Proceedings of the Process Specification Language (PSL) Roundtable

Craig Schlenoff, Amy Knutilla, Steven Ray

Abstract

In April, 1997, the Process Specification Language (PSL) Project held a Roundtable discussion at the National Institute of Standards and Technology (NIST). The goals of the Roundtable was to assemble key champions and stakeholders of various approaches towards process representation in order to discuss the relative merits to reach consensus on a language architecture and to establish a technical approach for proceeding. It was agreed that the language architecture should be based upon a formal semantic foundation, upon which would be layered a number of syntactic mappings, each with one or more presentations.

In discussions about principal concepts of any process representation, it was agreed that "process" and "participant (resource)" are basic. A number of possible other concepts were suggested, but no consensus was reached. Additionally, five potential uses for the PSL were identified and discussed. They were: 1) provide a description of a process that has already occurred; 2) provide a "recipe" (prescription) describing how a process can occur; 3) provide a semantic model to determine concepts and establish the scope of systems; 4) enable interoperability between manufacturing systems, enterprise systems, and/or AI systems; 5) enable technology transfer between manufacturing and other disciplines.

Finally, three teams were formed to define:

- A set of scenarios to support the identification and definition of semantic concepts and to provide potential uses of the language;
- A semantic description covering a small subset of the core language requirements;
- Three syntactic interpretations of that semantic description, mapping to object-oriented, KIF, and constraint-based presentations. A relational presentation was also

deemed important, but no assignment was made. Much of this work must wait until an initial set of semantic concepts is determined.

1.0 Introduction

The goal of the NIST Process Specification Language project [1] is to identify or create a process specification language (PSL) that can be 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, the PSL should be sufficiently well-defined to ensure complete and correct 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.

The PSL Project reached a pivotal point in Spring, 1997, with the determination of a representation structure for the PSL. Through extensive research and with feedback from colleagues in industry and academia, a set of requirements for specifying process was defined. The project then completed an analysis of existing process representations with respect to these requirements in order to identify constructs and characteristics of these representations that address the requirements. Based on this analysis, the next step was to identify the fundamental foundation for a Process Specification Language, likely including multiple characteristics and constructs of existing representations.

With the intent to achieve wide understanding and consensus among those in the community who have worked in this area, as well as those who will benefit from standards for process specification, the PSL project sponsored a Roundtable on April 21-22, 1997, at NIST, to discuss the important issues associated with developing the foundation for the PSL. The attendee mix included six representatives from industry, ten from academia, and four from the government.

2.0 Roundtable Format and Agenda

The PSL Roundtable was unique with respect to other similar Roundtable discussions in the way it was formatted. There were actually two distinct portions of the event, the virtual and the physical portion of the Roundtable, where the virtual portion fed directly into the physical portion.

2.1 Virtual Roundtable

The Roundtable began on March 5 with an email discussion among the participants. The purpose of this email discussion was two-fold:

1. To allow participants to introduce themselves to the other participants. By doing this electronically, it allowed us to jump directly into the technical issues at the beginning of the physical Roundtable.
2. To allow participants to introduce issues they would like to see discussed at the physical Roundtable. This discussion allowed the PSL team to create an agenda that was

directly in line with the participants interests while addressing the goal of determining an architecture and technical approach for the specification language.

Issues discussed during the virtual Roundtable included:

- The use of multiple notations, e.g., can a constraint layer lie beneath a directed graph?
- The architecture of the language
- The use of ontologies within the PSL architecture
- The preliminary findings of the existing process representation analysis
- The modeling of authority in the PSL (the right for a resource to act under a set of conditions).

All relevant messages were archived and accessible through the web. In addition, a paper copy of the archived messages was handed out to all participants at the beginning of the physical Roundtable.

2.2 Physical Roundtable

The physical Roundtable began with an overview by Steven Ray, who outlined the meeting objectives, gave a summary of the NIST PSL project, and discussed the need to determine and agree on definitions of language terms to be used at the workshop. The rest of the day focused on a discussion of two topics, each of which was summarized, followed by a discussion and written submission of participants' conclusions.

The first topic, "Constructs and Features for Representing Processes," was facilitated by Amy Knutilla from NIST. The approach and results of the representation analysis phase of the PSL project (Phase 2) [2] were presented and discussed in order to answer the question, "Are the conclusions from this analysis valid?" In this analysis, twenty-six process representation approaches were studied with respect to the requirements identified in the first phase of the PSL project [3]. The constructs and features used to address requirements were identified by people familiar with the representations. The resulting matrix was then analyzed to identify types of constructs and features used to address the requirements for specifying processes. The results of this final analysis were presented by Amy Knutilla, Craig Schlenoff and Rich Anderson.

The second topic, "The Architecture for the PSL" was facilitated by Craig Schlenoff. The goal of this topic was to identify the necessary components of the language and how these components interrelate (i.e. identify the structure of the language). This decision of the structure will determine, to a large extent, the performance and behavior of the language as well as which requirements will be represented. The discussion was started by revisiting the terminology to be used.

Once the terminology was agreed upon, the discussion focused on addressing this topic's questions:

- What will the structure of the language be?
- Can multiple representation features be effectively combined to form the basis for the PSL?
- What role do ontologies play?

- Will the architecture hold for "beyond the core" requirements?

On the second day, a discussion took place involving an attempt to determine the principal concepts constituting a process representation. The goal was to find out what concepts were inherent to any specified process. From these principles concepts, a simplified semantic model was created.

At the conclusion of this exercise, decisions were made on how to proceed and the Roundtable concluded.

3.0 Roundtable Discussion and Results

The Roundtable was broken up into three topics focusing on: the constructs and features necessary for representing process; the architecture of the PSL; and an exercise to create a semantic model of the most basic concepts that underlie process. Each one of these topics are discussed in detail below.

3.1 Topic 1: Constructs and Features for Representing Process

The first topic, "Constructs and Features for Representing Processes," described the approach and results of the representation analysis phase of the PSL project (Phase 2). In this analysis, twenty-six process representation approaches were studied with respect to the requirements identified in the first phase of the PSL project.

To briefly summarize the analysis, similar requirements were grouped into the following six categories [2]:

- Plan/task representation
- Resource representation
- Resource/task characteristics and association
- Sequence/precedence
- Constraints
- Date/time

Within these classifications, the constructs and features of the representational approaches were[2] studied. A set of loosely described approaches was identified which addressed these requirements:

- object-based
- constraint-based
- graph-based
- relationships
- conditions
- inference
- task decomposition

- annotations
- activity-resource matrix
- task-network
- classes (groups)
- timer
- time-intervals/time-points

Most representations used several of these approaches to address requirements. A table was presented which shows the approaches used to address the sets of requirements (see Appendix A). While this table revealed that object-based and constraint-based approaches addressed all classifications of requirements, the table also unintentionally showed a mutual exclusivity among approaches. Foe example. a set of requirements captured using a ‘condition’ construct may also be viewed as being captured by a type of ‘relationship’ construct. It was agreed by all that examples via scenarios would be useful for understanding representation approaches.

It was generally agreed that there is much existing work upon which the PSL can build, as demonstrated by this analysis.

Although the intent of this session was to focus on the constructs and features of existing process representations that could serve as a basis for the PSL, the conversation quickly turned to other subjects that, in the participants’ minds, had to be resolved before this issue could be addressed. These issues were: the anticipated use of the PSL; the definition of a process; the basic concepts necessary for representing process; and how to proceed in creating the language.

3.1.1 The Anticipated Uses of a Process Specification Language

Before one can specify what constructs need to be in a process specification language, it is important to determine the anticipated uses of the language. Throughout the course of this discussion, there were five possible uses identified:

- provide a description of a process that has already occurred (a descriptive language);
- provide a “recipe” describing how a process can occur (a prescriptive language);
- provide solely a semantic model used to define concepts and establish the scope of systems
- enable interoperability between:
  - manufacturing systems
  - enterprise systems
  - AI systems
- enable technology transfer between manufacturing and other disciplines.

After the previous uses were identified, a discussion ensued to determine if the PSL would be primarily used in a prescriptive or descriptive fashion. It was generally agreed that this decision must be made before one could determine the requirements that must go into the language. For example, if the language was meant to be used in a prescriptive fashion, one may want to associate multiple time durations with a given task (estimated time, average time, actual time, etc.). If the language was meant to be used in a descriptive fashion, only the actual time would be of any relevance since the process would have already happened. Given that the primary objective of the PSL is to enable interoperability among process-centered manufacturing systems, it was generally agreed that the basis of the PSL should be descriptive but still have the ability to function in a prescriptive fashion.

3.1.2 The Definition of Process

In order to create a language to specify process, there must be a clear understanding of what a process is. It was generally agreed that if you ask twenty different people to define process, you would get twenty different answers. Chris Menzel[1] describes process as "an objective real world event, described totally as a sequence of events (activities, subprocesses, whatever) occurring over time containing certain objects having certain properties standing in certain relations." Jeffery Herrmann adds "A process can be decomposed into other processes. A process begins and ends at points in time. One can view a process from different perspectives that include different things. Objectives or drivers may be part of one perspective but not another; if included, they could be seen as instructions." While there emerged no commonly accepted definition of process, it was determined that identification of the basic semantic concepts intrinsic to all processes would be useful for defining process.

3.1.3 Basic Concepts Intrinsic to Process Specification

There were actually two separate issues being addressed here. The first was to determine the requirements necessary to specify process, the second was to determine the primitive concepts that are intrinsic to any process specification, independent of why the process is being specified. There seemed to be general agreement that the PSL Requirements Document [3] includes a comprehensive list of requirements necessary to specify process, leaving the need to identify the primitive concepts in which these requirements would be based.

The two concepts that were generally agreed to be primitive were PROCESS (transformation over time) and RESOURCE (or more generally PARTICIPANT). Other concepts for which no consensus was achieved were TIME, PRODUCT, DRIVER (OBJECTIVE), CONTROL, CONTEXT, and ATTRIBUTE (PROPERTY), some of which are described below. Regarding the inclusion of these additional concepts, Dana Nau correctly mentioned that, "there was general agreement, however, the language should not preclude adding them later on."

The concept of TIME met with some debate when proposed as a primitive concept. Some people felt that because TIME was a concept in itself (it existed outside of the realm of process) thus it should not be a primitive concept. Others felt that TIME was an intrinsic enough aspect of process to warrant it as a primitive.

The concept of PRODUCT was a bit controversial since processes can be performed without specifying a physical product (e.g. sweep the floor). Also, because PRODUCT could be seen as a type of RESOURCE, there may be no need to pull it out as a separate

1. For company affiliation of all participants mentioned in this document, please refer to Appendix C.

concept. On the other hand, most processes are performed with the goal of creating a product, hence PRODUCT is intrinsic enough to include as a primitive.

The concept of DRIVER met with some debate. A DRIVER refers to the objective or goal of a given process (i.e. why the process was performed). Essentially everyone at the Roundtable agreed that a DRIVER is an important characteristic of a process but there was disagreement as to whether it is necessary to describe process (which is the goal of the primitives in the PSL). Since a process can be described without specifying why it is performed, some people felt that a DRIVER was not justified as a primitive.

CONTEXT was proposed as a primitive since to characterize a process, it is necessary to understanding the environment in which it occurred. This may be most important in capturing process in a descriptive fashion.

ATTRIBUTES are used to describe characteristics of primitives. One point of view is that it is hard to picture how anyone can describe processes without the ability to further describe the concepts which constitute it. On the other hand, does that mean ATTRIBUTE is a basic concept in itself or is it “something separate?”

3.1.4 How to Proceed

At the conclusion the first topic, the group tried to determine how to proceed. Two suggestions were made. The first suggestion focused on determining the concepts that were needed to be represented in the language before determining the constructs that would represent them. Michael Gruninger summed this up well when he wrote:

> “Rather than prematurely committing to a specific set of terminology, we need to first agree on the set of requirements and the intended interpretations that people have for these requirements. Thus, we do not yet provide a definition for process or resource; we merely say that a process or resource is anything that satisfies their requirements.”

Dana Nau added a caution when he wrote “One basic problem we’ll have to face is whether to make sure we develop all of the formal theory first, or forge ahead with the specification before finishing the formal theory.” Later on he added “If we do the formal theory first, we run the risk of getting bogged down and never getting beyond that -- but if we forge ahead, we run the risk of having problems later on with different people interpreting the specification in different ways.”

The second suggestion was to create the specification language in an iterative fashion. This would involve first laying down the formal theory for some of the most basic concepts within the language, then handling the specification for those concepts, and then continuously expanding and refining these concepts until the specification language is complete.

3.2 Topic 2: The Architecture of the PSL

The goal of the second topic, “The Architecture for the PSL” was to identify the necessary components of the language and how these components interrelate (i.e. identify the structure of the language). Although the conclusions reached during Topic 1 were not those anticipated, enough consensus was reached to allow us to discuss the components of a high-level architecture for PSL. The discussion was started by a presentation suggesting the terminology to be used:

- Semantic (model) - Capturing the meaning of the various components of PSL
- Syntax - An interpretation of the semantic, composed of at least one lexicon, grammar and presentation
- Lexicon - A formal definition of the symbols supporting a given syntax, including the interpretation (mapping) to the underlying semantic
- Grammar - The rules for combination of the symbols appearing within the lexicon
- Presentation - An encoding of the symbols defined in the lexicon

Once this terminology was agreed upon, the topic focused on three issues: the determination of architecture on the language; PSL from the industry perspective; and the use of scenarios.

3.2.1 The Architecture of the Language

The goal of this topic was to reach consensus on what the PSL would "look like." There was an overwhelming consensus that the PSL should have at least the following three components, as described by Selden Stewart:

- a semantic level - to describe what real processes are;
- one or many syntactic levels (languages) - how we specify (represent) those processes;
- a mapping from the syntactic level to the semantic level - preserves all concepts that are necessary in the semantic level.

Selden Stewart adds that "there is also a need for decomposition of processes which will ultimately end at a primitive process. This requirement is at both the semantic level and the language level."

The issue of the merits of computer-readable representations, and the supporting tools, was discussed. The conclusion reached was that such tools are tremendously useful at the syntactic level, but could not be applied at the semantic level by virtue of the fact that computers cannot "understand" semantics without an associated syntax. EXPRESS models were discussed as examples of syntactic interpretations of (undefined) semantic structures.

Chris Menzel adds that "There was overwhelming agreement that the architecture of PSL be founded upon a basic ontology, that is, a set of identified primitives that are then axiomatized in way that will fix their meanings adequately for further developments and also to support the use of the PSL when the work is complete." This ontology would serve as the semantic level of the above architecture. Menzel goes on to say that the ontology should not be created from scratch, it should "heavily leverage other commonly accepted work (e.g., Hayes and Allen's axiomization of temporal intervals [4])."

When asked what approach the project should take in creating such an architecture, Menzel replied with the following:

> "We should first begin by defining the basic semantical structures for PSL. This will involve, first, a choice of semantic primitives (e.g., "process", "time-interval", and "resource") and then, second, a rigorous characterization of the properties of, and relations among, those primitives. This characterization is usually given in "mathematical English", i.e., natural language supplemented with whatever mathematical tools are

> needed to characterize the semantic structures rigorously and unambiguously. This often involves the development of an abstract mathematical model of the semantical structures.
>
> Next, a given lexicon is chosen for talking about process. One then interprets the elements of the lexicon in terms of the basic semantical concepts. Finally, the semantics is captured in the formal language in terms of a set of axioms."

The last subject we discussed related to this issue was how different representational approaches (constraint-based, directed graph, object-oriented, etc.) fit into the PSL architecture. The overwhelming response was that these approaches were simply syntactical preferences which played no role in the semantic underlayer (the first step in the architecture). Therefore, the decision of which approach(es) to use within the language could be delayed until later in the project.

3.2.2 PSL from the Industry Perspective

Considering the probable theoretical complexity of the above architecture, there was concern from the industry representatives that the PSL may get so consumed in the technical details of formal semantic definitions, multiple syntaxes, etc. that the final product would be unintuitive and therefore unusable. Debra Stephens captures this well when she wrote "One large concern is the applicability of PSL to industry and vendors. Simplicity, interoperability, and comprehensibility are essential to ensure that industry will embrace and implement PSL. We must make sure that the theoretical decisions that we make will not negatively effect the usability of the PSL." Kurt Freimuth added "I hope we agreed on a "multi-layered" approach with a language to precisely define primitive and a user-friendly language on top." Michael Duffey also recognized this concern and suggested "If real industry usage is the goal, then at the least there needs to be a "bottom up" approach in parallel to formal language development that uses some types of scenarios as a reality check."

3.2.3 Scenarios

There seemed to be general agreement that the use of scenarios would be imperative to ensure, among other things, proper understanding of various aspects of the project's goals. In this context, a scenario is a description of a set of activities which is to be represented (if possible) by the process specification language. Jeffery Herrmann identifies three uses of scenarios:

1. to allow us to explore the semantic concepts in question;
2. to allow us to compare different representations and architectures; and
3. to show what type of information the PSL will convey.

In addition to these uses, scenarios can also help us understand what types of situations are common in industry which would allow us to ensure the specification language is able to handle these situations. This will ensure that the PSL truly addresses real-world requirements.

3.3 Topic 3: Creating a Semantic Model: An Exercise

The second day of the workshop focused on the creation of a basic semantic model including some of the most common concepts within process. Five common semantic concepts were identified:

- Process
- Participant
- Duration
- Sequence
- Driver

The participants broke into five groups, each focusing their efforts on defining one of the five concepts. In the limited amount of time available, each group arrived at the following definitions:

- Process - something that occurs over an interval of time, and includes one or more participant. One might also want to include that it must involve change and it must have a set of preconditions in order for it to occur.
- Participant - something that is acted upon by a process or something needed for the process to occur.
- Duration - the measure of an interval between two points in time. A process has zero or more durations including: planned, actual, desired, and estimated.
- Sequence - a temporal ordering of processes or states.
- Driver - the motive for the existence of a process and/or activity. Activities have drivers in reality which need not be communicated. Drivers may be related to each other, and there may exist a hierarchy among them.

These natural language definitions would then be formalized, possibly using formal mathematical english, i.e., natural english supplemented by first-order logic. From here, at least one syntax would be created/determined and the semantic concepts would be mapped to that syntax.

When this exercise was complete, the Roundtable moved on to consider possible future actions to be taken. A list of possible tasks was constructed, listed below:

- Define scenarios
- Define semantics

    Scavenge other approaches

    Provide formal description

- Define lexicon/syntax
- Define structure of PSL
- Determine performance measure

    Map to requirements

    Map to existing languages

These actions eventually resulted in team assignments. The following teams were formed:

- Scenarios team - to create a set of scenarios from a number of different domain/discipline perspectives.
- Semantics team - to define, both informally and formally, the requirements found in the first phase of the PSL project. This will most likely involve the determination of primitive concepts necessary to describe process and then basing all of the requirement's definitions on these primitive concepts.
- Syntax/Presentation team - to define the mapping between the semantic concepts and various presentation approaches (constraints, relational databases, KIF, Object Oriented, Unified Modeling Language (UML))

These teams, composed of Roundtable participants, agreed to communicate using email exploders that NIST would set up and to report back with preliminary results by June 21, 1997.

After the team assignments, the Roundtable concluded. Along the way, a number of additional ("parking lot") issues came up, some of which were addressed during the discussions. Here is the full list of the "parking lot" issues:

- Can multiple syntaxes or presentations coexist? (Yes)
- Underlying formalism/API or notations
- Need scenarios and examples (Assigned to a team)
- Need to understand requirements: - definition (ontology) - basis for PSL core
- What methodologies would be appropriate to capture the semantic model? (A combination of approaches used by Menzel, Gruninger and Tate seems likely)

4.0 Conclusion and Future Direction

The goal of the PSL Roundtable was to assemble key champions and stakeholders of various representational approaches for process in order to discuss the relative merits, in the hopes of reaching consensus on a language architecture and to establish a technical approach for proceeding. This goal was in fact achieved, with an agreement, among other things, that the language architecture should be based upon a formal semantic foundation, upon which would be layered a number of syntactic mappings, each with one or more presentations.

Throughout the course of the Roundtable, a number of key issues were identified that could not be resolved during the limited time available. At the conclusion of the Roundtable, three teams were formed to address these issues. These teams, referred to as the scenarios team, the semantics team, and the presentation team, will provide the following deliverables by June 21st:

- Scenarios team - a set of real-world scenarios to support the identification and definition of semantic concepts.
- Semantics team - a semantic description covering a small subset of the core language requirements

- Presentations team - three syntactic interpretations of that semantic description, mapping to object-oriented, KIF, and constraint-based presentations. A relational presentation was also deemed important, but no assignment was made.

The communication within each team and among the different teams will be facilitated by NIST through the use of a set of mailing lists. Each team will also include a NIST representative to help facilitate the discussions.

Much of the success of the Roundtable can be attributed to the wide spectrum of people involved and to the Roundtable's format. With representatives from industry, academia, and government all providing comments and suggestions from various perspectives, we could ensure that the decisions made would provide the robustness necessary to allow the language to be acceptable to all parties involved.

The format of the Roundtable also proved to be an asset which allowed us to make maximum use of the minimal amount of time we were physically together. By starting the Roundtable virtually with an email discussion, many of the preliminary tasks that are necessary for any Roundtable, such as introductions and identification of issues, could be done up-front and therefore provided more time to tackle the issues during the time we were physically together. During the physical Roundtable, the presentation/discussion/write-up format allowed us to clearly define the issues, allowed plenty of time for discussion, and then allowed participants to write about their perspective of what went on during the discussion plus add any other comments they didn't get a chance to bring up. Many of these write-ups contributed directly to the creation of this proceedings.

5.0 References

[1] Process Specification Language project, http://www.nist.gov/psl/, September 26, 1997.

[2] Knutilla, A., Schlenofff, C., Ray, S., Process Specification Language: Analysis of Existing Process Representations, *to be published as a NIST Internal Report in October, 1997*, National Institute of Standards and Technology, Gaithersburg, MD, October, 1997.

[3] 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.

[4] Hayes, Patrick J., A Catalog of Temporal Theories, Beckman Institutes and Departments of Philosophy and Computer Science, University of Illinois.

Appendix A: Requirement Groups vs. Representational Constructs

(Requirements in each group are listed on the following page)

Requirement Groups / Representational Constructs	Resource Representation and Characteristics	Plan/Task Representation	Resource/Task Characteristics	Precedence /Sequences	Constraints	Date/Time
Object-Based	xxxxxxxxxx	xxxxxxxxxx	xxxxxxxxxx	xxxxxxx	xxxxxxx	xxxxxxxx
Constraints-based	xxxxxxxxxx	xxxxxxxxxx	xxxxxxxxxx	xxxxxxx	xxxxxxx	xxxxxxxx
Graph-based	-	-	-	xxxxxxx	xxxxxxx	xxxxxxxx
Relationships	-	-	-	xxxxxxx	xxxxxxx	xxxxxxxx
Conditions	-	-	-	xxxxxxx	xxxxxxx	-
Inference	-	-	-	xxxxxxx	-	-
Task Decomposition	-	xxxxxxxxxx	xxxxxxxxxx	-	-	-
Annotations	-	xxxxxxxxxx	-	-	-	-
Activity-Resource Matrix	-	-	xxxxxxxxxx	-	-	-
Task-Network	xxxxxxxxxx	xxxxxxxxxx	-	-	xxxxxxx	-
Classes (Groups)	-	xxxxxxxxxx	-	-	-	-
Timer	-	-	-	-	-	xxxxxxxx
Time Intervals/ Timepoints	-	-	-	-	-	xxxxxxxx

KEY

xxx - the representational constructs could support the requirements groups in the representations analyzed

Precedence/Sequences

- simple precedence
- simple sequences
- complex sequences
- alternative tasks
- concurrent tasks
- conditional tasks
- iterative loops
- parallel tasks
- serial tasks

Resource Representation/Characteristics

- resource representation
- simple resource capability/characteristics
- product representation
- product characteristics
- resource categorization and grouping

Plan/Process Representation

- task representation
- simple task characteristics
- simple grouping of tasks

Resource/Task Characteristics

- task executor
- resource requirements for a task
- level of effort
- cost data
- ad hoc notes

Constraints

- general constraints
- pre- and post- conditions
- state existence constraints
- temporal constraints

Date/Time

- date and time
- task duration

Appendix B: Initial Agenda

21 APR: 8:30 - 5:00

8:30 - 9:00	Registration
9:00 - 10:00	PSL Directions: Goals, Definitions, the Roundtable Approach (Presented by Steven Ray)
	BREAK
10:30 - 12:30	Presentation/Discussion/Written summaries of Topic 1 (Convened by Amy Knutilla)
12:30 - 1:30	LUNCH
1:30 - 3:30	Presentation/Discussion/Written summaries of Topic 2 (Convened by Craig Schlenoff)
	BREAK
4:00 - 5:00	Discussion of identified issues, topics
6:30	DINNER The Old Town Tavern, Gaithersburg

22 APR: 9:00 - 3:00

9:00 - 11:30	Presentation/Discussion/Written Summaries, Topic 3 *
11:30 - 12:30	Identify and discuss remaining issues
12:30 - 1:30	LUNCH (NIST Cafeteria)
1:30 - 3:00	Conclude discussions / Present revised strawman representation approach to PSL

TOPICS:

TOPIC 1: Constructs and features for representing processes. As a result of the first two phases of the PSL project and based on your input, we have identified families of constructs and features used to address process specification requirements. The results of this analysis will serve as the basis for identifying appropriate constructs for the PSL schema.

> Are the conclusions from the “Process Specification Requirements vs. Existing Process Representations” analysis valid?

TOPIC 2: The architecture of the PSL. Fundamental to the design of the PSL is its architecture. This decision will determine to a large extent the performance and behavior of the language as well as which requirements will be easily represented.

What is the structure of the language?

Can multiple representation features be effectively combined to form the basis for the PSL?

What role do ontologies play?

Will the architecture hold for “beyond the core” requirements?

* OTHER CANDIDATE TOPICS:

TOPIC: Related efforts. Several well-researched projects are pursuing goals similar to PSL that could benefit with leverage, coordination, etc. What are the complementary, overlapping, and distinguishing aspects of other related projects (PSL, PIF, ARPI, WfMC, TOVE)?

TOPIC: Scenarios. Scenarios (benchmarks) have been proposed as a mechanism to describe both requirements and representation approaches to addressing requirements. These scenarios would facilitate understanding of requirements and of representation approaches.

What scenarios need to be developed?

TOPIC: Requirements for specifying process. In the first phase of the PSL project, we identified a set of requirements for specifying process. We initially structured these as core, outer core, extension, and application requirements. We have received many suggestions from different perspectives for better defining process specification requirements.

What’s a better approach to defining process specification requirements?

What is the benefit(s) of this approach at this phase of PSL development?

Did we miss important requirements for specifying process?

Appendix C: Participants and Affiliations

Participant	**Affiliation**
Richard Anderson	George Washington University
Shu Chiun Cheah	University of Maryland, College Park
Michael Duffey	George Washington University
Salah Elmaghraby	North Carolina State University
Kurt Freimuth	Institute of Advanced Manufacturing Sciences (IAMS)
Michael Gruninger	University of Toronto
Jeffrey W Herrmann	University of Maryland, College Park
Amy Knutilla	NIST
Chris Menzel	Knowledge Based Systems Inc. (KBSI)
Dana Nau	University of Maryland, College Park
Adam Pease	Teknowledge Corp.
Steve Polyak	Artificial Intelligence Applications Institute (AIAI)
Steve Ray	NIST
Craig Schlenoff	NIST
Robert Shapiro	Meta Software Corp.
Debra Stephens	Raytheon
Selden Stewart	NIST
Austin Tate	Artificial Intelligence Applications Institute (AIAI)
Florence Tissot	Knowledge Based Systems Inc. (KBSI)
Richard Wysk	Pennsylvania State University