An Analysis of Requirements for Specifying Manufacturing Engineering and Business Processes

Amy Knutilla
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
220/A127
Gaithersburg, MD 20899

Craig Schlenoff
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
220/A127
Gaithersburg, MD 20899

Steven R. Ray
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
220/A127
Gaithersburg, MD 20899

Abstract

A wide range of manufacturing software applications deal with the manipulation and expression of collections of activities. Examples include manufacturing process planning, production scheduling, simulation, project management, workflow, business process reengineering, and product realization process modeling. While each of these applications serves a specific audience and need and focuses on particular aspects of a process, much could be gained by sharing process information among applications. One of the primary obstacles to such integration is the lack of any common representation of what is really the underlying concept of process. The objective of the work described here is to investigate the feasibility of a unifying specification of process that is applicable to all of the above manufacturing applications, yet powerful and robust enough to meet each set of requirements. The results of the first phase of the work – that of researching the process specification requirements for design/manufacturing process life-cycle applications – are described and analyzed. Alternative views of the process specification requirements provide the ability to better understand them, to ensure their completeness, and to envision the structure and approach of a future, generic process specification language (PSL). Task, resource, product, and time are identified and analyzed as the fundamental aspects of process, offering insights to understanding, analyzing, and improving manufacturing and business processes.

1.0   Introduction

The motivation for this work is to move closer to the goal of integrated manufacturing engineering and production systems. This goal has been elusive, despite the great strides taken in information technology in the past few decades. Achieving integration requires at least compatibility of data representations, communication paradigms, and system architectures. Advances have been made in each of these areas, such as product data exchange, e.g., ISO 10303 (ISO-a 1995), communication protocols, e.g., TCP/IP (Arnett *et al.* 1995), OSI[1], and architectures, e.g., Computer Integrated Manufacturing (CIM) Framework (Eng et al. 1996), Common Object Request Broker Architecture (CORBA) (Seigel 1996).

One area of data representation that has received relatively little attention, when compared to product data, is process data. This is particularly interesting when one considers that most aspects of a manufacturing enterprise involve some form of process. Just as significant is that each manufacturing function dealing with process typically has its own representational approach, i.e., its own means of describing, manipulating and representing processes, resulting in difficulties when attempting to share process information among applications. An approach to sharing process information would be to have a process specification language (PSL) that is common to all manufacturing applications, generic enough to be decoupled from any given application, and robust enough to be able to represent the necessary process information for any given application. Additionally, this PSL would be sufficiently well defined to enable exchange of process information among established applications. This PSL would facilitate communication between the various applications because they would all "speak the same language", either as their "native" language or a "second" language, for exchange.

To build a truly common process specification language, those process characteristics that are common among process-centered applications must be identified. At the National Institute of Standards and Technology (NIST), the Process Specification Language project was designed to develop this language, with the first step being the identification of requirements for specifying process. While the resulting set of requirements is intended to provide a strong foundation for the development of a generic PSL, it should also stand on its own as a comprehensive list of process characteristics that can be used by developers of process ontologies, simulators, or other related application development efforts.

Section 2 is a description of the PSL project, providing a context for the analysis of process specification requirements. Section 3 is an overview of the types of requirements identified in the PSL project. Section 4 is an analysis of the process specification requirements from multiple perspectives. Section 5 is a discussion of both the significance of the three different perspectives and the fundamental elements of process. Section 6 concludes the paper.

2.0 The Process Specification Language (PSL) Project

2.1   Technical Approach

[1] Bay Networks Open System Interconnect (OSI), http://bbs-koi.uniinc.msk.ru/product/bay/routers/protocol/bridge/osi.htm, July 25, 1996.

The approach taken for the PSL project has been to break it into distinct phases: requirements gathering, existing process representation analysis, semantic concepts definition, language grammar/syntax/presentation development, pilot implementation and validation, and finally, submission as a candidate international standard. In the first phase, requirements for specifying process are identified. (This phase is further elaborated in this paper.) The intent of the representation analysis is to determine how well existing process representation methodologies support the requirements found in phase 1. This analysis will provide an objective basis from which to develop a comprehensive language and will promote the leveraging of existing work. The language's semantic concepts, grammar, syntax, and presentation(s) will be developed, building on the project's previous results. By this time, a suitable scenario or group of scenarios will have been identified for a prototype implementation to ensure the completeness of the process specification language. It is this validated, documented language that will be submitted to appropriate organizations as a candidate standard. Feedback and consensus from the process community has been and will continue to be aggressively pursued during all phases of the project.

2.2 Gathering Process Specification Requirements

The objective of the first phase of the PSL project was to obtain a comprehensive set of requirements for specifying processes and, from this, to determine the feasibility of a common language for specifying the processes used in manufacturing engineering and business software applications. The first step was to explore a cross section of manufacturing applications that use process information and to determine the process specification requirements for each. The applications explored were manufacturing process planning, production scheduling, simulation, workflow, business process re-engineering, project management, and product realization process modeling. This was not meant to be an exhaustive list of applications that use manufacturing process information, but rather to serve as a good sampling to determine a comprehensive set of requirements necessary to represent process information.

The identification of process characteristics was primarily accomplished through a literature survey of the manufacturing process applications and by interviewing application experts. Additional process specification requirements were identified as a result of a literature survey and interviews with researchers who have gathered similar requirements to represent process but limited their scope to particular industries. Finally, existing software packages, modeling languages, architectures, and standards were examined. Only a subset of these was selected due to limited resources and time constraints. Some of the efforts explored were: ALPS (A Language for Process Specification) (Catron and Ray 1991); MOSES (Model Oriented Simultaneous Engineering Systems) (Molina *et al*. 1994); BPFL (Berkeley's Process Flow Language), CS/CAPP[TM] process planning system (CS/CAPP 1994); AP 213 (Application Protocol in STEP: Numerical Control process plans for machined parts) (ISO-b 1995); MetCAPP[TM] (MetCAPP 1994); Workflow Management Coalition (WfMC) specifications (WfMC Members 1994); Knowledge Interchange Format (KIF) (Genesereth and Fikes 1996); Microsoft Project [TM]

(Microsoft 1994), Process Interchange Format and Framework (PIF) (Lee *et al*. 1996); and SDEF (Standard Data Exchange Format) (East 1995). [2]

2.3   Collaboration

To ensure a comprehensive set of process specification requirements, active participation of practitioners and researchers has been encouraged. Several mechanisms were established to allow colleagues to be kept informed and to actively participate in the PSL project. The first was the creation of project information pages made available through the World Wide Web (http://www.nist.gov/psl/). These pages were regularly updated to reflect on-going status. Linked to these pages are feedback forms to allow readers to provide comments and suggestions. The second mechanism was the creation of two electronic mailing lists set up for the purpose of informing colleagues about project status in a more pro-active fashion. The first list was created to allow one way communication from the project participants to interested parties for notification of major milestone completion with pointers to relevant information on the web pages. This growing list currently includes over 200 participants. The second list, with over 70 participants, was created to allow for discussion among the project participants and the external community. Finally, an extension of the web pages was created to maintain the results of research and analysis conducted by an extended PSL team. Active project participants had access to password-protected pages both to view early results and to contribute ideas to the project. Through aggressive communication and feedback, the scope and direction of the project has been and will continue to be tailored to better match the expectations and desires of the community.

3.0   Examples of Requirements for Process Specification

Many types of process characteristics were identified during the analysis of manufacturing engineering and business applications. Some were clearly associated with specifying processes, and appeared as key characteristics in all process models, e.g., task, resource, sequence. For example, to specify a process for sweeping a floor, a language must be able to characterize the task (sweep), resource (broom), and sequence (first get the broom, then sweep). Other requirements were associated with the applications, for example, manufacturing simulation applications used for process analysis require the ability to model stochastic characteristics. Some characteristics were not for process specification at all, but rather were generic language requirements, like the need for an extensible language and good documentation. The full set of requirements is available both from NIST and on-line (Schlenoff *et al*. 1996).

3.1   Generic language requirements for a PSL

Some of the identified requirements were not directly needed to represent manufacturing processes but were more applicable to the language in general. These requirements provide additional guidelines for the PSL development.

---

[2] No approval or endorsement of any commercial product in this paper by the National Institute of Standards and Technology is intended or implied. This paper was prepared by United States Government employees as part of their official duties and is, therefore, a work of the U.S. Government and not subject to copyright.

The language must:
* be extensible - there must be a mechanism in place to allow a user to add additional information to the pre-defined data constructs. One such mechanism could be the addition of stubs for user-defined information.
* support inheritance.
* use derived attributes to minimize redundant information.
* support multiple views of data.
* allow an instance of the language to be human-readable and computer-interpretable.
* have constructs in place to easily access outside information, including information from other databases.
* be able to handle binding and scoping of variables.
* be able to perform mathematical and logical operations. This could help to derive values from existing data, help with decision-making, and facilitate branching (AND/OR splits, etc.).
* work equally well for simple and complex circumstances - this language must provide easy use for those who wish to only include a high-level non-detailed representation of their process, yet it must be robust and complete enough to allow for those who wish to provide a more detailed representation of their process.
* allow for templates of a task or process. These templates might include the types of information that a process needs to know but not the values that go along with them. It is the user's or program's responsibility to determine and populate the values. Templates could provide a good mechanism for re-using process elements.
* be easy-to-use and have good documentation.

4.0   Understanding the Requirements from Multiple Perspectives

Several alternatives for categorizing the process specification requirements have been proposed:

* to improve understanding of the requirements,
* to aid in establishing the basis for the PSL format,
* to identify omissions,
* to improve readability, and
* to reveal fundamental process concepts.

The first approach to categorizing the process specification requirements is from the perspective of manufacturing engineering and business process applications in the manufacturing domain. Process specification requirements for various process-centered manufacturing applications were identified. The second approach views the process specification requirements in terms of how they relate to one another. The third approach is from the perspective of existing methodologies for specifying processes, identifying clusters of process specification requirements that lend themselves to common constructs and features of existing languages, standards, and methodologies.

4.1   Domain Perspective

Requirements for the process specification language were initially defined and grouped into four major categories. A full list of requirements and additional discussion can be found at (Schlenoff *et al*. 1996). Definitions and examples of these categories are described below.

- **Core:** the most basic, essential requirements inherent to all processes. To represent process, it is either critical that these requirements be included, or these requirements are so common that every application either explicitly or implicitly uses them. While all processes contain core requirements, the core requirements provide the basis for representing only the simplest of processes (e.g., task, duration).
- **Outer Core**: the pervasive, but not essential, requirements for describing processes common to most applications (e.g., temporal constraints, resource grouping, and alternative tasks).
- **Extensions**: the groupings of related requirements, common to some, but not all, applications that together provide an added functionality. Although the requirements listed within the extensions are not inherently necessary for representing process, they may be useful during implementation to provide their respective functionality. They are included here because the PSL must be able to represent information that will ultimately allow this functionality. Six extensions have been defined, each described below.
  1. Administrative/Business - requirements supporting the description of the policies, priorities, and rules of a company that could affect processes. Concepts such as business practices, manufacturing rules, design reviews, and go/no-go decisions would be supported here.
  2. Planning/Scheduling/Quality/Analysis - requirements supporting "what if" analysis, quality checking, and process performance measurements.
  3. Real-Time/Dynamic - requirements supporting real-time monitoring of a process, such as process/resource status and states.
  4. Process Intent - requirements supporting the description of the functions and goals of processes, such as process purpose and decision rationale.
  5. Aggregate Resources/Processes - requirements supporting the description of the characteristics of a group of resources. Concepts such as parallelism and implicit resource association are included here.
  6. Stochastic/Statistics - requirements supporting the description of the random or probabilistic aspects of a process. Concepts such as probabilities of down time and performance variability are included here.
- **Application Specific**: the requirements only relevant within specific applications (e.g., dynamic rescheduling for the production scheduling application).

This study identified sixteen core requirements, 41 Outer Core requirements, 41 Extension requirements, and 12 Application requirements. The Core requirements are listed here – the ordering does not imply any priority. (Schlenoff *et al*. 1996)

Core Requirements for Specifying Process

- product (work item, artifact) - intermediate and final output of tasks (and processes).
- product characteristics - information about an intermediate and final product which a task will produce.
- resource (participant, agent) - a single resource or a group of resources. Some types of resources are equipment, people, information, and in-progress goods.

- simple resource capability/characteristics - a high-level description of the characteristics of a resource.
- resource requirement(s) for a task - the relationship between one or more resources and a task.
- task (action, activity, process) representation - one or more tasks, actions, activities or processes.
- simple task characteristics - a simple, high-level description of the task.
- simple groups of tasks - very basic, high-level set of tasks. One example is the grouping of tasks and sequences that make up a process plan or that make up a project phase.
- simple sequences - linear, time-sequential groups of tasks. More sophisticated relationships such as parallel and alternative tasks can be found in the outer core.
- simple precedence - a high-level description of the precedence constraints of one task on another. A more detailed description of precedence and constraints can be found in the outer core.
- task duration - the time required to complete a task or group of tasks.
- cost data - the cost associated with a resource or task. This could be a fixed cost, cost rate, or a cost derived from other attributes such as duration and level of effort. Costs associated with uncertainty, variability, tolerances, etc. could also be included in this requirement.
- level of effort - description of the amount of a resource needed, in any given unit, to accomplish a task. Some example levels of effort are equipment-hour, labor-hour, and crew size.
- task executor - who is responsible for executing a task or group of tasks. Examples include a person, controller, or external company if the task is contracted out.
- resource allocation/deallocation for one or many tasks - the assignment and release of one or more resources to a task or group of tasks.
- ad hoc notes and annotations optionally associated with any component of a plan - on-the-fly, off-the-cuff notes, and documentation used to specify process. This could be voice, video, as well as text.

4.2   Objects/Characteristics/Relationships Perspective

The Core and Outer Core requirements could be stated more clearly as primitive concepts (what are the basic types of things that people talk about when modeling or describing process?), characteristics intrinsic to a concept (how do people characterize their processes, resources, products, etc.?), and relationships among these concepts.[3] There are on-going discussions to determine the "primitive" concepts. Let us propose the basic or primitive concepts as:

- task (action, process, activity)
- resource (participant, agent)
- product (work item, artifact)
- time

For the most part, the requirements for specifying process found in the Core and Outer Core can be assigned to one or more of these three categories; primitives, characteristics of primitive concepts, and relationship among primitive concepts. The resulting table (Table 1) revealed

---

[3] Electronic mail communication with Florence Tissot, Knowledge Based Systems, Inc. (http://www.kbsi.com) October 1996.

certain omissions, such as the lack of a "task representation" requirement, and improved the visualization and understanding of the requirements.

The "Primitive Concept" column indicates the primitive requirements, the "Characteristic of Primitive Concepts" column shows which primitive concept(s) these type of requirements characterize, and the column for "Relationship among Primitive Concepts" indicates the two or more primitive concepts involved in the relationship implied by the requirement. For example, the requirement to be able to specify "sequences" is characterized as a relationship among multiple tasks and some element of time (such as "before"), therefore, the relationship for sequences is stated as "task/task/time".

Table 1: Core and Outer Core Requirements for Specifying Process

Requirements	Primitive Concept	Characteristic of Primitive Concept(s)	Relationship among Primitive Concepts
task	task		
task characteristics		task	
manual vs automated tasks		task	
conditional tasks		task	
highlights, milestones		task	
parameters and variables		task	
dispatching		task	
task grouping		task	task/task
simultaneously maintained levels of abstraction		task	task/task
alternative task			task/task
parallel tasks			task/task
serial tasks			
concurrent tasks			
ancestry or class of a task			task/task
information exchange between tasks			task/task
iterative loops			task/task
decomposition/composition			task/task
exception handling/recovery			task/task
precedence			task/task
sequences			task/task
synchronization of task sequences			task/task/time
resource			task/task/time
resource characteristics			task/task/time
resource location	resource		
queues, stacks, lists		resource	
resource categorization and grouping		resource	
implicit/explicit resource association		resource	
time		resource	resource/resource
clock time			resource/resource
task duration	time		
product		time	
product characteristics		task, time	task/time
manufacturing product quantity	product		

Table 1: Core and Outer Core Requirements for Specifying Process

Requirements	Primitive Concept	Characteristic of Primitive Concept(s)	Relationship among Primitive Concepts
constraints		product	
ad hoc notes		product	
incompleteness/vagueness		task, resource, product, time	
confidence levels		task, resource, product, time	
uncertainty, variability, tolerance		task, resource, product, time	
associated illustrations/drawings		task, resource, time	
state representations		task, resource, time	
cost data		task, resource, product	
deadline management		task, resource, product	
task executor		resource, task	
eligible resources			task/time
level of effort			resource/task
resource allocation/ de-allocation			resource/task
resource requirements for a task			resource/task
resource/task combined characteristic			resource/task
resource/product combined characteristic			resource/product
product/task combined characteristic			product/task

4.3   Clustering of Requirements

The original set of requirements was used to analyze existing process representations in the second phase of the PSL project. Over a dozen experts in the process representation community participated in completing a matrix (on-line via the World Wide Web[4]) that evaluates twenty-six process representations with respect to these requirements. For this analysis, experts were asked to rate the representation's ability to address individual requirements, and then to describe the constructs and features used to address each requirement. Once this matrix was complete, further analysis involved understanding the types of approaches used to address requirements. What emerged from this study was yet a third means of clustering the requirements that was used to contrast and compare the twenty-six approaches. It was found that similar methods and constructs were used to represent clusters of requirements. For example, one type of representation would treat both resource and product as OBJECT that is related to ACTIVITY, either as a "is modified by" relationship in the case of product, or "is consumed by" relationship in the case of resource. This clustering is shown below.

Process Specification Requirements Clusters

Task/Process Representation and Basic Characteristics
- task representation
- task characteristics
- grouping of tasks

Resource Representation and Basic Characteristics
- resource representation
- resource capability and characteristics
- product representation
- product characteristics
- grouping of resources

Resource/Task Combined Characteristics and Associations
- task executor
- resource requirements for a task
- level of effort
- cost data
- ad hoc notes

Constraints
- pre- and post-processing constraints
- state existence constraints
- temporal constraints

Sequences/Precedence
- simple precedence
- simple sequences
- alternative tasks
- concurrent tasks
- iterative loops
- parallel tasks
- serial tasks

Time
- date and time
- task duration

The complete results of the analysis of process representations can be found at (Knutilla *et al.*, 1998, Polyak and Tate, 1997).

---

[4] http://www.nist.gov/psl/

5.0   Discussion

5.1   Viewing requirements from different perspectives

The requirements for specifying processes or characteristics of processes have been presented from three different perspectives. The original was from the domain perspective--what are the requirements for specifying different types of manufacturing engineering and business processes? This analysis revealed that there exist common process characteristics among all domains. For example, even though processes for manufacturing process planning are quite different from business process reengineering, there exists a priority (i.e., core, outer core, extension, application) of requirements, common among all domains.

The second approach is a structured analysis of the requirements that acknowledges that within the core and outer core there are different types of requirements--some requirements can be considered primitive while the others are characteristics of and/or relationships among the primitives. This approach may contribute to describing the semantic concepts of the PSL.

The third approach for presenting and understanding the requirements was from the perspective of existing methods used to represent processes. The analysis of a wide variety of representational methodologies identified clusters of requirements that were represented using similar constructs within individual methodologies. This also aids in identifying and defining concepts as well as the syntax and notation for the PSL.

5.2   Identifying fundamental process characteristics

The first approach to categorizing requirements identified sixteen core requirements which were common among all manufacturing applications researched. Among these core requirements were the primitive concepts identified in the second approach: task, product, resource, and time (duration).

To explore this further, let us consider these four concepts as fundamental characteristics of process, i.e., all processes can be described in terms of task, time, resource, and/or product. Note how analyzing combinations of two of these while "freezing" the third, i.e., considering a single "value" of the third characteristic, for all values of the fourth, offers differing perspectives useful in various domains. For example, here are some combinations:

- A view of Task versus Time, with Resource frozen, for all Products could be a schedule for an individual resource. See Figure 1 for an example of three tasks that can be accomplished by the resource called Milling Machine A, over time, for all products A, B, and C that are manufactured in the facility. The result is a schedule for this particular resource, showing when Milling Machine A is milling (and which Product it is milling), when it is undergoing maintenance, and when it is idle.
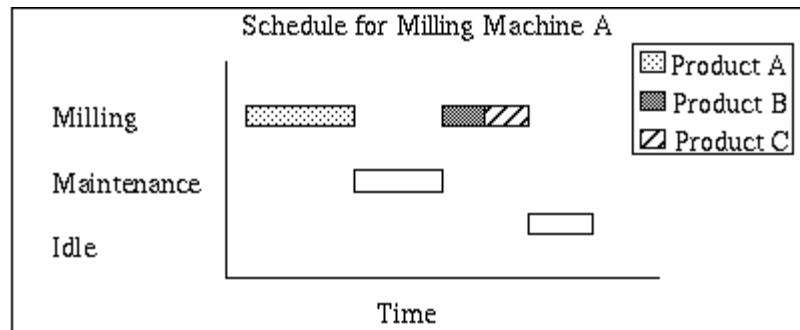
Figure 1: Task versus Time for a single Resource for all Products

- A view of Task versus Time, with Product frozen, for all Resources could be a schedule of tasks for a specific product – or project, to be used for project planning purposes. In Figure 2, the tasks are activities required for a project (the product) over time, and the resources are the people (AK, CS, and SR) assigned to each task.
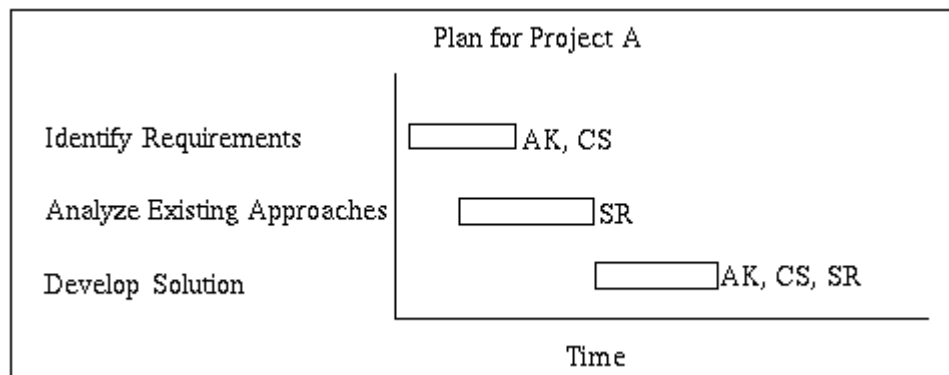
Figure 2: Task versus Time for a single Product for all Resources

- A view of Resource versus Time, with Task frozen, for all Products could indicate time spent by resources on activities of interest, e.g., hazardous activities, maintenance downtime, and union activities. In Figure 3, the single type of activity, maintenance, is reported for resources (Lathe A, Milling Machine B, and Lathe C, for all products produced.
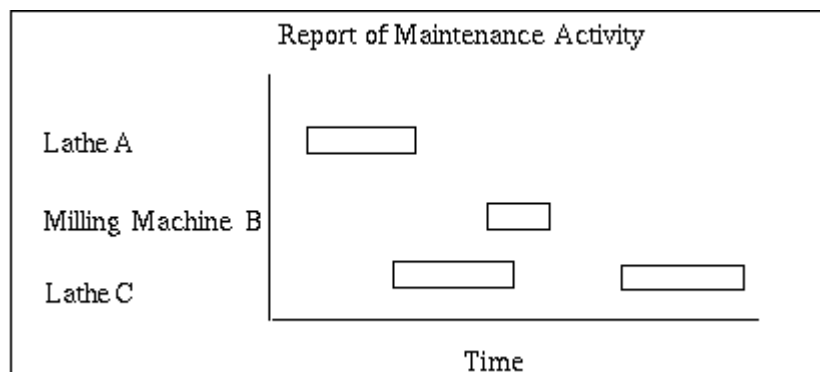
Figure 3: Resource versus Time, for a single Task, for all Products

The previous examples show three views of process information by manipulating the four primitive elements of process, providing information that is useful from a variety of perspectives. Figure 1 could be used for scheduling purposes, Figure 2 for planning purposes, and Figure 3 for reporting purposes.

An interesting view of these four dimensions of process is shown in the following three-dimensional depiction of process for a specific product, i.e., we freeze product. A specific "block" in the process cube represents a specific activity, time, and resource. Activity A, for example, is a reaming task, at time T(i+1), using the resource Machine D.
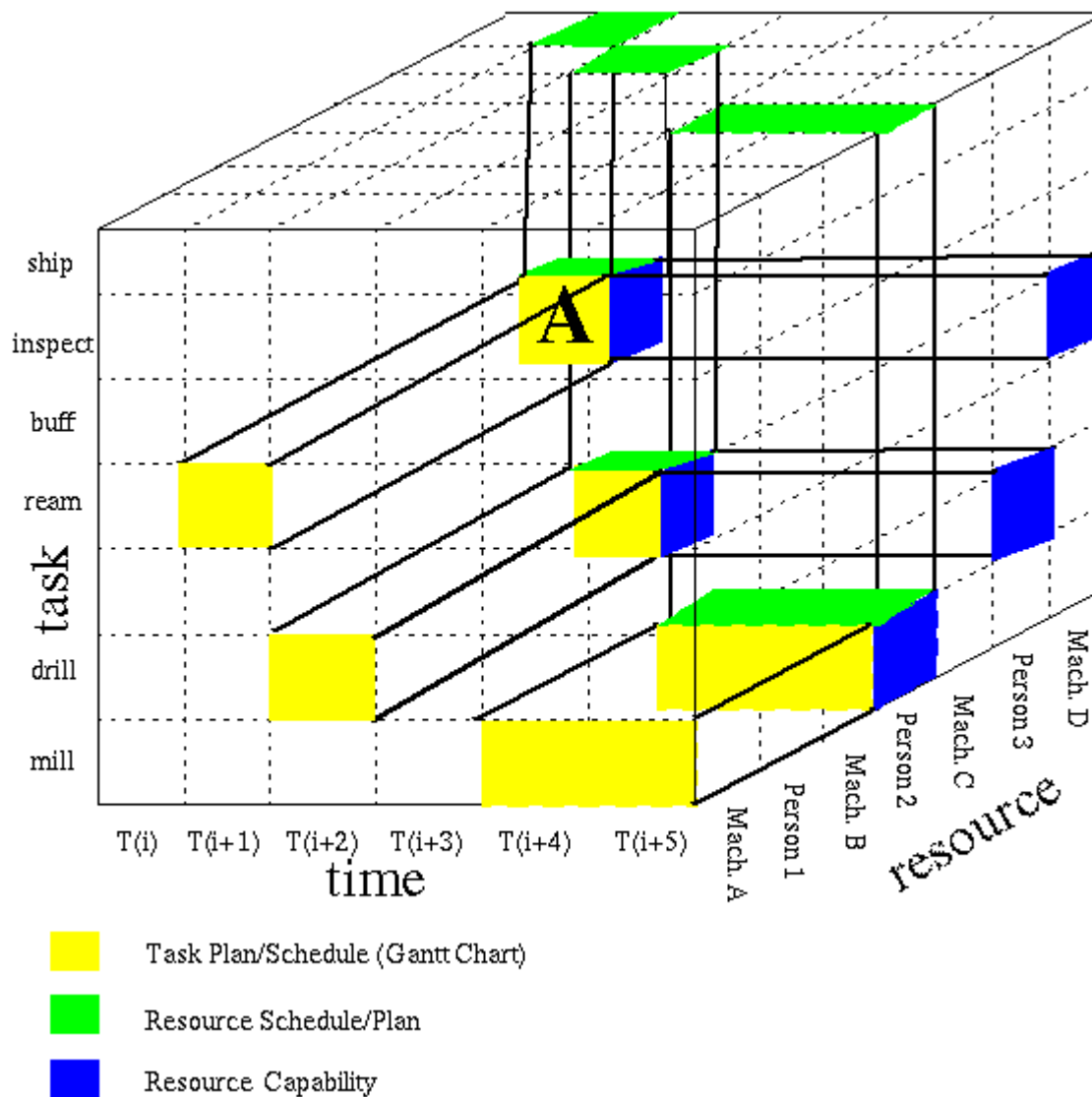

Figure 4: The Process Cube for a specific product

This multi-dimensional analysis seems to support task, time, product, and resource as the fundamental characteristics of process. Furthermore, the charts and reporting mechanisms commonly used in manufacturing and business today can be seen as two dimensional properties of a four dimensional environment. Additional analysis of all possible combinations could reveal interesting, previously unknown or unused, perspectives on process.

6.0   Conclusion

The first phase of the PSL project identified a set of requirements for specifying process under the premise that there exists a generic set of requirements for specifying manufacturing engineering and business processes. The initial results supported this premise, identifying a set of requirements for specifying processes that apply across all domains. Alternative views of these requirements have helped to better understand them, to ensure their completeness, and to envision the structure and approach of the future PSL.

The set of core and outer core requirements defined in this paper has been reviewed by many in the research community and is supported as the basis for defining a process specification language. In fact, related projects are already using the preliminary results of this phase to enhance their research. For example, the PSL requirements were used by the Toronto Virtual Enterprise (TOVE) project (Fox and Gruninger 1994, Gruninger and Fox 1994) to ensure that they had a complete characterization of classes of activities in their activity ontology. The PSL requirements helped them to realize that the definitions of certain classes of activities that were necessary for the ontology were indeed missing, and the activity ontology was extended accordingly. Raytheon used the PSL requirements to validate their preliminary process model created for the Manufacturing Simulation Driver (MSD) (Sands *et al.* 1997) project. The ARPA/Rome Planning Initiative (ARPI) (Tate 1996) has leveraged PSL requirements research to support analysis of their plan representations. (Polyak and Tate 1997)

In conclusion, the identification of requirements for the development of a generic process specification language for information exchange has provided a strong foundation for the PSL research and development and has resulted in a comprehensive set of requirements that has proven useful beyond the PSL research effort. An analysis of the results has revealed interesting and unique observations about the fundamental aspects of process. It is our hope that these observations provoke further insights in understanding, analyzing, and improving manufacturing and business processes.

Acknowledgments

References

ARNETT, MATTHEW FLINT, ET AL., 1995, *Inside TCP/IP, Second Edition*, (New Riders Publishing).

CS/CAPP II Version 6.5 System Manager's Manual, 1994, CIMx.

CATRON, BRYAN A., RAY, STEVEN R., 1991, ALPS: A Language for Process Specification. *International Journal for Computer Integrated Manufacturing* **4**(2).

DAVENPORT, T. H., 1993*, Process Innovation: Reengineering Work Through Information Technology*, (Harvard Business School Press, Boston).

EAST, E. W., 1995, The Standard Data Exchange Format for Critical Path Method Scheduling, US Army Corps of Engineers Construction Engineering Research Laboratories (USACERL) Technical Report-95/40.

ENG, LAWRENCE, ET AL., 1996, Computer Integrated Manufacturing (CIM) Application Framework Specification 1.3, SEMATECH Technology Transfer #93061697F-ENG.

FOX, M.S. AND GRUNINGER*, M.*, 1994, Ontologies for Enterprise Integration, In *Proceedings of the 2nd Conference on Cooperative Information Systems*, Toronto, Ontario.

GENESERETH, M., FIKES, R., 1996, Knowledge Interchange Format Version 3.0 Reference Manual, http://logic.stanford.edu/kif/Hypertext/kif-manual.html, Stanford, CA, May 15, 1996.

GRUNINGER, M., AND FOX, M.S, 1994, The Role of Competency Questions in Enterprise Engineering, *Benchmarking - Theory and Practice* (Chapman).

ISO-a, 1995, Product data representation and exchange: Part 1: Parts Library - Part 1: Overview and Fundamental Principles, ISO Standard 10303-1.

ISO-b, 1995, Product data representation and exchange: Part 213: Application Protocol: Numerical Control process plans for machined parts, ISO Standard 10303-213.

KNUTILLA, A., ET AL, 1998, Process Specification Language: An Analysis of Existing Representations, NISTIR, National Institute of Standards and Technology, Gaithersburg, MD, May 1998. Also available at http://www.mel.nist.gov/msidlibrary/.

LEE, J., ET AL., 1996, The PIF Process Interchange Format and Framework, University of Hawaii, http://soa.cba.hawaii.edu/pif/, May 1996.

Microsoft Project Business Project Planning System, 1994, Version 4.0 User's Guide, Microsoft Corporation.

MetCAPP V4 Process Planning Application, 1994*,* Version 4.1 User's Manual, Institute of Advanced Manufacturing Sciences Inc., Cincinnati, OH.

MOLINA, A., ELLIS, T.I.A., YOUNG, R.I.M., BELL, R., 1994, Methods and Tools for Modelling Manufacturing Information to Support Simultaneous Engineering, In *Proceedings of 2nd IFAC/IFIP/IFORS Workshop, IMS '94*, Vienna, June 1994.

POLYAK, S. T., AND TATE, A., 1997, Analysis of Candidate PSL Process/Plan Representations, AIAI-PR-66, Artificial Intelligence Applications Institute (AIAI), Edinburgh, UK.

SANDS, B., STEPHENS, D., LALIBERTY, T. J., RAJA, N., 1997, Manufacturing Simulations Based on Integrated Product/Process Models, In *Proceedings of DETC'97: 1997 ASME Design Engineering Technical Conference*, DETC97/DFM-4355, Sacramento, CA.

SCHLENOFF, C., KNUTILLA, A., RAY, S., Unified Process Specification Language: Requirements for Modeling Process, NISTIR 5910, National Institute of Standards and Technology, Gaithersburg, MD, September 1996. Also available at http://www.mel.nist.gov/msidlibrary/summary/9660.html.

SIEGEL, JON, 1996, *CORBA Fundamentals and Programming*, ISBN 0471-12148-7 (Wiley).

TATE, A., 1996, *Advanced Planning Technology: Technological Advancements of the ARPA-Rome Laboratory Planning Initiative* (AAAI Press, Menlo Park, CA).

WORKFLOW MANAGEMENT COALITION MEMBERS, 1994, Glossary: A Workflow Management Coalition Specification, Belgium, November 1994.