-capacity />
    <station-status>
      <operational-state />
      <maintenance-state />
      <setup-definition-keys />
      <employee-keys />
      <work-assignments />
    </station-status>
    <machine-keys />
  </station>
  <station-group type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <hourly-rate />
    <employee-capacity />
    <employee-keys />
    <work-assignments />
    <station-keys />
  </station-group>
</stations>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

employee-capacity: the maximum number of employees that can be assigned to this station at the same time.

station: information about a station, its status, machines, and work assignments.

station-group: a set of stations that have similar characteristics.

stations: a collection of individual stations.

station-status: information about the operational state, maintenance state, current setup, and work assignments of a station. Note that a machine associated with a station will normally identify the fixture and tool setups. Some stations, e.g., manual welding stations, may not be associated with any machine, hence the *setup-definition-keys* should be included at this level, if appropriate.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this element: *employee-keys*, *machine-keys*, *reference-keys*, and *setup-definition-keys*. Elements within cranes, machines, operation-definitions, *resources-assigned*, *resources-required*, *resources-used*, and *schedules* use the *station-group-key* and *station-key* to reference the data elements in the *stations* element.

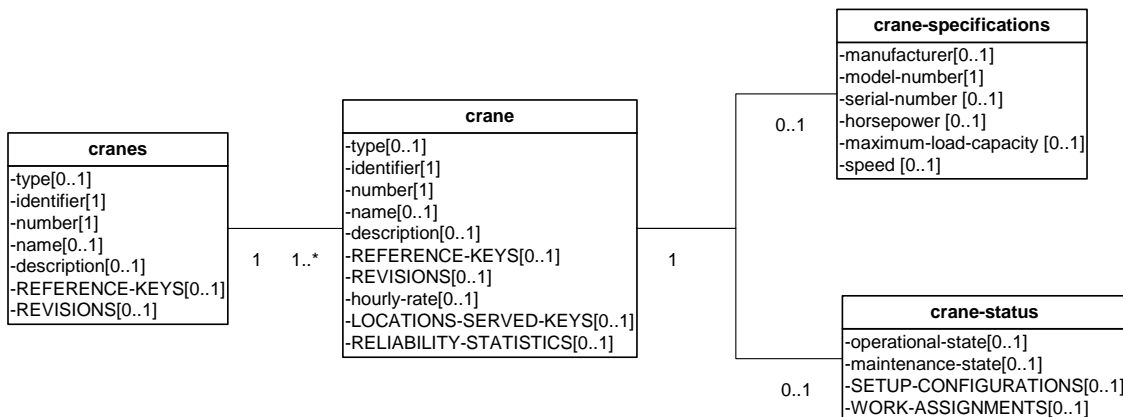
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *hourly-rate*, *identifier*, *operational-state*, *maintenance-state*, *name*, *number*, *revisions*, *type*, and *work-assignments*.

3.7 Cranes

cranes specifies all the cranes that may be assigned to work in the shop, their status, the stations served, machines served, and work assignments.

The UML information model for the *cranes* element is shown below:



The XML structure for the *cranes* element is shown below:

```
<cranes type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <crane type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <hourly-rate />
    <crane-specifications>
      <manufacturer />
      <model-number />
      <serial-number />
      <horsepower />
      <maximum-load-capacity />
      <speed />
    </crane-specifications>
    <locations-served-keys />
    <crane-status>
      <operational-state />
      <maintenance-state />
      <setup-configurations />
      <work-assignments />
    </crane-status>
    <reliability-statistics />
  </crane>
</cranes>
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

crane: information about a crane.

cranes: a collection of individual cranes.

crane-specifications: information about the crane's technical specifications including the crane's manufacturer, model-number, serial-number, horsepower, maximum-load-capacity, and speed.

crane-status: information about the status of operation and maintenance, current setup, and work assignments of the crane.

horsepower: the horsepower rating of the crane

manufacturer: the name of the manufacturer of the machine

maximum-load-capacity: maximum carrying capacity of the crane in specified weight units.

speed: the speed at which a crane can move from one location served to the next.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this element: *locations-served-keys* and *reference-keys*. *locations-served-keys* identifies the stations, machines, and storage buffers that are accessible to the crane. Elements within *resources-assigned*, *resources-required*, *resources-used*, and *schedules* use the *crane-key* to reference the data elements in the *cranes* element.

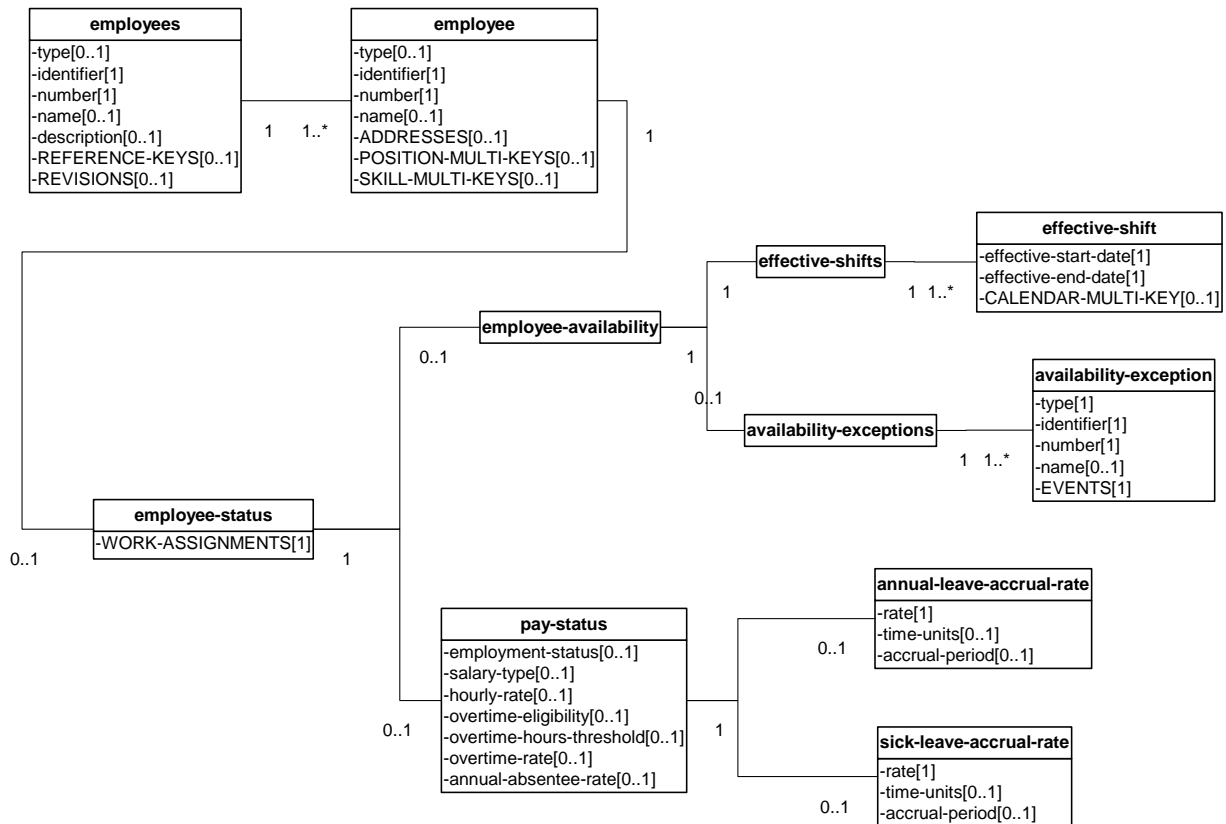
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *hourly-rate*, *identifier*, *maintenance-state*, *model-number*, *name*, *number*, *operational-state*, *reliability-statistics*, *revisions*, *serial-number*, *setup-configurations*, *type*, and *work-assignments*.

3.8 Employees

employees specifies all the employees that may be assigned work in the shop, their status, and scheduled assignments to work items. Each employee's addresses, positions in the organization, skills, shift assignments, availability exceptions, pay and leave information, and work assignments are also defined.

The UML information model for the *employees* element is shown below:



The XML structure for the *employees* element is shown below:

```
<employees type="" identifier="" number="">
```

```

<name />
<description />
<reference-keys />
<revisions />
<employee type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <addresses />
  <employment-status />
  <position-multi-keys />
  <skill-multi-keys />
  <employee-status>
    <employee-availability>
      <effective-shifts>
        <effective-shift>
          <effective-start-date />
          <effective-end-date />
          <calendar-multi-key />
        </effective-shift>
      </effective-shifts>
      <availability-exceptions>
        <availability-exception type="" identifier="" number="">
          <name />
          <events />
        </availability-exception>
      </availability-exceptions>
    </employee-availability>
  <pay-status>
    <employment-status />
    <salary-type />
    <hourly-rate />
    <overtime-eligibility />
    <overtime-hours-threshold />
    <overtime-rate />
    <annual-leave-accrual-rate>
      <rate />
      <time-units />
      <accrual-period />
    </annual-leave-accrual-rate>
    <sick-leave-accrual-rate>
      <rate />
      <time-units />
      <accrual-period />
    </sick-leave-accrual-rate>
    <annual-absentee-rate />
  </pay-status>
  <work-assignments />
</employee-status>
</employee>
</employees>

```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

accrual-period: a period of time that used to accrue annual leave or sick leave for an individual employee.

annual-level-accrual-rate: the number of annual leave hours accrued per pay period.

availability-exception: a period of time that the employee is unavailable to be scheduled for regular duties.

availability-exceptions: a collection of availability exceptions for an individual employee.

effective-shift: a particular shift in which the employee has duty.

effective-shifts: a collection of shifts in which the employee has duty.

employee: information about the individual working for the organization.

employee-availability: information about an employee's shift assignments, and periods of time that they will not be available, i.e., availability exceptions.

employees: a collection of employee elements.

employee-status: a collection of status elements about an employee including shift assignments, availability exceptions, pay status, and work assignments.

employment-status: a code indicating employment status, e.g., part-time, full-time, and retired.

overtime-eligibility: the eligibility status to earn overtime.

overtime-hours-threshold: the number of hours at which employee goes into over-time status.

pay-status: a collection of elements regarding an employee's pay status.

sick-leave-accrual-rate: the number of sick leave hours accrued per pay period.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this element: *calendar-multi-key*, *position-multi-keys*, *reference-keys*, and *skill-multi-keys*. Elements within *departments*, *resources-assigned*, *resources-required*, *resources-used*, *revisions*, *schedules*, *stations*, and *timesheets* use the *employee-key* and *skill-multi-keys* to reference the data elements in the *employees* element.

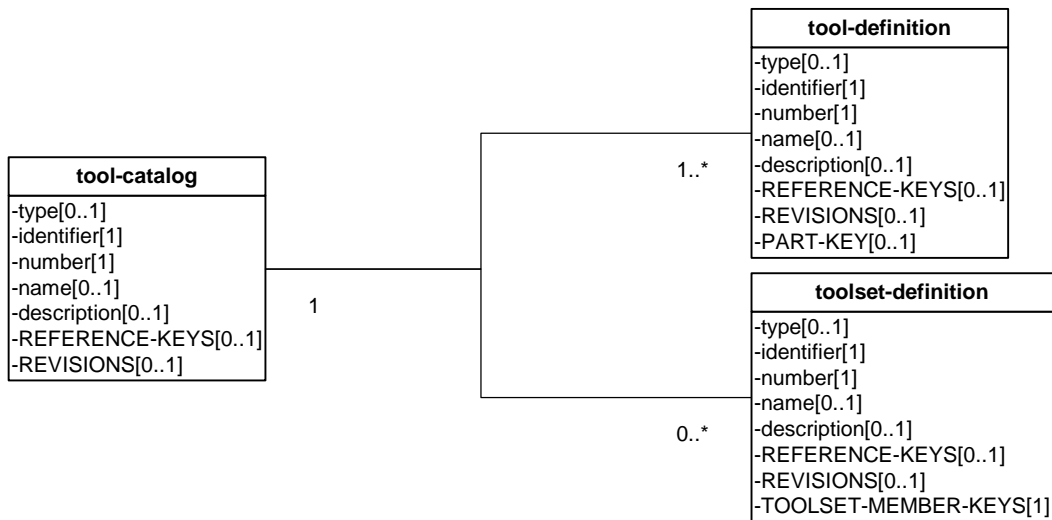
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *addresses*, *annual-absentee-rate*, *description*, *effective-end-date*, *effective-start-date*, *events*, *hourly-rate*, *identifier*, *name*, *number*, *overtime-rate*, *rate*, *revisions*, *salary-type*, *time-units*, and *type*.

3.9 Tool Catalog

tool-catalog defines all the tools and toolsets that may be assigned to tasks in the shop. Bill of materials associated with the tools, and tools that included in each toolset are also defined.

The UML information model for the *tool-catalog* element is shown below:



The XML structure for the *tool-catalog* element is shown below:

```

<tool-catalog type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <tool-definition type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <part-key />
  </tool-definition>
  <toolset-definition type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <toolset-member-keys />
  </toolset-definition>
</tool-catalog>
  
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

tool-catalog: a collection of elements that define tools that are available in the shop tool crib.

tool-definition: information about the characteristics of a particular tool.

toolset-definition: information about the characteristics of a particular tool set or collection of tools.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this element: *part-key*, *reference-keys*, and *toolset-member-keys*. Elements within *setup-components*, *tool-inventory*, and *tool-orders* use the *tool-definition-key*, and *toolset-definition-key* to reference the data elements in the *tool-catalog* element.

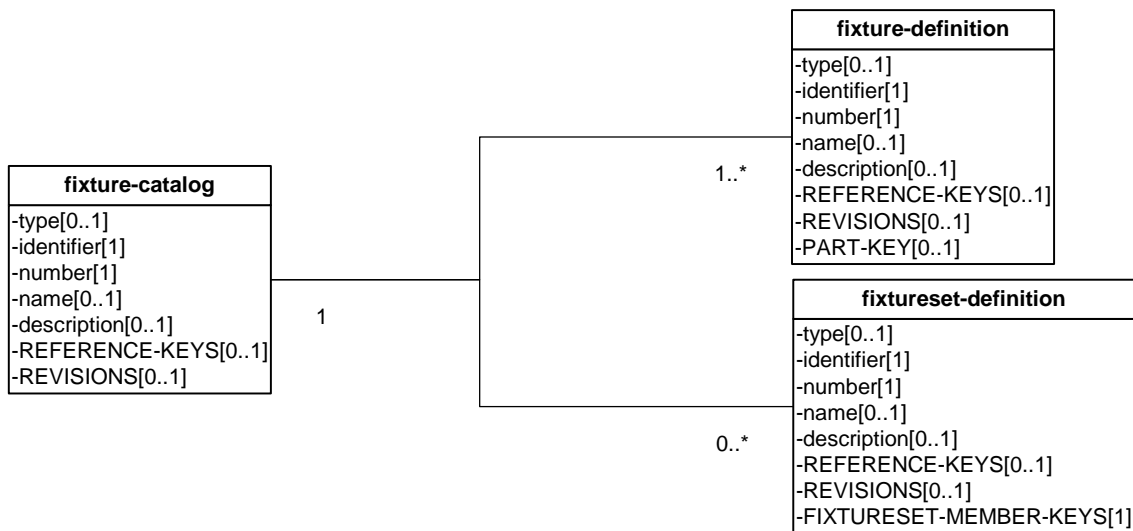
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *identifier*, *name*, *number*, *revisions*, and *type*.

3.10 Fixture Catalog

fixture-catalog defines all the fixtures and fixturesets that may be assigned to tasks in the shop. Bill of materials associated with the fixtures, and fixtures that included in each fixtureset are also defined.

The UML information model for the *fixture-catalog* element is shown below:



The XML structure for the *fixture-catalog* element is shown below:

```

<fixture-catalog type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <fixture-definition type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <part-key />
  </fixture-definition>
  <fixtureset-definition type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
  </fixtureset-definition>
</fixture-catalog>
  
```

```
        <fixtureset-member-keys />
    </fixtureset-definition>
</fixture-catalog>
```

The unique data elements, data element keys, and common data elements introduced in this element are described below.

a) Unique Data Elements

fixture-catalog: a collection of fixtures that are available in the shop's fixture crib.

fixture-definition: information about the characteristics of a particular fixture.

fixtureset-definition: information about the characteristics of a particular fixture set.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this element: *part-key*, *reference-keys*, and *fixtureset-member-keys*. Elements within *fixture-inventory* and *setup-components* use the *fixture-definition-key*, and *fixtureset-definition-key* to reference the data elements in the *fixture-catalog* element.

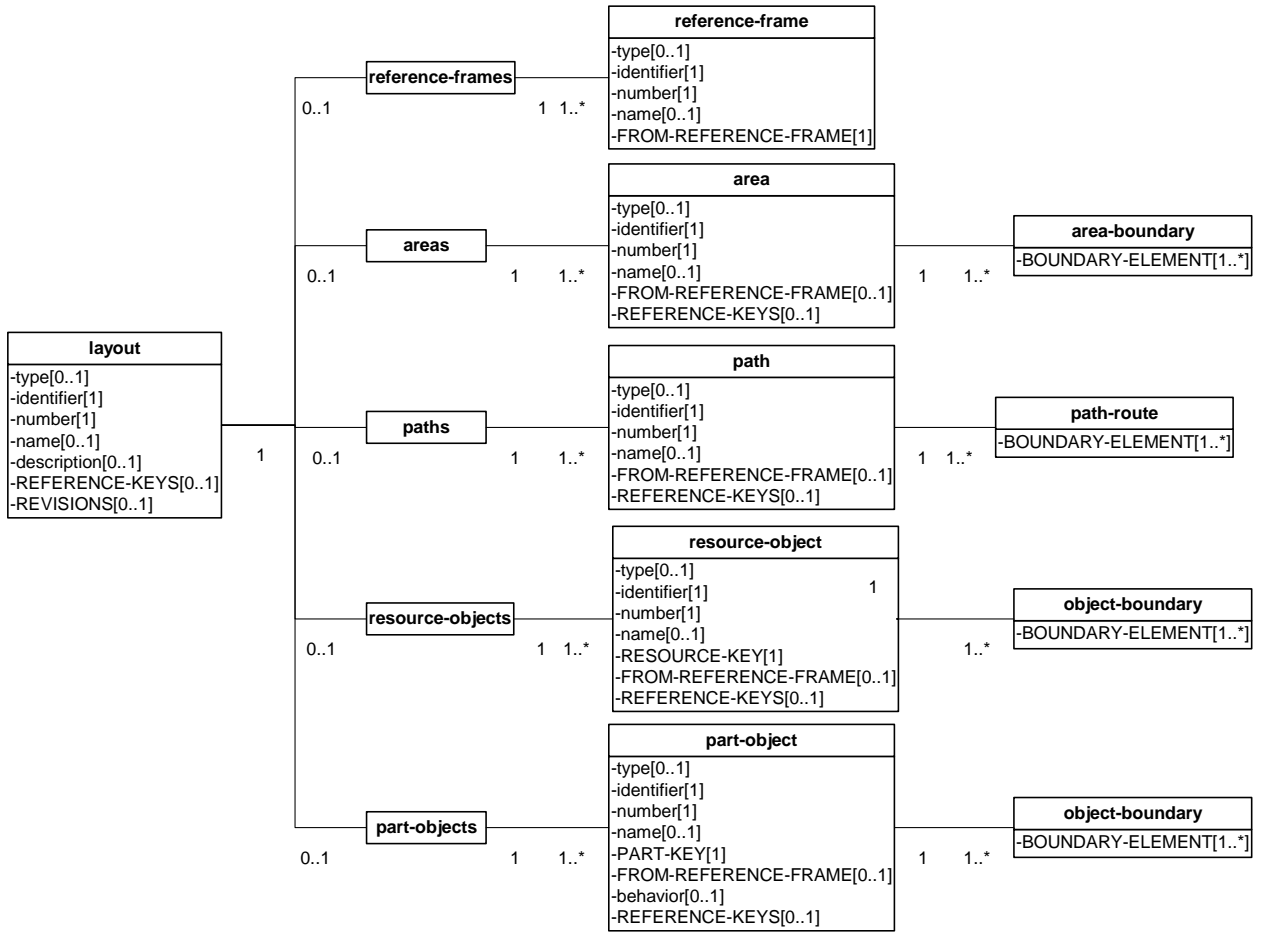
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *identifier*, *name*, *number*, *revisions*, and *type*.

3.11 Layout

layout defines the location of reference points within the shop, area boundaries, paths, resource, and part objects. It contains reference pointers to external graphics files that use appropriate graphics standards to further define these elements.

The UML information model for the *layout* element is shown below:



The XML structure for the *layout* element is shown below:

```
<layout type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <reference-frames>
    <reference-frame type="" identifier="" number="">
      <name />
      <from-reference-frame />
    </reference-frame>
  </reference-frames>
  <areas>
    <area type="" identifier="" number="">
      <name />
      <from-reference-frame />
      <area-boundary>
        <boundary-element />
      </area-boundary>
      <reference-keys />
    </area>
  </areas>
  <paths>
    <path type="" identifier="" number="">
      <name />
      <from-reference-frame />
      <path-route>
        <boundary-element />
      </path-route>
      <reference-keys />
    </path>
  </paths>
  <resource-objects>
    <resource-object type="" identifier="" number="">
      <name />
      <resource-key />
      <from-reference-frame />
      <object-boundary>
        <boundary-element />
      </object-boundary>
      <reference-keys />
    </resource-object>
  </resource-objects>
  <part-objects>
    <part-object type="" identifier="" number="">
      <name />
      <part-key />
      <from-reference-frame />
      <object-boundary>
        <boundary-element />
      </object-boundary>
      <behavior />
      <reference-keys />
    </part-object>
  </part-objects>
</layout>
```

</layout>

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

<Note: Definitions of the following elements are not provided here due to these elements are still under development.>

area:
area-boundary:
areas:
behavior:
layout:
object-boundary:
part-object:
part-objects:
path:
path-route:
paths:
reference-frame:
reference-frames:
resource-object:
resource-objects:

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this element: *part-key*, *reference-keys*, and *resource-key*.

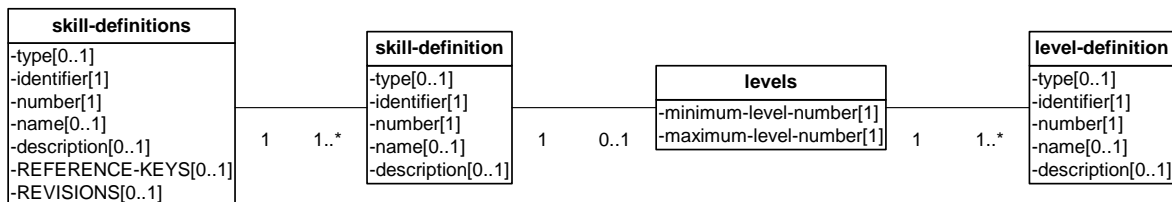
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *boundary-element*, *description*, *from-reference-frame*, *identifier*, *name*, *number*, *revisions*, and *type*.

3.12 Skill Definitions

skill-definitions lists the skills that an employee may possess and the levels of proficiency associated with those skills. *skill-definitions* are referenced in *employee* definitions to identify the skill types and levels each employee has. They are also referenced in employee resource requirements contained in *process-plans* and *work items*. An example of a skill might be the ability to operate a particular type of machine. *levels* indicate the range of skill levels that an employee may possess.

The UML information model for the *skill-definitions* element is shown below:



The XML structure for the *skill-definitions* element is shown below:

```
<skill-definitions type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <skill-definition type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <levels>
      <minimum-level-number />
      <maximum-level-number />
      <level-definition type="" identifier="" number="">
        <name />
        <description />
      </level-definition>
    </levels>
  </skill-definition>
</skill-definitions>
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

level-definition: a definition of a particular skill level. A *level-definition* is defined by a *type*, an *identifier*, a *number*, a *name*, and a *description*.

levels: An element that defines the levels associated with a particular skill. *levels* contains a *minimum-level-number*, a *maximum-level-number*, and one or more *level-definition* data elements.

skill-definition: an element for defining the allowable skills that employees may possess, and the levels of proficiency associated with each skill. A *skill-definition* contains a *type* attribute, an *identifier* attribute, a *number* attribute, a *name* element, a *description* element, and a *levels* complex element.

skill-definitions: a collection of individual skill definitions.

b) Data Element Keys, Multi-keys, and Key Lists

reference-keys is used in this structure. Elements within *employees* and *resources-required* use a *skill-multi-keys* to reference the data elements in the *skill-definitions*.

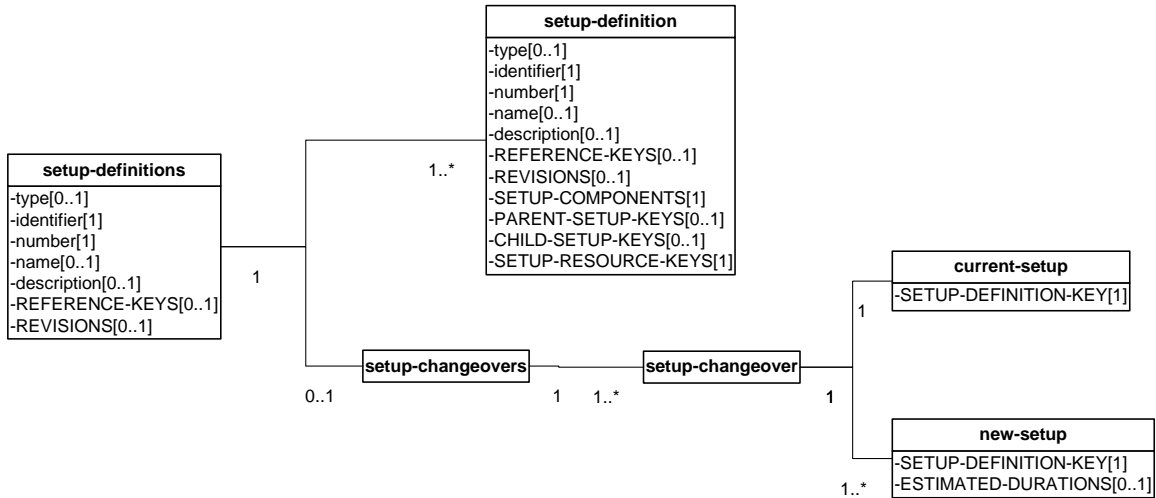
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *identifier*, *maximum-level-number*, *minimum-level-number*, *name*, *number*, *revisions*, and *type*.

3.13 Setup Definitions

setup-definitions typically specifies tool or fixture setups on a machine. Tool setups are typically the tools that are required in the tool magazine. Fixture setups are workholding devices mounted on the machine. Setups may also apply to cranes or stations.

The UML information model for the *setup-definitions* element is shown below:



The XML structure for the *setup-definitions* element is shown below:

```

<setup-definitions type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <setup-definition type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <setup-components />
    <parent-setup-keys />
    <child-setup-keys />
    <setup-resource-keys />
  </setup-definition>
  <setup-changeovers>
    <setup-changeover>
      <current-setup>
        <setup-definition-key />
      </current-setup>
      <new-setup>
        <setup-definition-key />
        <estimated-durations />
      </new-setup>
    </setup-changeover>
  </setup-changeovers>
</setup-definitions>

```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

current-setup: an initial setup definition for which a changeover time is being specified.

new-setup: the new setup definition that a resource's setup is to be changed into.

setup-changeover: information about the machine or other resources changeover from one existing setup to any other setup configuration. It identifies the current setup and new setup by setup definition keys and the time requirement for doing the changeover.

setup-changeovers: a collection of individual setup changeovers that corresponds to a row in a changeover matrix. The row identifies one current configuration and all allowable new configurations.

setup-definition: a specification for a class of setup configuration for a resource, e.g., a machine, a station, or a crane.

setup-definitions: a collection of setup definitions.

b) Data Element Keys, Multi-keys, and Key Lists

child-setup-keys, *parent-setup-keys*, *reference-keys*, *setup-definition-key*, and *setup-resource-keys* are used in this structure. Elements within *resources-assigned*, *resources-used*, *setup-configurations*, *stations*, and *work-assignments* use a *setup-definition-key* to reference the data elements in the *setup-definitions*.

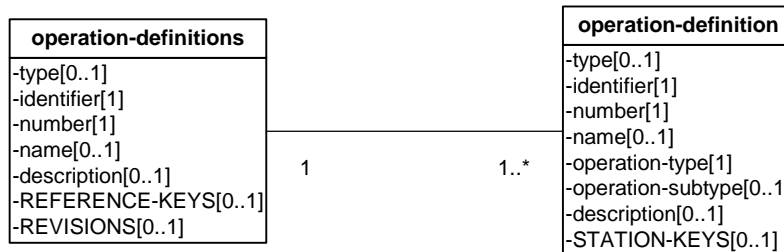
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *estimated-durations*, *identifier*, *name*, *number*, *revisions*, *type*, and *setup-components*.

3.14 Operation Definitions

operation-definitions lists all the definitions of the operations that may be performed at a particular station or group of stations in the shop.

The UML information model for the *operation-definitions* element is shown below:



The XML structure for the *operation-definitions* element is shown below:

```

<operation-definitions type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <operation-definition type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
  </operation-definition>
</operation-definitions>
  
```

```

    <operation-type />
    <operation-subtype />
    <description />
    <station-keys />
  </operation-definition>
</operation-definitions>

```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

operation-definition: an element for specifying the particular operations that may be performed at a particular station or group of stations in the shop.

operation-definitions: a collection of operation definitions.

b) Data Element Keys, Multi-keys, and Key Lists

The *reference-keys* and *station-keys* elements are used in this structure. Elements within *plan-definition* use an *operation-definition-key* to reference the data elements in the *operation-definitions*.

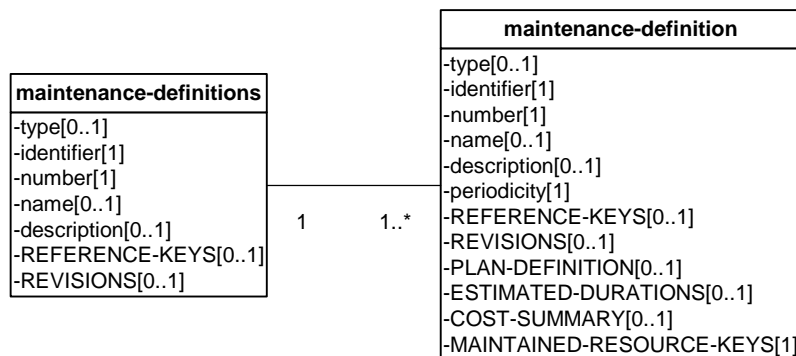
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *identifier*, *name*, *number*, *revisions*, *type*, *operation-subtype*, and *operation-type*.

3.15 Maintenance Definitions

maintenance-definitions specifies preventive or corrective maintenance to be done on machines or other maintained resources.

The UML information model for the *maintenance-definitions* element is shown below:



The XML structure for the *maintenance-definitions* element is shown below:

```

<maintenance-definitions type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />

```

```

<maintenance-definition type="" identifier="" number="">
  <name />
  <description />
  <periodicity />
  <reference-keys />
  <revisions />
  <plan-definition />
  <estimated-durations />
  <cost-summary />
  <maintained-resource-keys />
</maintenance-definition>
</maintenance-definitions>

```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

maintenance-definition: a specification for performing a particular maintenance action on a maintained resource.

maintenance-definitions: a collection of maintenance definitions for one or more maintained resources, for example a group of machines.

periodicity: the frequency at which this maintenance action should be performed.

b) Data Element Keys, Multi-keys, and Key Lists

The *maintained-resource-keys* and *reference-keys* elements are used in this structure. Elements within *maintenance-orders* use a *maintenance-definition-key* to reference the data elements in the *maintenance-definitions* element.

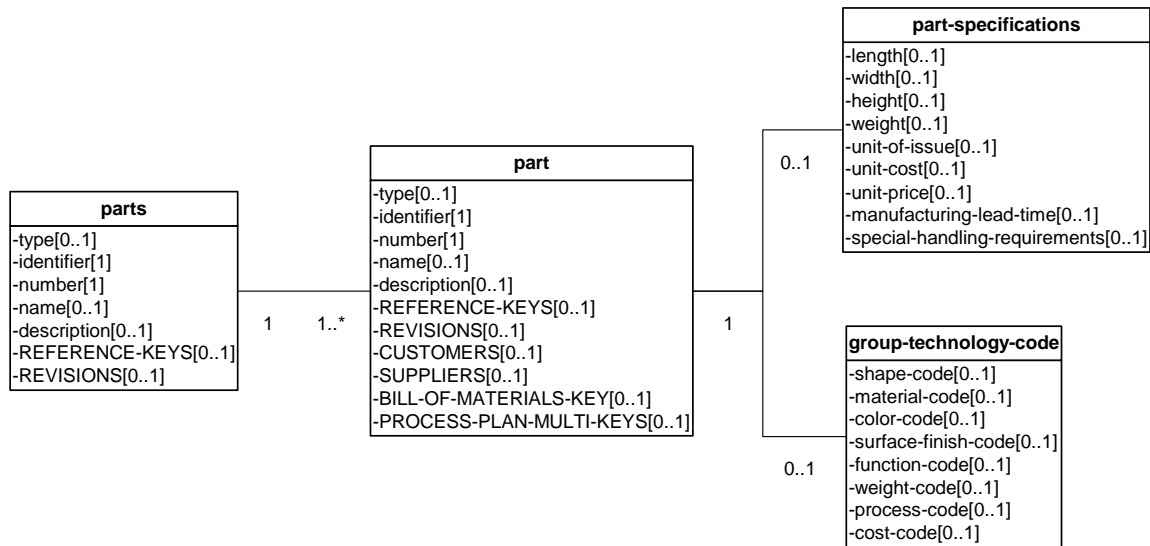
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *cost-summary*, *description*, *estimated-durations*, *identifier*, *name*, *number*, *plan-definition*, *revisions*, and *type*.

3.16 Parts

parts provides elements for part specifications, group technology codes, customers, suppliers, as well as links to bill of materials, process plans, drawings, part models, and other references.

The UML information model for the *parts* element is shown below:



The XML structure for the *parts* element is shown below:

```

<parts type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <part type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <part-specifications>
      <length />
      <width />
      <height />
      <weight />
      <unit-of-issue />
      <unit-cost />
      <unit-price />
      <manufacturing-lead-time />
      <special-handling-requirements />
    </part-specifications>
    <group-technology-code>
      <shape-code />
      <material-code />
      <color-code />
      <surface-finish-code />
      <function-code />
      <weight-code />
      <process-code />
      <cost-code />
    </group-technology-code>
    <customers />
    <suppliers />
    <bill-of-materials-key />
  </part>
</parts>

```

```
        <process-plan-multi-keys />
    </part>
</parts>
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

group-technology-code: a code system that defines a particular part. The system identifies the part's shape, material, color, surface finish, function, weight, process, and cost using a predefined set of codes.

height: a number indicating the height of the part in the specified units of measurement.

length: a number indicating the length of the part in the specified units of measurement.

manufacturing-lead-time: the manufacturing lead time for the part in days.

part: a record for a part that may be a component, work-in-process, or finished goods.

parts: a collection of part records.

part-specifications: a section of the part element containing part specifications.

special-handling-requirements: a note for any special handling that may be required because the part is fragile, contains hazardous materials, etc.

unit-cost: the cost to make or buy the part.

unit-price: the price at which the part is sold.

unit-of-issue: the unit type for which the part is issued, e.g., each, dozen, box.

weight: the weight of the part in the specified units of measurement.

width: the width of the part in the specified units of measurement.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this structure: *bill-of-materials-key*, *process-plan-multi-keys*, and *reference-keys*. Elements within *bill-of-materials*, *inventory*, *layout*, *process-plans*, *procurements*, and *work* use the *part-key* to reference the data elements in the *parts* element.

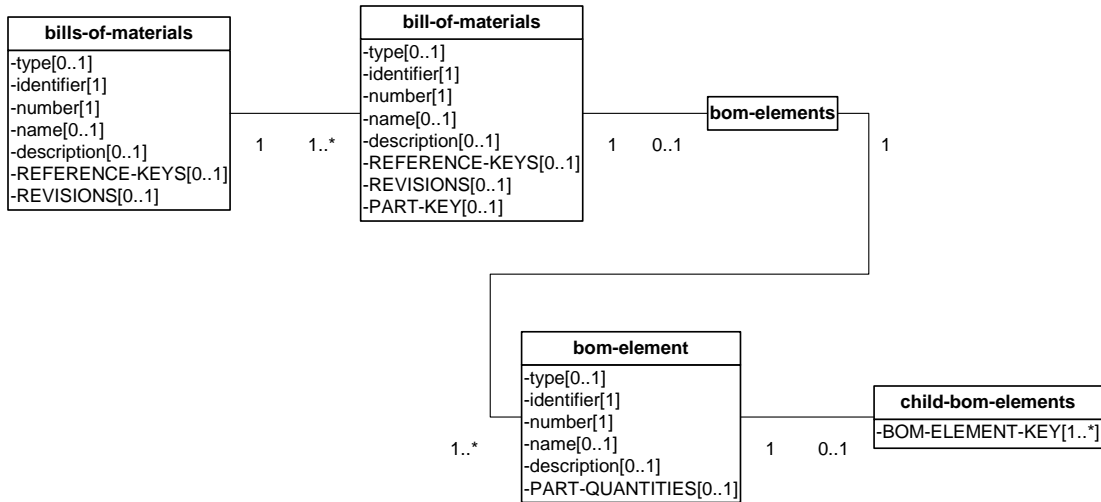
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *color-code*, *cost-code*, *customers*, *description*, *function-code*, *identifier*, *material-code*, *name*, *number*, *process-code*, *revisions*, *shape-code*, *suppliers*, *surface-finish-code*, *type*, and *weight-code*.

3.17 Bills-of-Materials

bills-of-materials cross-references the parts and quantities required in a hierarchical bill-of-materials structure.

The UML information model for the *bills-of-materials* element is shown below:



The XML structure for the *bills-of-materials* element is shown below:

```

<bills-of-materials type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <bill-of-materials type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <part-key />
    <bom-elements>
      <bom-element type="" identifier="" number="">
        <name />
        <description />
        <part-quantities />
        <child-bom-elements>
          <bom-element-key />
        </child-bom-elements>
      </bom-element>
    </bom-elements>
  </bill-of-materials>
</bills-of-materials>
  
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

bill-of-materials: a bill-of-materials structure for one part.

bills-of-materials: a collection of bill-of-materials records for one or more different parts.

bom-element: a component element in the bill-of-materials for one part, i.e., a bill-of-materials line item.

bom-elements: a collection of component elements that makes up the bill-of-materials for a part.

child-bom-elements: a collection of keys for component elements of a single bill-of-materials element or line item.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this structure: *bom-element-key*, *part-key*, and *reference-keys*. Elements within *parts* use the *bill-of-materials-key* to reference the data elements in the *bills-of-materials* element.

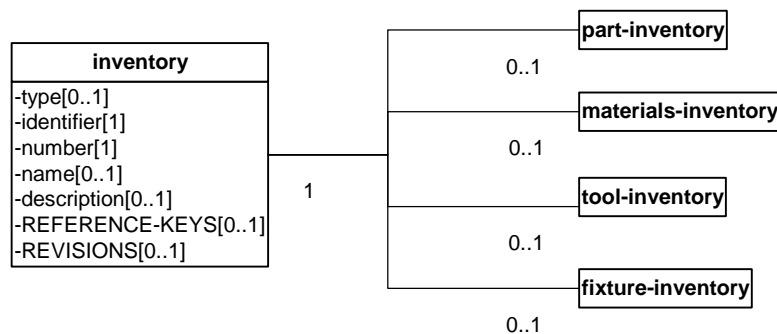
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *identifier*, *name*, *number*, *part-quantities*, *revisions*, and *type*.

3.18 Inventory

inventory identifies the instances and locations for part, materials, tool, and fixture inventory.

The UML information model for the *inventory* element is shown below:



The XML structure for the *inventory* element is shown below:

```
<inventory type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <part-inventory />
  <materials-inventory />
  <tool-inventory />
  <fixture-inventory />
</inventory>
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

fixture-inventory: a collection of inventory records for fixtures and sets of fixture components.

inventory: a collection of inventories for parts, materials, tools, and fixtures.

materials-inventory: a collection of inventory records for materials, i.e., goods that may be partially-consumed (bar stock, sheet stock, wire spools, etc.)

part-inventory: a collection of inventory records for component parts, work-in-process, and finished goods.

tool-inventory: a collection of inventory records for tools and sets of tools.

b) Data Element Keys, Multi-keys, and Key Lists

The *reference-keys* element is used in this element. Elements within *layout*, *resources-assigned*, *resources-used* and *setup-definitions* use *resource-key*, *part-instance-key*, and *setup-resource-keys* to reference the data elements in the *inventory* element.

c) Common Data Elements

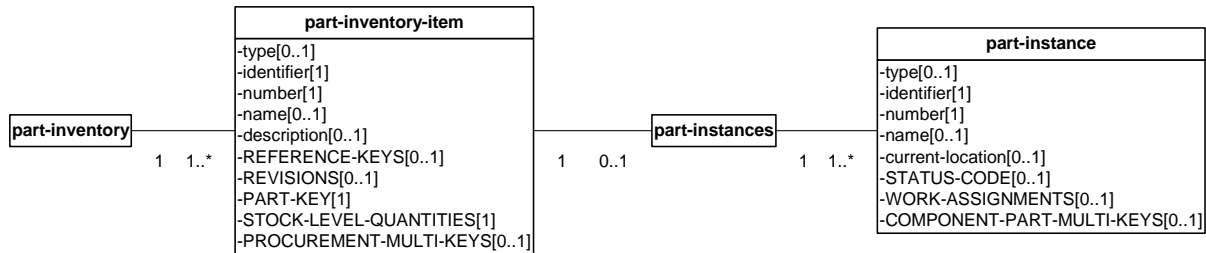
See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *identifier*, *name*, *number*, *revisions*, and *type*.

d) Example Part

3.19 Part Inventory

part-inventory identifies the instances and locations for inventory of individual parts, such as components, work-in-process, and finished goods.

The UML information model for the *part-inventory* element is shown below:



The XML structure for the *part-inventory* element is shown below:

```

<part-inventory>
  <part-inventory-item type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <part-key />
    <stock-level-quantities />
  <part-instances>
    <part-instance type="" identifier="" number="">
      <name />
  
```

```

        <current-location />
        <status-code />
        <work-assignments />
        <component-part-multi-keys />
    </part-instance>
</part-instances>
<procurement-multi-key />
</part-inventory-item>
</part-inventory>

```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

part-instance: an individual instance of a part number in inventory, each instance has a current location, a status code, possibly work assignments that is allocated to, and pointers to component part instances if it is an assembly.

part-instances: a collection of instances of a part number in inventory.

part-inventory: a collection of inventory records for one or more part numbers.

part-inventory-item: the record for one part number maintained in inventory.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this structure: *component-part-multi-key*, *part-key*, *procurement-multi-keys*, and *reference-keys*. Elements within *resources-assigned* and *resources-used* use a *part-instance-key* to reference the data elements in the *part-inventory* element.

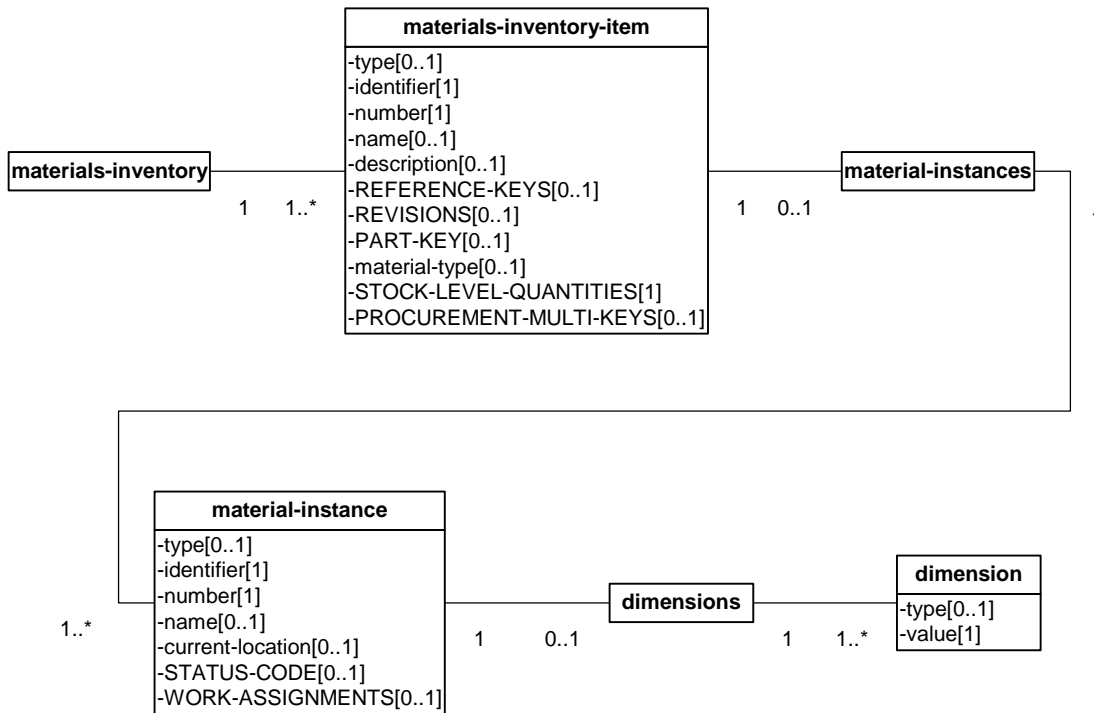
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *current-location*, *description*, *identifier*, *name*, *number*, *revisions*, *status-code*, *stock-level-quantities*, *type*, and *work-assignments*.

3.20 Materials Inventory

materials-inventory identifies the instances and locations for materials inventory. Materials are parts that may be partially consumed, e.g., sheet stock, bar stock, tubing, spools of wire.

The UML information model for the *materials-inventory* element is shown below:



The XML structure for the *materials-inventory* element is shown below:

```

<materials-inventory>
  <materials-inventory-item type="" identifier="" number="">
    <name/>
    <description />
    <reference-keys />
    <revisions />
    <part-key />
    <material-type />
    <stock-level-quantities />
    <material-instances>
      <material-instance type="" identifier="" number="">
        <name />
        <current-location />
        <status-code />
        <work-assignments />
        <dimensions>
          <dimension>
            <type />
            <value />
          </dimension>
        </dimensions>
      </material-instance>
    </material-instances>
    <procurement-multi-keys />
  </materials-inventory-item>
</materials-inventory>
  
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

dimension: one dimension of piece of a materials-type part instance.

dimensions: a collection of dimensions of a piece of a materials-type part instance.

material-instance: an individual instance of a materials-type part number in inventory, each instance has a current location, a status code, possibly work assignments that is allocated to, and dimensions of the piece of materials.

material-instances: a collection of instances of a materials-type part number in inventory

materials-inventory: a collection of inventory records for one or more materials-type part numbers.

materials-inventory-item: the record for one materials-type part number maintained in inventory.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this structure: *part-key*, *procurement-multi-keys*, and *reference-keys*.

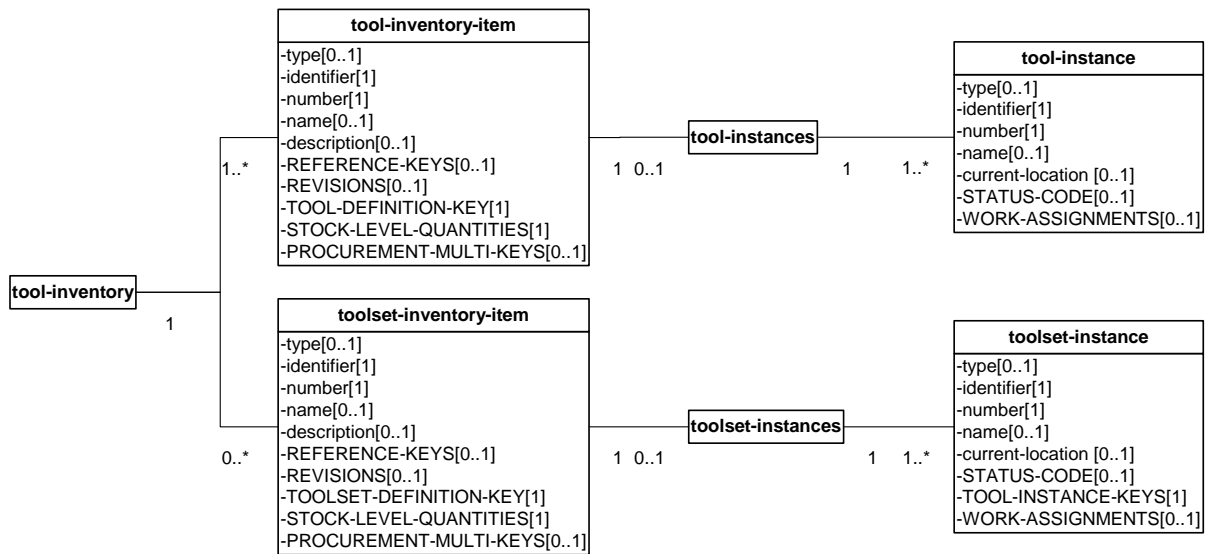
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *current-location*, *description*, *identifier*, *material-type*, *name*, *number*, *revisions*, *status-code*, *stock-level-quantities*, *type*, *value*, and *work-assignments*.

3.21 Tool Inventory

tool-inventory identifies the instances and locations for tools and toolsets.

The UML information model for the *tool-inventory* element is shown below:



The XML structure for the *tool-inventory* element is shown below:

```

<tool-inventory>
  <tool-inventory-item type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <tool-definition-key />
    <stock-level-quantities />
    <tool-instances>
      <tool-instance type="" identifier="" number="">
        <name />
        <current-location />
        <status-code />
        <work-assignments />
      </tool-instance>
    </tool-instances>
    <procurement-multi-keys />
  </tool-inventory-item>
  <toolset-inventory-item type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <toolset-definition-key />
    <stock-level-quantities />
    <toolset-instances>
      <toolset-instance type="" identifier="" number="">
        <name />
        <current-location />
        <status-code />
        <tool-instance-keys />
        <work-assignments />
      </toolset-instance>
    </toolset-instances>
  </toolset-inventory-item>
</tool-inventory>
  
```

```
        </toolset-instances>
        <procurement-multi-keys />
    </toolset-inventory-item>
</tool-inventory>
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

tool-instance: a specific instance of a tool, i.e., copy of a tool type that has a unique identifier and location.

tool-instances: a collection of tool instances of a particular tool type.

tool-inventory: a collection of tool inventory item records.

tool-inventory-item: the inventory records for a particular type of tool, i.e., tool definition.

toolset-instance: a specific instance of a toolset, i.e., a group of tool that has a unique identifier and location.

toolset-instances: a collection of toolset instances of a particular toolset type.

toolset-inventory-item: the inventory records for a particular type of toolset, i.e., tool group configuration.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this structure: *procurement-multi-keys*, *reference-keys*, *tool-definition-key*, *tool-instance-keys*, and *toolset-definition-key*.

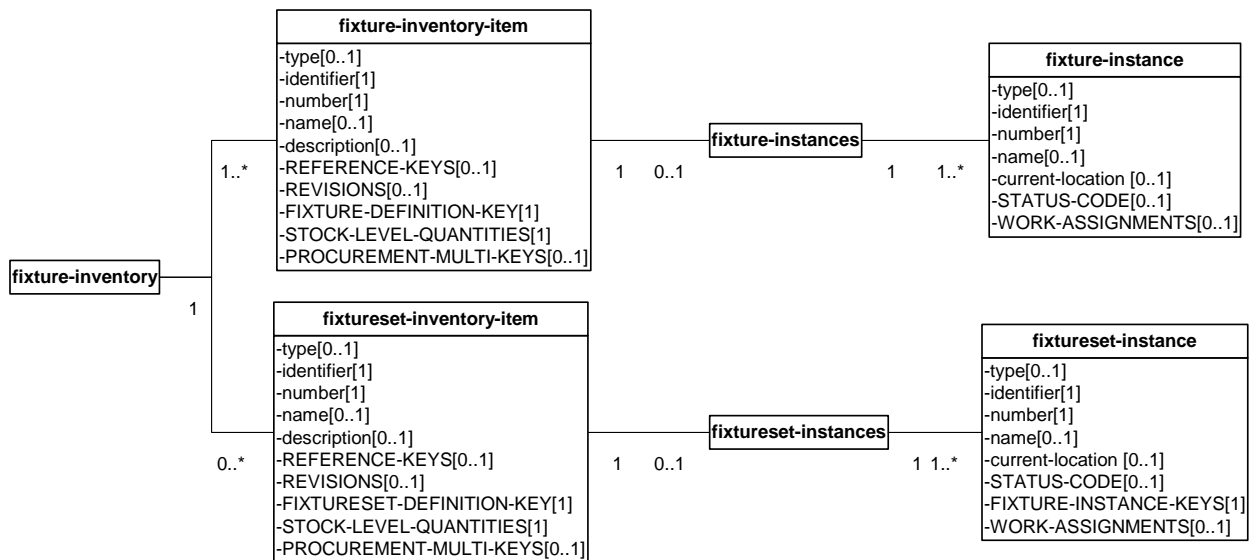
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *current-location*, *description*, *identifier*, *name*, *number*, *revisions*, *status-code*, *stock-level-quantities*, *type*, and *work-assignments*.

3.22 Fixture Inventory

fixture-inventory identifies the instances and locations for workholding devices, i.e., fixtures and fixture sets.

The UML information model for the *fixture-inventory* element is shown below:



The XML structure for the *fixture-inventory* element is shown below:

```

<fixture-inventory>
  <fixture-inventory-item type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <fixture-definition-key />
    <stock-level-quantities />
    <fixture-instances>
      <fixture-instance type="" identifier="" number="">
        <name />
        <current-location />
        <status-code />
        <work-assignments />
      </fixture-instance>
    </fixture-instances>
    <procurement-multi-keys />
  </fixture-inventory-item>
  <fixtureset-inventory-item type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <fixtureset-definition-key />
    <stock-level-quantities />
    <fixtureset-instances>
      <fixtureset-instance type="" identifier="" number="">
        <name />
        <current-location />
        <status-code />
        <fixture-instance-keys />
        <work-assignments />
      </fixtureset-instance>
    </fixtureset-instances>
  </fixtureset-inventory-item>
</fixture-inventory>
  
```

```
        </fixtureset-instance>
    </fixtureset-instances>
    <procurement-multi-keys />
</fixtureset-inventory-item>
</fixture-inventory>
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

fixture-instance: a specific instance of a fixture, i.e., copy of a fixture type that has a unique identifier and location.

fixture-instances: a collection of fixture instances of a particular fixture type.

fixture-inventory: a collection of fixture inventory item records.

fixture-inventory-item: the inventory records for a particular type of fixture, i.e., fixture definition.

fixtureset-instance: a specific instance of a fixtureset, i.e., a group of fixture that has a unique identifier and location.

fixtureset-instances: a collection of fixtureset instances of a particular fixtureset type.

fixtureset-inventory-item: the inventory records for a particular type of fixtureset, i.e., fixture group configuration.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this structure: *fixture-definition-key*, *fixture-instance-keys*, *fixtureset-definition-key*, *procurement-multi-keys*, and *reference-keys*.

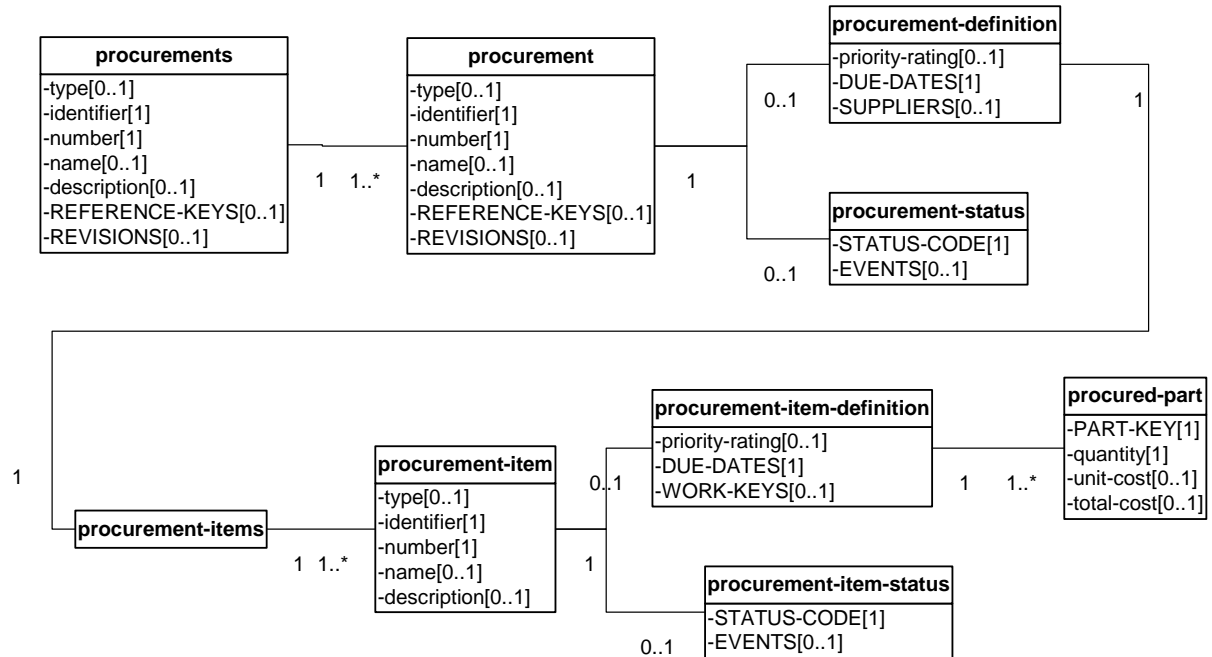
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *current-location*, *description*, *identifier*, *name*, *number*, *revisions*, *status-code*, *stock-level-quantities*, *type*, and *work-assignments*.

3.23 Procurements

procurements identifies the external purchases that have been created to satisfy part inventory and manufacturing requirements.

The UML information model for the *procurements* element is shown below:



The XML structure for the *procurements* element is shown below:

```

<procurements type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <procurement type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <procurement-definition>
      <priority-rating />
      <due-dates />
      <suppliers />
    <procurement-items>
      <procurement-item type="" identifier="" number="">
        <name />
        <description />
        <procurement-item-definition>
          <priority-rating />
          <due-dates />
          <procured-part>
            <part-key />
            <quantity />
            <unit-cost />
            <total-cost />
          </procured-part>
          <work-keys />
        </procurement-item-definition>
      </procurement-item>
    </procurement-items>
  </procurement>
</procurements>
  
```

```

        <procurement-item-status>
            <status-code />
            <events />
        </procurement-item-status>
    </procurement-item>
</procurement-items>
</procurement-definition>
<procurement-status>
    <status-code />
    <events />
</procurement-status>
</procurement>
</procurements>

```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

procured-part: a section of the procurement item definition that identifies the part key, quantities, unit and total cost information.

procurement: one procurement order to a supplier.

procurement-definition: a subsection of an individual procurement order that defines the items to be purchased, due dates, priorities, etc.

procurement-item: the section of a procurement that contains a line item for one item.

procurement-item-definition: the definition of the characteristics of one procurement item, e.g., the part key, due dates, priority.

procurement-items: a collection of procured items in a single procurement order.

procurement-item-status: the status of one procured item in a single procurement order.

procurements: a collection of one or more procurement orders.

procurement-status: the overall status of a single procurement order.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this structure: *part-key*, *reference-keys*, and *work-keys*. Structures within *inventory* use the *procurement-multi-keys* to reference the data elements in the *procurements* structure.

c) Common Data Elements

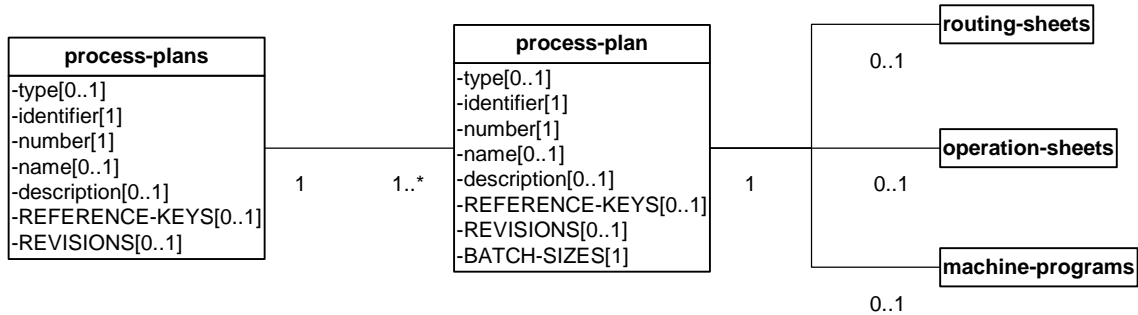
See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *due-dates*, *events*, *identifier*, *name*, *number*, *priority-rating*, *quantity*, *revisions*, *status-code*, *suppliers*, *total-cost*, *type*, and *unit-cost*.

3.24 Process Plans

process-plans specifies a set of process plans that are associated with production and support activities for a particular part or parts. A process plan has routings sheets and/or operation sheets that correspond to the job

and/or task level in the work hierarchy. *process-plan* contains *routing-sheets*, *operation-sheets*, and *machine-programs*.

The UML information model for the *process-plans* element is shown below:



The XML structure for the *process-plans* element is shown below:

```

<process-plans type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <process-plan type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <batch-sizes />
    <routing-sheets />
    <operation-sheets />
    <machine-programs />
  </process-plan>
</process-plans>

```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

machine-programs: (refer to Section 3.27)

operation-sheets: (refer to Section 3.26)

process-plan: a group of documents that specifies the manufacturing processes for a particular part or parts. Documents include routing sheets for job level work, operation sheets for task level work, and machine programs.

process-plans: a collection of individual process plans, routing sheets, operation sheets, and machine programs.

routing-sheets: (refer to Section 3.25)

b) Data Element Keys, Multi-keys, and Key Lists

The *reference-keys* element is used in this element. Elements within *parts* and *works* use a *process-plan-multi-key(s)* to reference the data element in the *process-plans* element.

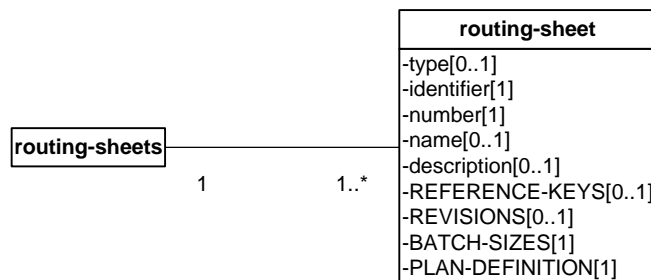
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *batch-sizes*, *description*, *identifier*, *name*, *number*, *revisions*, and *type*.

3.25 Routing Sheets

routing-sheets are complex process planning documents that specify production and support activities that may take place at different stations or work areas within the shop. Routing sheets correspond to the job level in the work hierarchy. A routing sheet defines the steps, precedence constraints between steps, and resources associated with the production of parts and performance of support activities. It may require one or more routing sheets to completely define a process plan. One routing sheet corresponds to one job. A job may contain one or more tasks. The steps of the routing sheet typically correspond to tasks that need to be created.

The UML information model for the *routing-sheets* element is shown below:



The XML structure for the *routing-sheets* element is shown below:

```
<routing-sheets>
  <routing-sheet type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <batch-sizes />
    <plan-definition />
  </routing-sheet>
</routing-sheets>
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Element

routing-sheet: a document that specifies a complex plan for production that typically takes place at multiple stations or areas where work is performed. The routing sheet determines what tasks need to be created. Steps in the routing may correspond to a critical path network where many activities go on in parallel.

routing-sheets: a collection of individual routing sheets.

b) Data Element Keys, Multi-keys, and Key Lists

This element uses *reference-keys*. Elements within *jobs* and *tasks* use *process-plan-multi-key* to reference the data elements in the *routing-sheets* element.

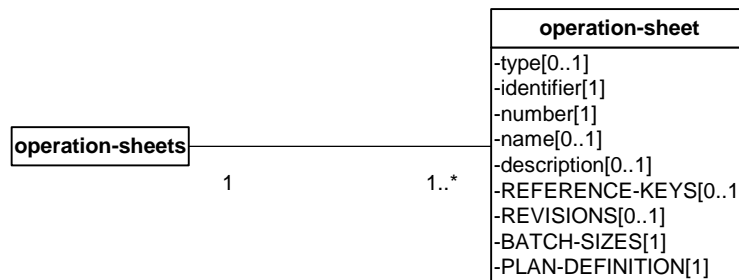
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *batch-sizes*, *description*, *identifier*, *name*, *number*, *plan-definition*, *revisions*, and *type*.

3.26 Operation Sheets

operation-sheets specifies the operation sheets that are associated with production and support activities. Operation sheets correspond to the task level in the work hierarchy. A task normally performs on a workpiece at one given station. An operation sheet defines the steps, precedence constraints between steps, and resources associated with the production of parts and performance of support activities. One operation sheet supports one task, and it requires one or more operation sheets to completely define a process plan.

The UML information model for the *operation-sheets* element is shown below:



The XML structure for the *operation-sheets* element is shown below:

```

<operation-sheets>
  <operation-sheet type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <batch-sizes />
    <plan-definition />
  </operation-sheet>
</operation-sheets>
  
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Element

operation-sheet: information about a set of operations or processes that are typically performed on a workpiece at a specific station and machine. The operation sheet specifies the detailed steps that are associated with carrying out a task. The task typically is associated with one step in a routing sheet level document within a process plan.

operation-sheets: a collection of individual operation sheets.

b) Data Element Keys, Multi-keys, and Key Lists

reference-keys are used in this element. *reference-keys* and *tasks* typically reference individual operation sheets using *process-plan-multi-key*.

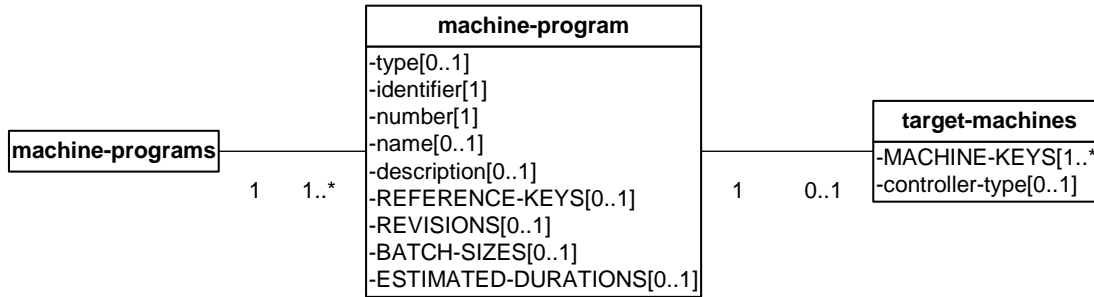
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *batch-sizes*, *description*, *identifier*, *name*, *number*, *plan-definition*, *revisions*, and *type*.

3.27 Machine Programs

machine-programs specifies the machine programs that are used with the operation sheet. It also defines the target machine and program duration of support activities.

The UML information model for the *machine-programs* element is shown below:



The XML structure for the *machine-programs* element is shown below:

```

<machine-programs>
  <machine-program type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <batch-sizes />
    <target-machines>
      <machine-keys />
      <controller-type />
    </target-machines>
    <estimated-durations />
  </machine-program>
</machine-programs>
  
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Element

controller-type: the type of controller the machine uses.

machine-program: the program used with the machine for a particular task and associated step or steps in an operation sheet.

machine-programs: a collection of machine programs.

target-machines: the machines and their controller type that the program was written to execute on.

b) Data Element Keys, Multi-keys, and Key Lists

This element uses *machine-keys* and *reference-keys*. *tasks* references *machine-programs* using *process-plan-multi-key*.

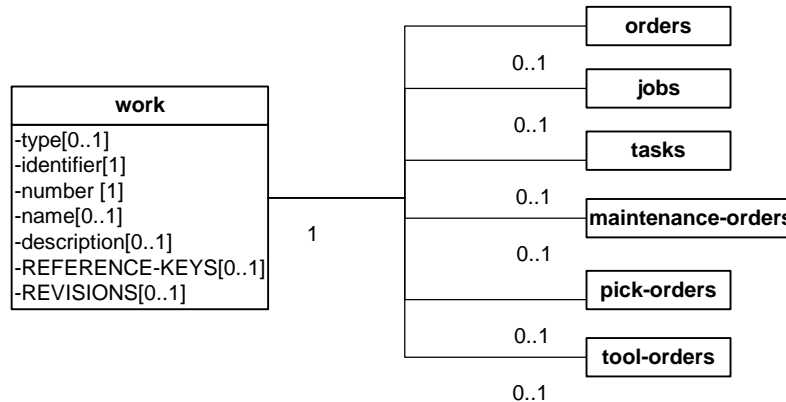
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *batch-sizes*, *description*, *estimated-durations*, *identifier*, *name*, *number*, *revisions*, and *type*.

3.28 Work

The *work* section is used to specify a collection of a hierarchy of production orders, jobs, and tasks. It is also used to specify a collection of internal support orders for maintenance activities, inventory picking, and tool preparation. Standard header data is provided for the collection element: *type*, *identifier*, *number*, *name*, *description*, *reference-keys*, and *revisions*.

The UML information model for the *work* element is shown below:



The XML structure for the *work* element is shown below:

```
<work type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <orders />
  <jobs />
  <tasks />
  <maintenance-orders />
  <pick-orders />
  <tool-orders />
</work>
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

jobs: the high level internal production activities associated with an order to produce goods. A job typically has a routing sheet process plan that specifies the steps of production at multiple stations, resources, and lower level plans (operation sheets) required to complete the job. A job may be decomposed into other jobs or tasks (refer to Section 3.31).

maintenance-orders: maintenance actions on machine tools and other equipment (refer to Section 3.33).

orders: orders for goods to be delivered to internal and external customers. Goods may be drawn from stock or produced to meet order requirements (refer to Section 3.29).

pick-orders: orders to draw goods from inventory (refer to Section 3.34).

tasks: the low level production activities associated with one step of a job level routing sheet. A task takes place at one station and may have an operation sheet that specifies the individual steps to be performed and the resources required (refer to Section 3.32).

tool-orders: tooling orders typically for kitting and/or assembly of tool components and assemblies in the tool crib (refer to Section 3.35).

work: a collection of orders, jobs, tasks, maintenance orders, tool orders, and pick orders.

b) Data Element Keys, Multi-keys, and Key Lists

reference-keys are used in this element. Elements within *child-work-items*, *precedent-constraints*, and *work-assignments* use a *work-keys* to reference the data elements in the *work* element.

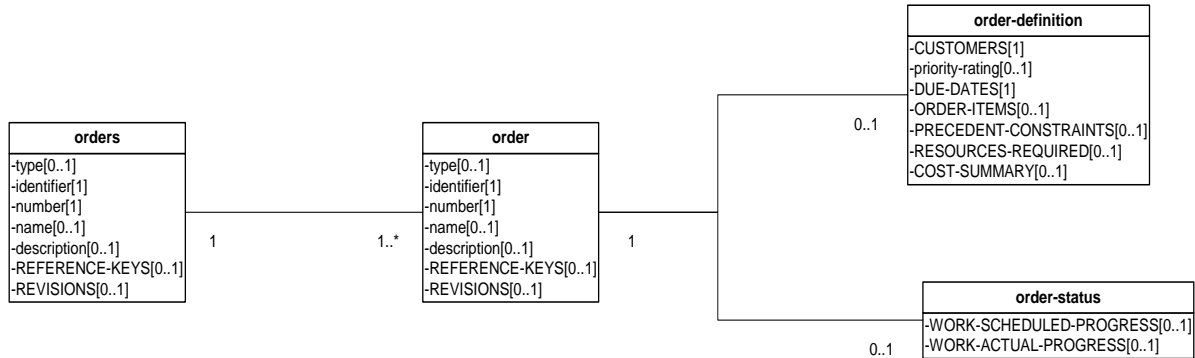
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *identifier*, *name*, *number*, *revisions*, and *type*.

3.29 Orders

orders specifies a collection of production work orders to be processed within the shop. Each *order* contains an *order-definition* and/or an *order-status* section. The *order-definition* contains attributes of the order including a list of *order-items*, i.e., a listing of individual parts. The same part may be listed in the order multiple times in different order items if each instance has unique attributes, e.g., different due dates.

The UML information model for the *orders* element is shown below:



The XML structure for the *orders* element is shown below:

```

<orders type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <order type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <order-definition>
      <customers />
      <priority-rating />
      <due-dates />
      <order-items />
      <precedent-constraints />
      <resources-required />
      <cost-summary />
    </order-definition>
    <order-status>
      <work-scheduled-progress />
      <work-actual-progress />
    </order-status>
  </order>
</orders>
  
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

order: an instance of a production order.

order-definition: information that specifies who the order is for (*customers*), the relative priority of the order (*priority-rating*), critical due dates for the order (*due-dates*), what output products are required (*order-items*), resource requirements (*resources-required*), precedence relationships between order items (*precedent-constraints*), and summary of estimated and actual cost data (*cost summary*).

order-items: information on specific parts that make up the order (refer to Section 3.30)

orders: a collection of individual orders.

order-status: information about scheduled and actual progress towards completing the order. The *work-scheduled-progress* and *work-actual-progress* common data elements are used to organize this information.

b) Data Element Keys, Multi-keys, and Key Lists

reference-keys are used in this element. Elements within *schedules* use an *order-multi-key* to reference the data elements in the *orders* element.

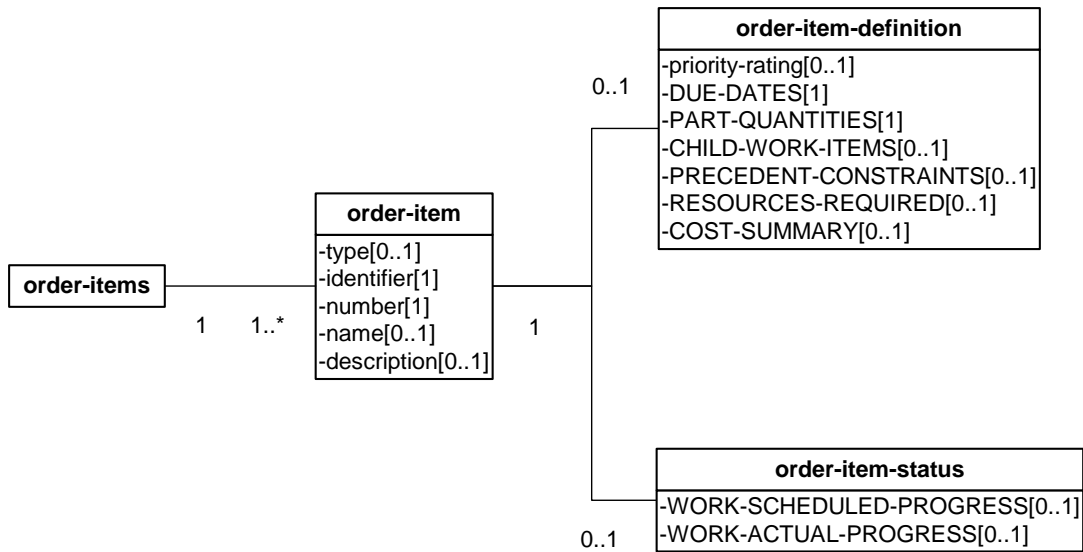
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *cost-summary*, *customers*, *description*, *due-dates*, *identifier*, *name*, *number*, *precedent-constraints*, *priority-rating*, *resources-required*, *revisions*, *type*, *work-actual-progress*, and *work-scheduled-progress*.

3.30 Order Items

order-items lists the individual parts that make up a particular order. Each *order-item* has an *order-item-definition* element that may specify a unique priority rating, due dates, part number, quantity needed, child work items (*jobs*), and summary cost information for a particular order item. The *order-item* also has an *order-status* element that contains scheduled and actual progress towards completing the order item.

The UML information model for the *order-items* element is shown below:



The XML structure for the *order-items* element is shown below:

```

<order-items type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <order-item type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <order-item-definition>
      <priority-rating />
      <due-dates />
      <part-quantities />
      <child-work-items />
      <precedent-constraints />
      <resources-required />
      <cost-summary />
    </order-item-definition>
    <order-item-status>
      <work-scheduled-progress />
      <work-actual-progress />
    </order-item-status>
  </order-item>
</order-items>
  
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

order-item: one type of part item in a production order.

order-item-definition: the relative priority of this order item within the order (*priority-rating*), critical due dates for the order item (*due-dates*), what products are required (*part-quantities*), resource requirements (*resources-required*), precedence relationships between order items (*precedent-constraints*), and summary of estimated and actual cost data (*cost summary*).

order-items: a collection of specific order items associated with individual parts that make up an order.

order-item-status: information about scheduled and actual progress towards completing the order. The *work-scheduled-progress* and *work-actual-progress* common data elements are used to organize this information.

b) Data Element Keys, Multi-keys, and Key Lists

No keys are used in this structure. Elements within *schedules* use an *order-multi-key* to reference the data elements in the *order-items* element.

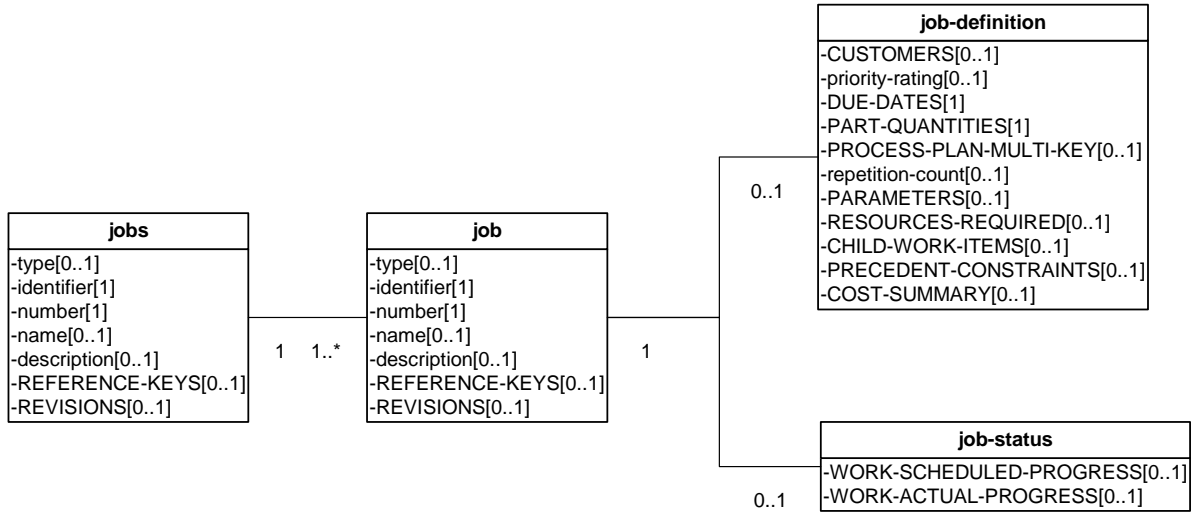
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *child-work-items*, *cost-summary*, *description*, *due-dates*, *identifier*, *name*, *number*, *part-quantities*, *precedent-constraints*, *priority-rating*, *resources-required*, *revisions*, *type*, *work-actual-progress*, and *work-scheduled-progress*.

3.31 Jobs

jobs specifies a collection of individual production jobs to be processed by the shop. Jobs are created to produce the parts called out in the *order-items* section of an *order*. Each *job* has a *job-definition* and a *job-status* section. The *job-definition* section specifies what needs to be done. The *job-status* section specifies planning progress and actual progress towards accomplishing the job. *child-work-items* are the subordinate work items that are created for a job, they may be either other jobs or tasks.

The UML information model for the *jobs* element is shown below:



The XML structure for the *jobs* element is shown below:

```

<jobs type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <job type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <job-definition>
      <customers />
      <priority-rating />
      <due-dates />
      <part-quantities />
      <process-plan-multi-key />
      <repetition-count />
      <parameters />
      <resources-required />
      <child-work-items />
      <precedent-constraints />
      <cost-summary />
    </job-definition>
    <job-status>
      <work-scheduled-progress />
      <work-actual-progress />
    </job-status>
  </job>
</jobs>
  
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

job: a child work item of an order.

job-definition: the relative priority of this job (*priority-rating*), critical due dates for the job (*due-dates*), what products are required (*part-quantities*), resource requirements (*resources-required*), precedence relationships between child work items of this job (*precedent-constraints*), and summary of estimated and actual cost data (*cost summary*).

jobs: a collection of individual jobs.

job-status: information about scheduled and actual progress towards completing the job. The *work-scheduled-progress* and *work-actual-progress* common data elements are used to organize this information.

b) Data Element Keys, Multi-keys, and Key Lists

The following key elements are used in this structure: *process-plan-multi-key* and *references-keys*. Elements within *schedules* use a *job-key* to reference the data elements in the *jobs* element.

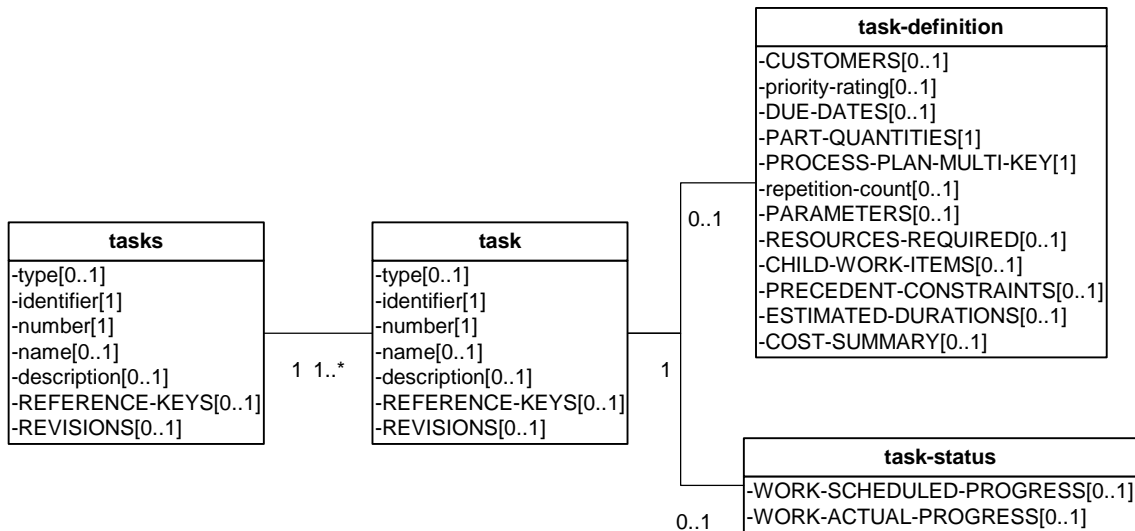
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *child-work-items*, *cost-summary*, *customers*, *description*, *due-dates*, *identifier*, *name*, *number*, *parameters*, *part-quantities*, *precedent-constraints*, *priority-rating*, *repetition-count*, *resources-required*, *revisions*, *type*, *work-actual-progress*, and *work-scheduled-progress*.

3.32 Tasks

tasks specifies the hierarchy of task items to be processed within the shop. Precedence constraints defined in operation sheets are mapped to associated task items. Scheduling data and resource assignments for each task item are maintained in the structure, as well as other data. *jobs* and *tasks* are cross-referenced to each other as well as routing and operation sheets respectively.

The UML information model for the *tasks* element is shown below:



The XML structure for the *tasks* element is shown below:

```

<tasks type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <task type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <task-definition>
      <customers />
      <priority-rating />
      <due-dates />
      <part-quantities />
      <process-plan-multi-key />
      <repetition-count />
      <parameters />
      <resources-required />
      <child-work-items />
      <precedent-constraints />
      <estimated-durations />
      <cost-summary />
    </task-definition>
    <task-status>
      < work-scheduled-progress />
      < work-actual-progress />
    </task-status>
  </task>
</tasks>

```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

task: a child work item of a job.

task-definition: the relative priority of this task (*priority-rating*), critical due dates for the task (*due-dates*), what products are required (*part-quantities*), resource requirements (*resources-required*), precedence relationships between child work items (*precedent-constraints*), and summary of estimated and actual cost data (*cost summary*). Child work items for a task may be *pick-orders* or *tool-orders*.

tasks: a collection of individual tasks.

task-status: information about scheduled and actual progress towards completing the task. The *work-scheduled-progress* and *work-actual-progress* common data elements are used to organize this information.

b) Data Element Keys, Multi-keys, and Key Lists

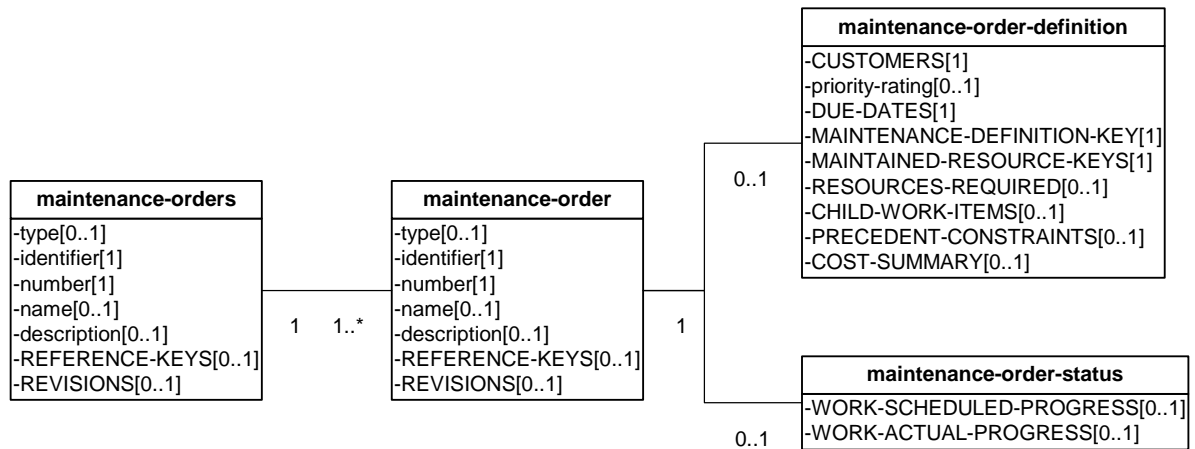
The following key elements are used in this structure: *process-plan-multi-key* and *references-keys*. Elements within *schedules* use a *task-key* to reference the data elements in the *tasks* element.

c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *child-work-items*, *cost-summary*, *customers*, *description*, *due-dates*, *estimated-durations*, *identifier*, *name*, *number*, *parameters*, *part-quantities*, *precedent-constraints*, *priority-rating*, *repetition-count*, *resources-required*, *revisions*, *type*, *work-actual-progress*, and *work-scheduled-progress*.

3.33 Maintenance Orders

maintenance-orders specifies maintenance work to be done on machines or equipment within the shop. The UML information model for the *maintenance-orders* element is shown below:



The XML structure for the *maintenance-orders* element is shown below:

```

<maintenance-orders type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <maintenance-order type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <maintenance-order-definition>
      <customers />
      <priority-rating />
      <due-dates>
      <maintenance-definition-key />
      <maintained-resource-keys />
      <resources-required />
      <child-work-items />
      <precedent-constraints >/
      <cost-summary />
    </maintenance-order-definition>
  </maintenance-order>
  <maintenance-order-status>

```



```

        <work-scheduled-progress />
        <work-actual-progress />
    </maintenance-order-status>
</maintenance-order>
</maintenance-orders>

```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

maintenance-order: an order to perform a maintenance action on one or more maintained resources.

maintenance-order-definition: a section of the maintenance order that identifies the customers, the resource to be maintained, a key to the definition of the maintenance action to be performed, resources required, child work items (typically subordinate maintenance orders), precedence relationships between child work items, and a cost summary.

maintenance-orders: a collection of maintenance orders.

maintenance-order-status: a section contained scheduled and actual progress towards completing the maintenance action.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this structure: *maintained-resource-keys*, *maintenance-definition-key*, and *reference-keys*. Elements within *schedules* use a *maintenance-order-key* to reference the data elements in the *maintenance-orders* element.

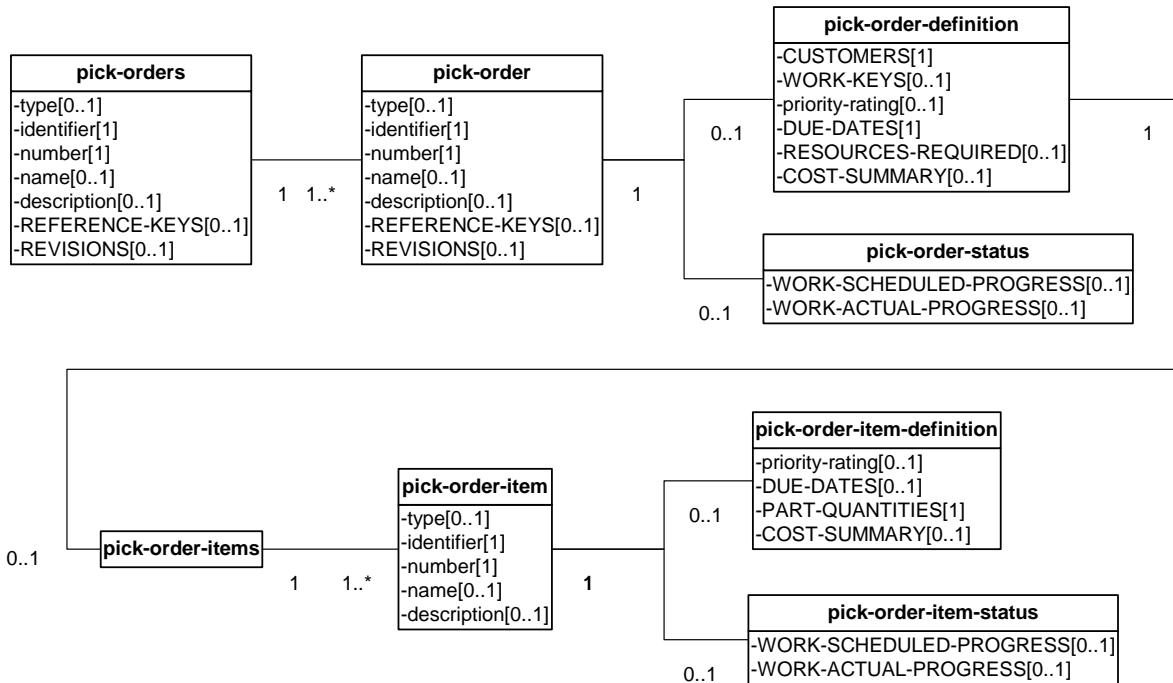
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *child-work-items*, *cost-summary*, *customers*, *description*, *due-dates*, *identifier*, *name*, *number*, *precedent-constraints*, *priority-rating*, *resources-required*, *revisions*, *type*, *work-actual-progress*, and *work-scheduled-progress*.

3.34 Pick Orders

pick-orders specifies orders to draw goods from inventory

The UML information model for the *pick-orders* element is shown below:



The XML structure for the *pick-orders* element is shown below:

```

<pick-orders type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <pick-order type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <pick-order-definition>
      <customers />
      <work-keys />
      <priority-rating />
      <due-dates />
      <resources-required />
      <pick-order-items>
        <pick-order-item type="" identifier="" number="">
          <name />
          <description />
          <pick-order-item-definition>
            <priority-rating />
            <due-dates />
            <part-quantities />
            <cost-summary />
          </pick-order-item-definition>
          <pick-order-item-status>

```

```

        <work-scheduled-progress />
        <work-actual-progress />
    </pick-order-item-status>
</pick-order-item>
</pick-order-items>
<cost-summary />
</pick-order-definition>
<pick-order-status>
    <work-scheduled-progress />
    <work-actual-progress />
</pick-order-status>
</pick-order>
</pick-orders>

```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

pick-order: an order to pull an item or a group of items from inventory.

pick-order-definition: a section of the pick order specifying the customer, priority rating, due dates, the work to be done, the work item that the pick order serves, the individual pick order items, resources required, etc.

pick-order-item: a specific item to be drawn from inventory.

pick-order-item-definition: specification of the details of one part item type and quantity to be drawn from inventory.

pick-order-items: a list of individual pick order items in a pick order definition.

pick-order-item-status: the status of one item to be picked in a pick order.

pick-order-status: the overall status of the pick order.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this structure: *reference-keys* and *work-keys*. Elements within *schedules* use a *pick-order-key* to reference the data elements in the *pick-orders* element.

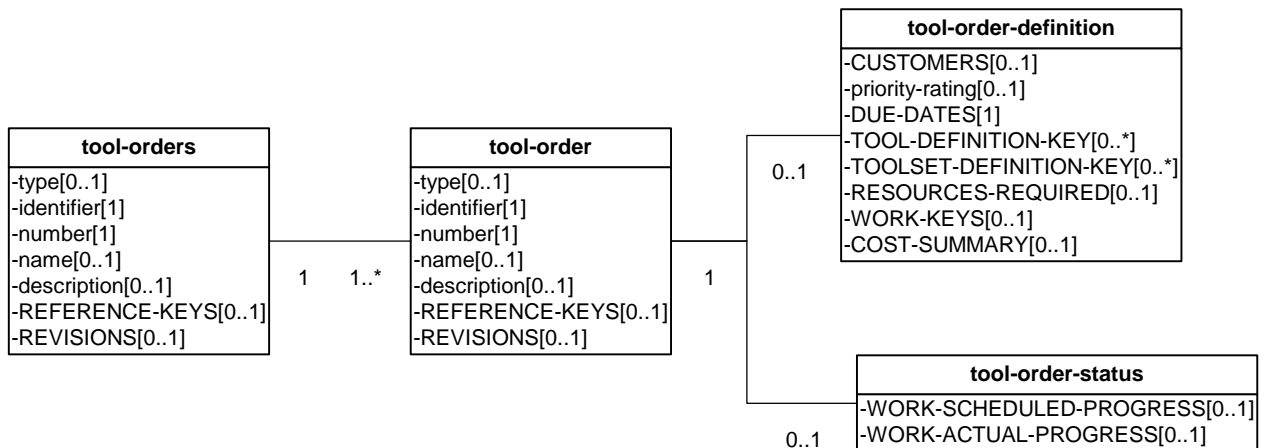
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *cost-summary*, *customers*, *description*, *due-dates*, *identifier*, *name*, *number*, *part-quantities*, *priority-rating*, *resources-required*, *revisions*, *type*, *work-actual-progress*, and *work-scheduled-progress*.

3.35 Tool Orders

tool-orders specifies orders for kitting and/or assembly of tool components and assemblies in the tool crib area.

The UML information model for the *tool-orders* element is shown below:



The XML structure for the *tool-orders* element is shown below:

```

<tool-orders type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <tool-order type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <tool-order-definition>
      <customers />
      <priority-rating />
      <due-dates />
      <tool-definition-key />
      <toolset-definition-key />
      <resources-required />
      <work-keys />
      <cost-summary />
    </tool-order-definition>
    <tool-order-status>
      <work-scheduled-progress />
      <work-actual-progress />
    </tool-order-status>
  </tool-order>
</tool-orders>
  
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

tool-order: an order to provide a tool (*tool-definition-key*) or a set of tools (*toolset-definition-key*) from the tool crib to a specified work item (*work-keys*). The *tool-definition-key* and *toolset-definition-key* elements may be repeated to allow for the specification of multiple tools or tool sets in one tool order. Construction of a tool assembly may be involved. The definition of the tool or tool set in the tool catalog will identify its components.

tool-order-definition: a section of the tool order specifying the customer, priority rating, due dates, the work to be done, the work item that the tool order serves, the individual tools or tool sets ordered, resources required, etc.

tool-orders: a collection of individual tool orders.

tool-order-status: the overall status of the tool order.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this structure: *reference-keys*, *tool-definition-key*, *toolset-definition-key*, and *work-keys*. Elements within *schedules* use a *tool-order-key* to reference the data elements in the *tool-orders* element.

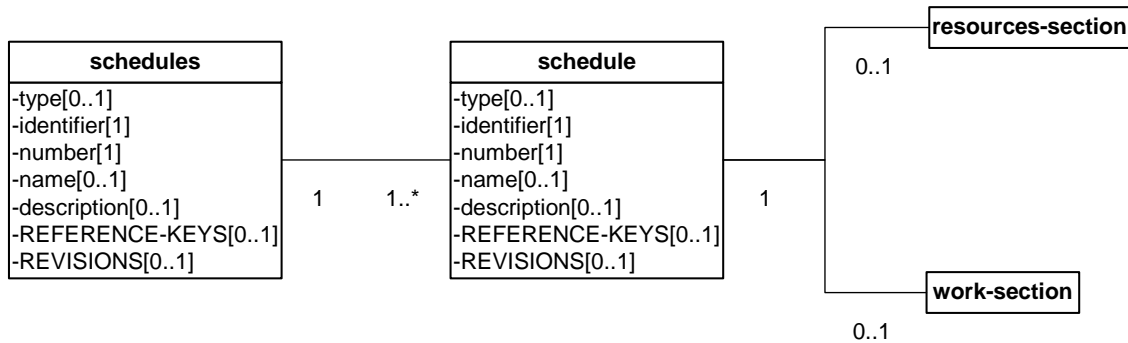
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *cost-summary*, *customers*, *description*, *due-dates*, *identifier*, *name*, *number*, *priority-rating*, *resources-required*, *revisions*, *type*, *work-actual-progress*, and *work-scheduled-progress*.

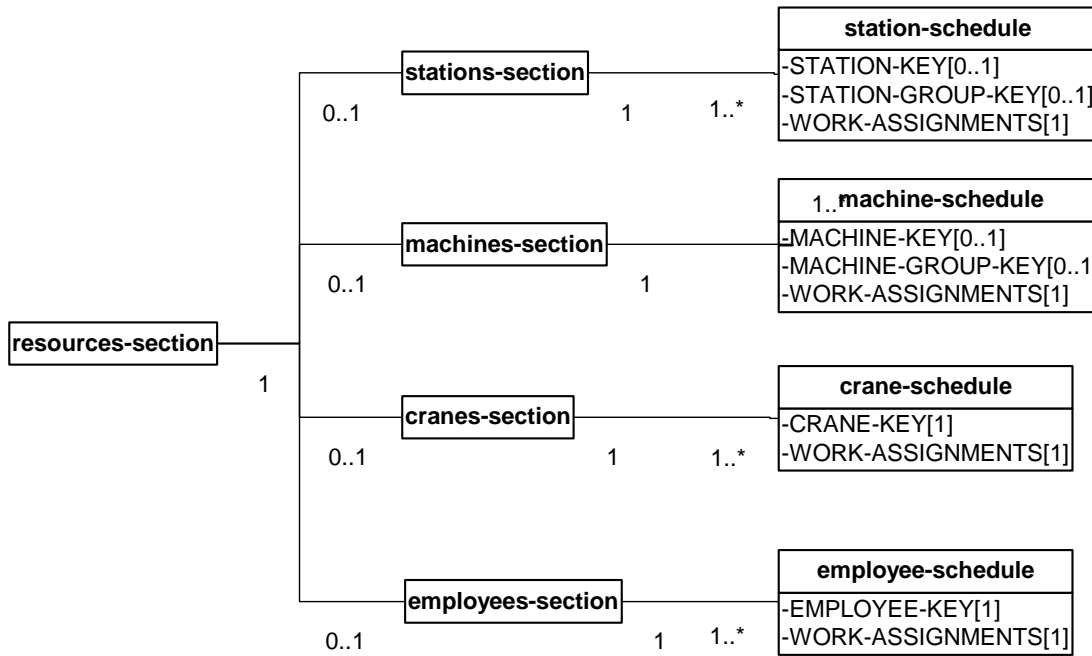
3.36 Schedules

schedules lists planned assignment or mapping of work to resources (*resources-section*) and resources to work (*work-section*). In the *resource-section*, there are sections for each type of resource, e.g., machines, stations, employees. A schedule for a particular resource gives the key for the resource followed by a list of work assignments. In the work section, work is grouped by type, e.g., jobs, tasks, pick orders. A particular work schedule element lists the key for the work followed by a list of all resource assignments.

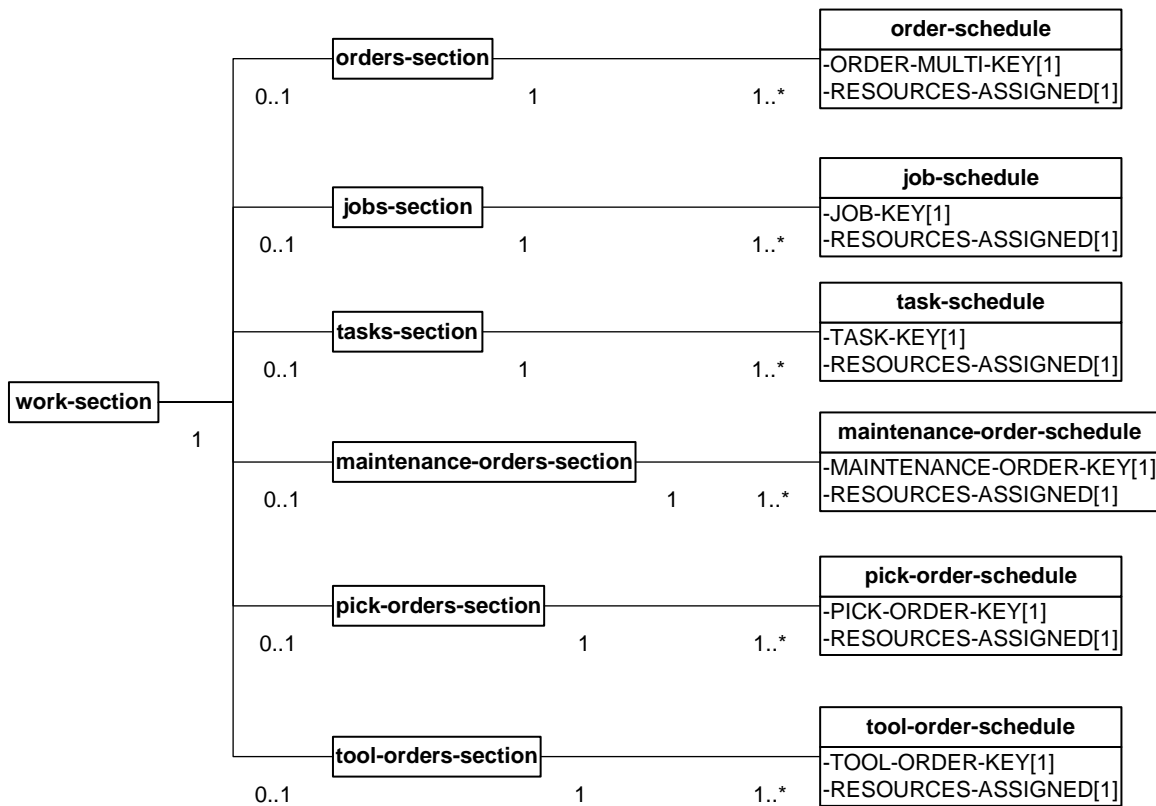
The high level UML information model for the *schedules* element is shown below:



The UML information model for the *resource-section* of *schedule* element is shown below:



The UML information model for the *work-section* of *schedule* element is shown below:



The XML structure for the *schedules* element is shown below:

```
<schedules type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <schedule type="" identifier="" number="" >
    <name />
    <description />
    <reference-keys />
    <revisions />
    <resources-section>
      <stations-section>
        <station-schedule>
          <station-key />
          <station-group-key />
          <work-assignments />
        </station-schedule>
      </stations-section>
      <machines-section>
        <machine-schedule>
          <machine-key />
          <machine-group-key />
          <work-assignments />
        </machine-schedule>
      </machines-section>
      <cranes-section>
        <crane-schedule>
          <crane-key />
          <work-assignments />
        </crane-schedule>
      </cranes-section>
      <employees-section>
        <employee-schedule>
          <employee-key />
          <work-assignments />
        </employee-schedule>
      </employees-section>
    </resources-section>
    <work-section>
      <orders-section>
        <order-schedule>
          <order-multi-key />
          <resources-assigned />
        </order-schedule>
      </orders-section>
      <jobs-section>
        <job-schedule>
          <job-key />
          <resources-assigned />
        </job-schedule>
      </jobs-section>
      <tasks-section>
        <task-schedule>
          <task-key />
        </task-schedule>
      </tasks-section>
    </work-section>
  </schedule>
</schedules>
```

```

        <resources-assigned />
    </task-schedule>
</tasks-section>
<maintenance-orders-section>
    <maintenance-order-schedule>
        <maintenance-order-key />
        <resources-assigned />
    </maintenance-order-schedule>
</maintenance-orders-section>
<pick-orders-section>
    <pick-order-schedule>
        <pick-order-key />
        <resources-assigned />
    </pick-order-schedule>
</pick-orders-section>
<tool-orders-section>
    <tool-order-schedule>
        <tool-order-key />
        <resources-assigned />
    </tool-order-schedule>
</tool-orders-section>
</work-section>
</schedule>
</schedules>

```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

job-schedule: a schedule for an individual job that identifies the job by key and associated resource assignments.

jobs-section: the subsection of the work section where all job schedules are located.

crane-schedule: a schedule for an individual crane that identifies the crane by key and associated work assignments.

cranes-section: the subsection of the resources section where all crane schedules are located.

employee-schedule: a schedule for an individual employee that identifies the employee by key and associated work assignments.

employees-section: the subsection of the resources section where all employee schedules are located.

machine-schedule: a schedule for an individual machine that identifies the machine by key and associated work assignments.

machines-section: the subsection of the resources section where all machine schedules are located.

maintenance-order-schedule: a schedule for an individual maintenance order that identifies the maintenance order by key and associated resource assignments.

maintenance-orders-section: the subsection of the work section where all maintenance order schedules are located.

order-schedule: a schedule for an individual order that identifies the order by key and associated resource assignments.

orders-section: the subsection of the work section where all production order schedules are located.

pick-order-schedule: a schedule for an individual pick order that identifies the pick order by key and associated resource assignments.

pick-orders-section: the subsection of the work section where all pick order schedules are located.

resources-section: the section of a schedule where all resource schedules are located, e.g., machine, station, employee, crane schedules.

schedule: an instance of a schedule containing a section for resource and work schedules.

schedules: a collection of resource and work schedules for the shop.

station-schedule: a schedule for an individual station that identifies the station or station group by key and associated work assignments.

stations-section: the subsection of the resources section where all station schedules are located.

task-schedule: a schedule for an individual task that identifies the task by key and associated resource assignments.

tasks-section: the subsection of the work section where all task schedules are located.

tool-order-schedule: a schedule for an individual tool order that identifies the tool order by key and associated resource assignments.

tool-orders-section: the subsection of the work section where all tool order schedules are located.

work-section: the section of a schedule where all work schedules are located, e.g., order, job, task, maintenance order, pick order, and tool order schedules.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this structure: *job-key*, *crane-key*, *employee-key*, *machine-key*, *maintenance-order-key*, *order-multi-key*, *pick-order-key*, *reference-keys*, *station-key*, *station-group-key*, *task-key*, and *tool-order-key*. Resource keys are used to identify the resource whose work assignments are being scheduled. Work keys are used to identify the order, job, task, etc. whose resource assignments are being scheduled.

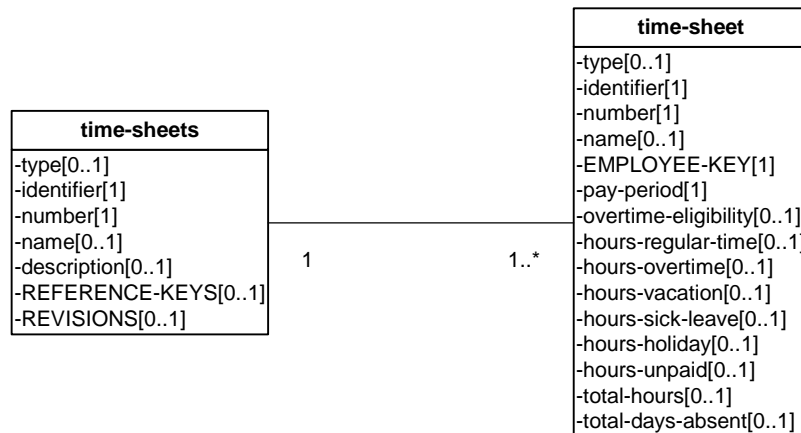
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *identifier*, *name*, *number*, *resources-assigned*, *revisions*, *type*, and *work-assignments*.

3.37 Time Sheets

time-sheets provides a list of individual time sheet elements. A *time-sheet* is used to log the hours that employee works, the time employee takes off from work, and accrual of leave hours.

The UML information model for the *time-sheets* element is shown below:



The XML structure for the *time-sheets* element is shown below:

```

<time-sheets type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <time-sheet type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <employee-key />
    <pay-period />
    <overtime-eligibility />
    <hours-regular-time />
    <hours-overtime />
    <hours-vacation />
    <hours-sick-leave />
    <hours-holiday />
    <hours-unpaid />
    <total-hours />
    <total-days-absent />
  </time-sheet>
</time-sheets>

```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

hours-holiday: hours of holiday time charged in this pay period.

hours-overtime: hours of overtime charged in this pay period.

hours-regular-time: hours of regular time charged in this pay period.

hours-sick-leave: hours of sick leave time charged in this pay period.

hours-unpaid: hours of unpaid time recorded in this pay period.

hours-vacation: hours of vacation time charged in this pay period.

overtime-eligibility: eligibility for overtime pay.

pay-period: the number of the pay period in the current fiscal year.

time-sheet: a time sheet for an individual employee indicated by an employee key.

time-sheets: a collection of time sheets.

total-days-absent: total days absent for sick leave during the current fiscal year.

total-hours: total hours charged this pay period all work and leave categories.

b) Data Element Keys, Multi-keys, and Key Lists

The following keys are used in this structure: *employee-key* and *reference-keys*.

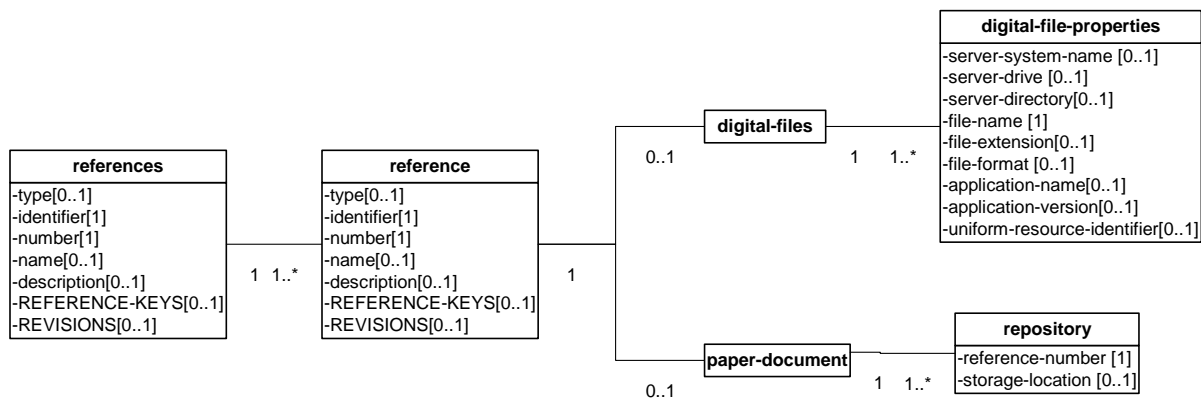
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *description*, *identifier*, *name*, *number*, *revisions*, and *type*.

3.38 References

references describes the information about the reference materials that support or further define the data elements contained within the shop data structure. A reference material may be in a digital or paper form. *references* is defined by a *type*, an *identifier*, a *number*, a *name*, a *description*, a *revisions*, and one or more *reference* data elements. The data associated with *reference* are the following: *identifier*, *number*, *name*, *revisions*, *digital-files*, and *paper-document*. *digital-files* specifies the reference that is stored in electronic form. *paper-document* specifies the reference that is stored in paper form.

The UML information model for the *references* element is shown below:



The XML structure for the *references* element is shown below:

```
<references type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <reference type="" identifier="" number="">
    <name />
    <description />
    <reference-keys />
    <revisions />
    <digital-files>
      <digital-file-properties>
        <server-system-name />
        <server-drive />
        <server-directory />
        <file-name />
        <file-extension />
        <file-format />
        <application-name />
        <application-version />
        <uniform-resource-identifier />
      </digital-file-properties>
    </digital-files>
    <paper-document>
      <repository>
        <reference-number />
        <storage-location />
      </repository>
    </paper-document>
  </reference>
</references>
```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

application-name: the name of the application system that used to save digital document.

application-version: the version of the application system that used to save digital document.

digital-file-properties: information describes how and where a digital file is stored.

digital-files: information about a reference that is stored in electronic form. It is presented by a *repository* data element.

file-extension: the file extension used by the digital document.

file-format: the file format used by the digital document.

file-name: the file name that the digital document stores.

paper-document: information about a reference that is stored in paper form. It is presented by a *repository* data element

reference-number: the reference number that used to keep track the paper document.

references: a collection of one or more references.

repository: a mechanism that gathers information about the resource, format, or location of a reference. It is represented by the following data elements: *server-system-name*, *server-drive*, *server-directory*, *file-name*, *file-format*, *application-name*, *application-version*, and *uniform-resource-indicator* if the reference is a digital document, or the following data elements: *reference-number*, and *storage-location* if the reference is a paper document.

server-directory: the name of the server directory that the digital document stores.

server-drive: digital-document: the server drive that the digital document stores.

server-system-name: the name of the server system that the digital document stores.

storage-location: the paper document's storage location.

uniform-resource-identifier: the uniform resource indicator (URI) that the digital document resides.

b) Data Element Keys, Multi-keys, and Key Lists

The *reference-keys* element is used in this structure.

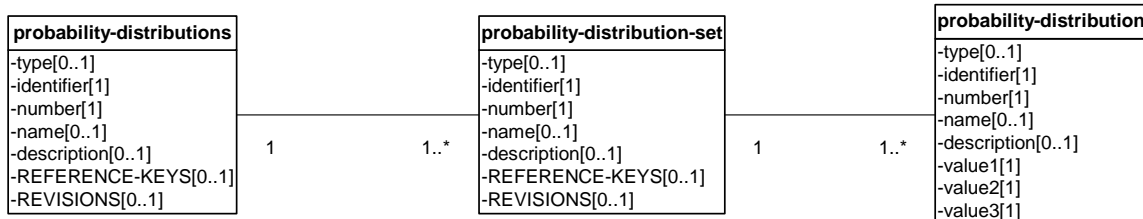
c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *identifier*, *type*, *number*, *name*, *description*, and *revisions*.

3.39 Probability Distributions

probability-distributions specifies distributions that are used to vary processing times, breakdown and repair times, availability of resources, etc. Distributions may be referenced from elsewhere using the key-structure, or may be specified within the structure by providing the distribution coefficient values. It is represented by the following data elements: *type*, *identifier*, *number*, *name*, *description*, and *probability-distribution-set*.

The UML information model for the *probability-distributions* element is shown below:



The XML structure for the *probability-distributions* element is shown below:

```
<probability-distributions type="" identifier="" number="">
```

```

<name />
<description />
<reference-keys />
<revisions />
<probability-distribution-set type="" identifier="" number="">
  <name />
  <description />
  <reference-keys />
  <revisions />
  <probability-distribution type="" identifier="" number="">
    <name />
    <description />
    <value1 />
    <value2 />
    <value3 />
  </probability-distribution>
</probability-distribution-set>
</probability-distributions>

```

The unique data elements, data element keys, and common data elements introduced in this structure are described below.

a) Unique Data Elements

probability-distribution: a distribution that is used to vary processing times, breakdown and repair times, availability of resources, etc. It is represented by the following data elements: *type*, *identifier*, *number*, *name*, *description*, *value1*, *value2*, and *value3*; *value1*, *value2* and *value3* are distribution coefficients.

probability-distributions: a collection of one or more probability distributions.

probability-distribution-set: a collection of probability-distribution data elements. It is represented by the following data elements: *type*, *identifier*, *number*, *name*, *description*, *probability-distribution*, and *reference-keys*.

b) Data Element Keys, Multi-keys, and Key Lists

The *reference-keys* element is used in this structure. Elements within *estimated-durations*, through the *duration-distributions element*, use a *probability-distribution-key* to reference the data elements in the *probability-distributions* element.

c) Common Data Elements

See Sections 2.2 and 2.3 for a discussion of these common data attributes and elements used by this element: *type*, *identifier*, *number*, *name*, *description*, *revisions*, *type*, *value1*, *value2*, and *value3*.

4 REFERENCES

- [ISO 1994] “ISO 10303-1, Part 1: Overview and Fundamental Principles. Industrial Automation Systems and Integration - Product Data Representation and Exchange,” ISO 10303, International Organization for Standardization, Geneva, Switzerland, 1994.
- [Kemmerer 1997] Kemmerer, Sharon, and Fowler, James E., “Initial Manufacturing Exchange Specification (IMES), IMES Concept Document For Manufacturing Systems Integration,” NISTIR 5978, National Institute of Standards and Technology, Gaithersburg, MD, 1997.
- [NIST 2003] National Institute of Standards and Technology (NIST), Systems Integration for Manufacturing Applications, 2003, Available online via <http://www.mel.nist.gov/msid/sima/sima.htm>
- [OMG 2003] Object Management Group (OMG), Unified Modeling Language, 2003, Available online via <http://www.omg.org/uml/>
- [SISO 2005] Simulation Interoperability Standards Organization (SISO), Simulation Interoperability Standards Organization, 2005, Available online via <http://www.sisostds.org/>
- [W3C 2000] World Wide Web Consortium (W3C), Extensible Markup Language (XML) 1.0 (second edition), 2000, Available online via <http://www.w3.org/TR/REC-xml.html>

APPENDIX A – DATA ELEMENT ENUMERATIONS

Data Element	Source	Enumeration Items
behavior	layout	TBD
controller-type	machine-program , machines	TBD
day-name	calendars	[SUN, MON, TUE, WED, THR, FRI, SAT]
day-number	calendars	[0, 1, 2, 3, 4, 5, 6]
efficiency-rating	machines	TBD
employment-status	employees	[FULL-TIME, PART-TIME, CONTRACTOR, LAID-OFF, FORMER-EMPLOYEE]
file-format	references	[WORD, TEXT,]
logical-relationship	child-work-items, plan-definition, precedent-constraints	[TRUE, FALSE, UNKNOWN]
maintenance-state	cranes, machines, stations	[COMPLETED, UNDER-REPAIR, TO-BE-REPAIR, UNKNOWN]
material-type	materials-inventory	
operational-state	cranes, machines, stations	[BUSY, IDLE, RESERVED, OUT-OF-ORDER]
operation-subtype	operation-definitions	TBD
operation-type	operation-definitions	TBD
overtime-eligibility	employees, time-sheets	[YES, NO]
priority-rating	departments, organizations	[1, 2, 3, 4, 5]
priority-rating	procurements	[REGULAR, URGENT]
priority-rating	work	[HOT, HOT-HOT, REGULAR]
salary-type	employees	TBD
status-code	inventory, work-actual-progress, work-scheduled-progress	[EXISTING, NOT-EXISTING, IN-ORDER, PENDING, ACCEPTED, RELEASED, IN-PROCESS, COMPLETE, CANCELLED]
units (angular-units)	units-of-measurement	[DEGREE, RADIAN]
units (currency-units)	units-of-measurement	[DOLLARS, EUROS, YENS]
units (distance-units)	units-of-measurement	[MILE, FOOT, INCH, KILOMETER, METER, CENTIMETER, YARD]
units (length-units)	units-of-measurement	[MILE, FOOT, INCH, KILOMETER, METER, CENTIMETER]
units (mass-units)	units-of-measurement	[TON, POUND, OUNCE, KILOGRAM, GRAM]
units (speed-units)	units-of-measurement	[MILES/HOUR, KILOMETERS/HOUR]
units (time-duration-units)	units-of-measurement	[YEARS, WEEKS, DAYS, HOURS, MINUTES, SECONDS]
units (volume-units)	units-of-measurement	[OUNCE, MILLILITER, LITER, GALLON, CUBICFEET, CUBICMETER, CUBICINCH, CUBICCENTIMETER]

APPENDIX B – DATA ELEMENT FORMATS

Data Element	Format
capacity	REAL
coefficient	REAL
cost	REAL / d d d d d d d d . c c (refer to units-of-measurements)
date	yyyy-mm-dd
duration	REAL / d d d . c c (refer to units-of-measurements)
horsepower	REAL
periodicity	REAL
time	hh:mm:ss
value (of parameter)	REAL