

# USING PROCESS SPECIFICATION LANGUAGE FOR PROJECT INFORMATION EXCHANGE

Jinxing Cheng<sup>1</sup> and Kincho H. Law<sup>2</sup>

## ABSTRACT

There are many project scheduling and management programs employed in the construction industry. Standards-based translation is one way to achieve interoperability. This study evaluates the applicability of the Process Specification Language (PSL) for exchanging project information among different applications. PSL has been initiated by National Institute of Standards and Technology (NIST) and is emerging as a standard exchange language for process information in the manufacturing industry. In this paper, we explore how PSL can be extended for exchanging project information for construction applications.

## KEY WORDS

Process Specification Language, PSL, ontology, information exchange, project management

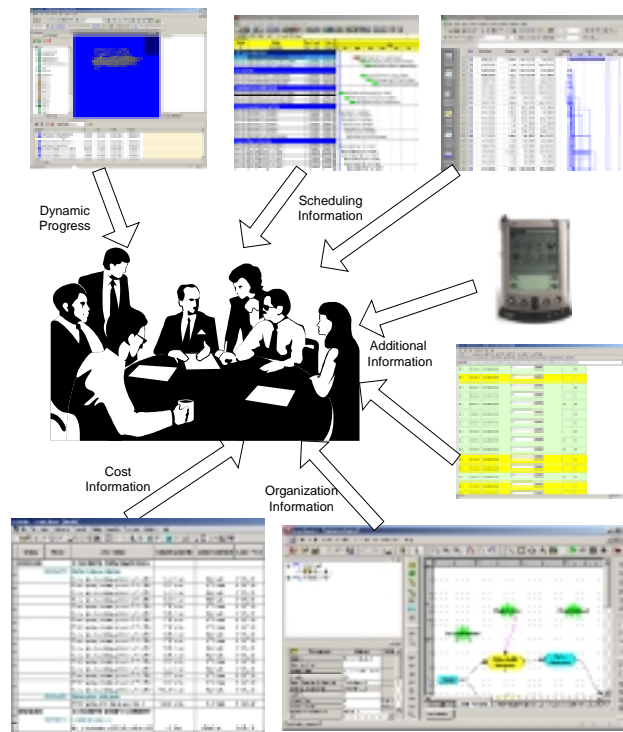
---

<sup>1</sup> PhD Student, Civil and Environmental Engineering Department, Stanford University, Stanford, CA 94305-4020, email: [cjx@stanford.edu](mailto:cjx@stanford.edu)

<sup>2</sup> Professor, Civil and Environmental Engineering Department, Stanford University, Stanford, CA 94305-4020, email: [law@cive.stanford.edu](mailto:law@cive.stanford.edu)

## 1. INTRODUCTION

As the use of information technology increases in the construction industry, the capability of software applications to interoperate has become increasingly important. A construction project usually involves volume of project information from different sources. There are many construction applications that could be employed in a construction project. For example, as shown in Figure 1, different members of a project team may use Primavera Project Planner (P3)<sup>TM</sup> or Microsoft Project<sup>TM</sup> to schedule the project, Vite<sup>TM</sup> to simulate the project organization, Timberline's Precision Estimating<sup>TM</sup> to estimate project cost, and 4D Viewer (McKinney and Fischer 1998) to view the progress of construction. In a distributed but concurrent engineering environment, information interoperability plays a significant role in project management.



**Figure 1: Construction Applications in Project Management**

There have been many ontology standards, such as STEP, IFC and aecXML, which aim to provide interoperability among different applications. Most of the existing ontology standards, however, focus more on product data rather than process information. Process Specification Language (PSL) is one such emerging standard proposed by NIST (National Institute of Standards and Technology) that is designed specifically to exchange process information among manufacturing applications. This study explores the applicability of using PSL as an interchange standard for construction project management applications.

## 2. EXISTING ONTOLOGY STANDARDS

For the last two decades there have been many efforts from the industry, academia and standards organizations to propose ontology standards for data exchange, such as STEP (ISO 1994), IFC (IAI 1997), aecXML (IAI 2002), etc.

- STEP (the Standard for the Exchange of Product Model Data) is a product data integration standard to enable information exchanging among different applications (Fowler 1995). STEP is based on the EXPRESS language, which enables STEP to provide unambiguous, computer interpretable representation of product data. EXPRESS is a data definition language that is used to represent the structure of data and any constraints that may apply to it.
- IFC (Industry Foundation Classes) is a data representation standard for defining product data for architectural and construction applications. There have been efforts to extend IFC from product modeling to support data for cost estimating and project management purpose. In short, IFC is designed to exchange data among AEC/FM (Architecture, Engineering, Construction and Facilities Management) applications. IFC is also based on the EXPRESS language.
- AecXML was initially proposed by Bentley Systems in 1998, and is now part of the effort under the IAI (International Alliance of Interoperability). AecXML includes XML schemas to describe information specific to the design, construction, and operation of buildings, plants, infrastructure, and facilities. Efforts have been made to share the IFC and aecXML developments.

The ontology standards described above focus mainly on product data and do not have extended information about process and task specifications.

## 3. PROCESS SPECIFICATION LANGUAGE (PSL)

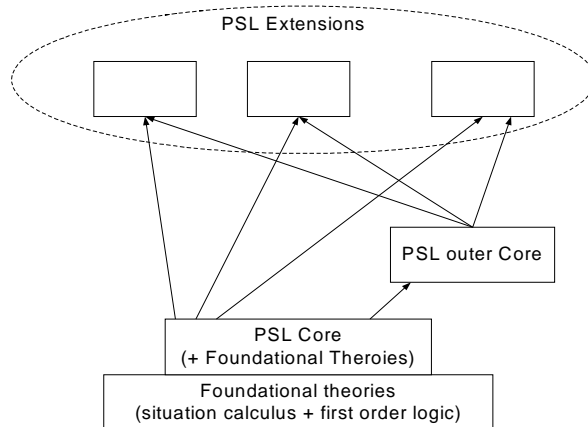
The development of Process Specification Language (PSL) is motivated by two basic reasons. First, there are not many existing standards for process information exchange. Second, current ontology standards lack a formal logic to define relationships and constraints. Every ontology in PSL is either formally defined (for non-primitive terms) or there are a set of axioms associated with it to constrain the meaning (for primitive term).

PSL is based on KIF (Knowledge Interchange Format), which is designed for knowledge interchange among disparate computer systems. KIF has declarative semantics, and is logically comprehensive (Genesereth and Fikes 1992). When combined with domain specific ontology, KIF has the expressive power to represent knowledge.

Figure 2 shows the overall organization of PSL, which includes the PSL core, the PSL outer core and PSL Extensions (Schlenoff et al. 2000).

- The PSL core is a set of axioms based on KIF. The PSL core includes four basic classes: Object, Activity, Activity\_Occurrence and Timepoint. Relations are defined among the classes, for example:  
*(occurrence-of activity-occurrence activity)*  
*(before timepoint timepoint)*

- PSL outer core consists of a small set of extensions, which are generic and pervasive in their applicability. The extensions in the PSL outer core include Subactivity Extension, Activity-Occurrence Extension and States Extension. Relations can be defined using the PSL outer core extensions, for example:  
*(subactivity-occurrence activity-occurrence activity-occurrence)*  
*(subactivity activity activity)*
- PSL extensions include ontology modules such as generic activities, ordering relations and schedules. Each module is motivated by a set of applications and covers concepts in certain domain. Below are some example relations in the PSL extensions:  
*(before-start activity-occurrence activity-occurrence activity-occurrence)*  
*(before-start-delay activity-occurrence activity-occurrence activity-occurrence duration)*



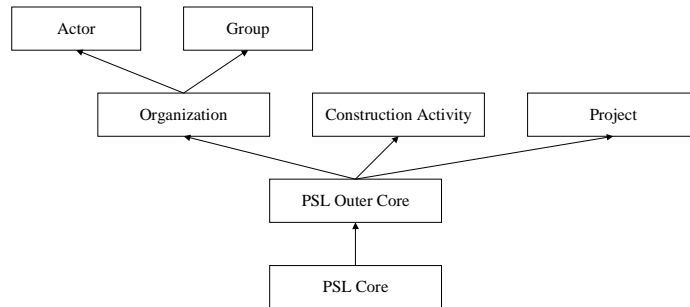
**Figure 2: PSL Ontology**

#### 4. PSL FOR PROJECT MANAGEMENT APPLICATIONS

While PSL was initially created mainly for manufacturing industry, it can be applied to construction project management applications by extending the ontology to model the essential project information.

Ontology (Guarino 1997) is an explicit specification of some topic. In other words, ontology includes a set of terms and the relationships among those terms. When two programs need to exchange process information, they not only need to agree on a representation language for the interaction, but also need to agree on an ontology in their domain. In our research we select Vite<sup>TM</sup> as the benchmarking application to evaluate the applicability of PSL in the construction industry. Vite<sup>TM</sup> is a project and organization modeling system designed to assist in developing organizational structures and identifying potential problems with project cost, time, or quality. It takes traditionally qualitative organizational management theory and builds a model that incorporates rough quantitative measures.

For a typical construction project, project information includes three basic categories: scheduling information, resource information and cost information. Currently PSL ontology covers primarily the scheduling part. After analyzing the information needed and the output by Vite™, extensions to the current PSL ontology are designed to include organization module, construction activity module and project module (Figure 3).



**Figure 3: Suggestion of Extensions to Current PSL Ontology**

One of the main concerns in the ontology extensions is that we should not include too many application specific concepts in the PSL ontology. PSL is not designed to capture all the information needed by each application. There should be some other internal schemas to represent all the application specific information. On the other hand, PSL should be sufficiently general and capable to exchange the essential process information among different applications. We should define the concepts, which are also relevant to other applications, generalize those concepts, and include them in the PSL ontology extensions.

The organization ontology focuses on organization structure, roles, authority and empowerment (Fox et al. 1996). An organization can be individual or group of individuals to which organizational attributes and relations are associated with. Some examples of the relations defined in the organization extension are:

*(Experience actor string)*

*(Director group actor)*

Construction activity is an activity in a construction process associated with certain attributes, such as priority, uncertainty, and dependency. Examples of the construction activity extensions include:

*(Priority ConstructionActivity number)*

*(Dependency ConstructionActivity ConstructionActivity number)*

Project ontology extension covers general project information, for example:

*(ActivityAssignment project ConstructionActivity actor number)*

*(WorkDay project number)*

For each new term in the ontology extensions, we provide some definition or axioms to define and limit the interpretation of the new term. As an example, the following shows the definition and the axiom for dependency defined in the ontology extensions.

***Definition of Dependency:***

*(defrelation dependency (?a1 ?a2 ?n) :=*

```
(=> (rework ?a2)
      (and (rework ?a1)
            (numrework ?a1 n)))
```

**Axiom: The dependency relationship is transitive.**

```
(forall (?a1 ?a2 ?a3)
  (=> (dependency ?a1 ?a2 ?n1)
        (dependency ?a2 ?a3 ?n2)
        (exist ?n3 (dependency ?a1 ?a3 ?n3))))
```

## 5. INFORMATION EXCHANGE USING PSL

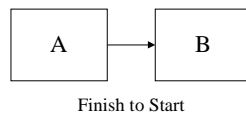
### 5.1 MAPPING CONCEPTS BETWEEN PSL AND CONSTRUCTION APPLICATIONS

It is not unusual that the same term is often associated with different meanings in different applications. To exchange project information, first we need to map the concepts in different applications onto PSL ontology, so that they are PSL compliant. Table 1 shows some terms that are related to activity relationships in PSL and Primavera Project Planner (P3)<sup>TM</sup>.

**Table 1: Terms in P3<sup>TM</sup> and PSL about activity relationship**

Concepts in P3	PSL Ontology
Successor	Successor
Predecessor	after-start after-start-delay .....

The example in Figure 4 shows that activity B is the successor of activity A with relationship type *FinishToStart* in a construction project P from P3<sup>TM</sup>.



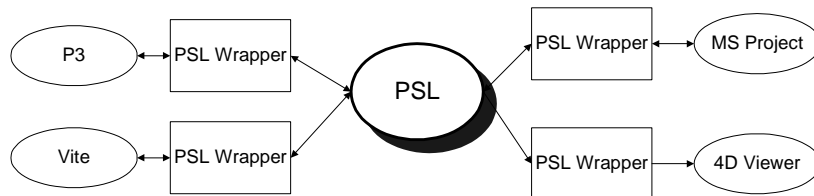
**Figure 4: Successor relationship in P3**

Suppose that the time lag is 3 days between activity A and B, then the successor concepts can be expressed using PSL ontology as:

```
(activity-occurrence A)
(activity-occurrence B)
(subactivity-occurrence A P)
(subactivity-occurrence B P)
(after-start A B P)
(after-start-delay A B 3)
```

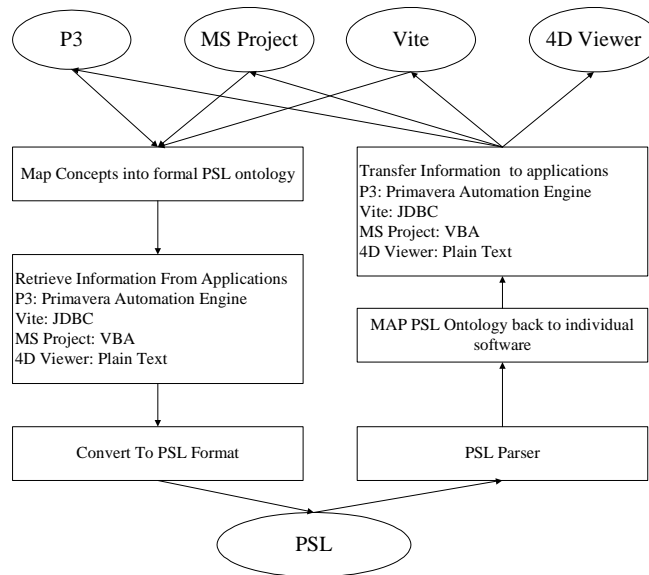
## 5.2 EXCHANGING PROJECT INFORMATION USING PSL

To exchange project information among different construction applications, we need to develop wrappers for each application (Figure 5). The wrappers are used to retrieve information from applications and convert the information into the PSL format. In addition, the wrappers are also used to parse information from PSL files and transfer the information to applications.



**Figure 5: PSL in the Information Exchange**

The basic process of using PSL for project information exchange can be illustrated in Figure 6. Although the basic processes for the PSL wrappers are essentially the same, different implementations are necessary to build the wrappers for different construction applications.



**Figure 6: PSL Wrappers**

- For Vite<sup>TM</sup>, the concepts in Vite<sup>TM</sup> are mapped onto the formal ontology described using PSL, which explicitly and unambiguously defines all terms introduced within the language. We then parse the relevant information stored in the Access database using Java Database Connectivity (JDBC), translate the information into PSL according to a set of rules, and create a PSL file. For the PSL to Vite<sup>TM</sup> translation, the information in the PSL file is parsed and rewritten into VNB (Access database) file format. Vite<sup>TM</sup> could open the VNB file and start the simulation.

- For P3™, Primavera Automation Engine (RA) is employed. RA is a set of object-oriented, OLE 2.0-based API, which allows object-oriented programming access to the P3™ scheduling engine and other applications. We use RA to communicate with P3™, such as retrieving project information from P3™ and transferring project information to P3™. Figure 7 shows a sample code to illustrate the PSL wrapper for P3™.

```

Sub setActivities()
  Dim I As Integer
  Dim bret As Boolean

  For I = 0 To numActs - 1 Step 1
    Dim act As Object
    Set act = m_Project.Activities.NewItem()
    act.ActivityID = acts(I).id
    act.description = acts(I).description
    act.OriginalDuration = acts(I).duration
    act.EarlyStart = acts(I).beginof
    bret = m_Project.Activities.Add(act)
  Next I
End Sub

```

**Figure 7: Sample Code of the PSL Wrapper for P3**

- For Microsoft Project™, VBA (Visual Basic for Application) is employed. The process here is very much similar to the development of communication protocols for Primavera P3™.
- For 4D Viewer (McKinney and Fischer 1998), the scheduling information from the PSL file is retrieved and converted into the format required by 4D Viewer.

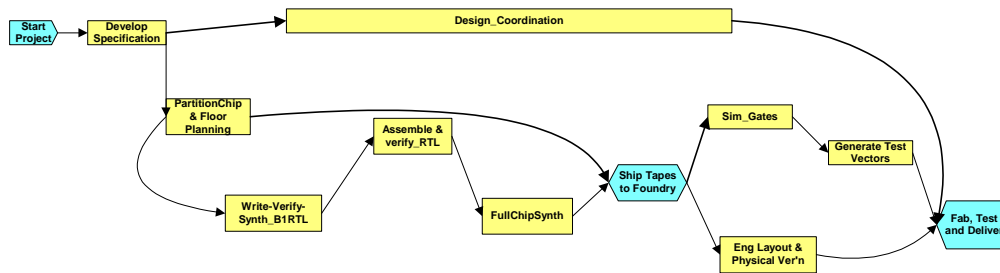
## **6. DEMONSTRATION OF INFORMATION EXCHANGE USING PSL**

### **6.1 EXAMPLE 1: A CHIP DESIGN SCENARIO**

We select a sample project from Vite™ to test PSL as an interchange standard for process information. Vite™ is a project and organization modeling system designed to assist in developing organizational structures and identifying potential problems with project cost, time, or quality. A Vite™ project is composed of a traditional CPM diagram and additional links showing failure dependence, reciprocal information and management structure.

The example scenario (Figure 8) is to design and fabricate a chip set for a new personal digital assistant (PDA) product within a tight schedule. There are 12 activities in this project. Among the 12 activities there are three milestone activities: ‘Start Project,’ ‘Ship Tapes to Foundry’ and ‘Fab, Test and Deliver.’ The activity ‘Design\_Coordination’ is to maintain the overall control of the project.





**Figure 8: Original CPM Diagram in Vite**

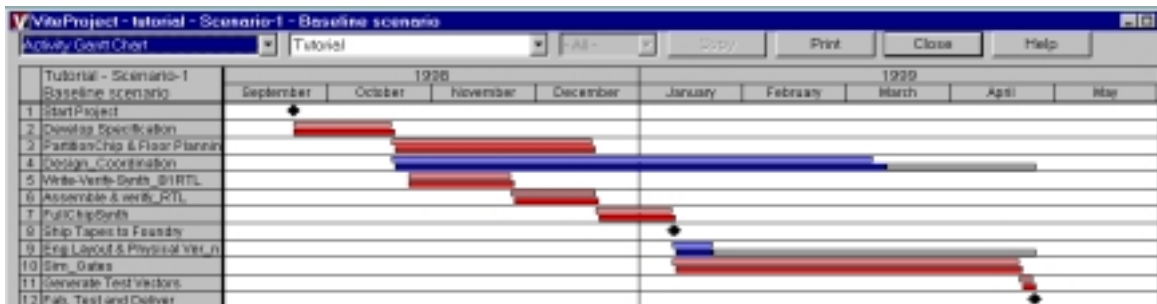
Using PSL as an interchange standard, we successfully exchange scheduling information among Vite™, Primavera Project Planner (P3)™ and Microsoft Project™. Figure 9 shows some selected logic sentences from the PSL file particular to this Project.

```

    (and
      (project TUTO)
      (doc TUTO "TUTORIAL Project")
      (beginof TUTO 9/18/1998)
      (subactivity-occurrence ID100 TUTO)
    )
    .....
  )
  (and
    (activity-occurrence ID190)
    (doc ID190 "PartitionChip & Floor Planning")
    (beginof ID190 10/19/1998)
    (duration-of ID190 42)
    (before-start ID190 ID130 TUTO)
    (before-start-delay ID190 ID130 TUTO 0)
  )
  .....
  )
  
```

**Figure 9: Sample PSL File**

Figures 10 to 12 illustrate the generated schedule in Vite™, P3™ and Microsoft Project™. Figure 10 is the original Gantt chart of the sample project in Vite™. Figures 11 and 12 show the regenerated project schedule in P3™ and Microsoft Project™, respectively. As shown in the figures, project information is being exchanged successfully among these three applications. Activities have the same start date and duration in all three applications. The critical paths are also the same in all three applications.



**Figure 10: Original Gantt Chart in Vite**

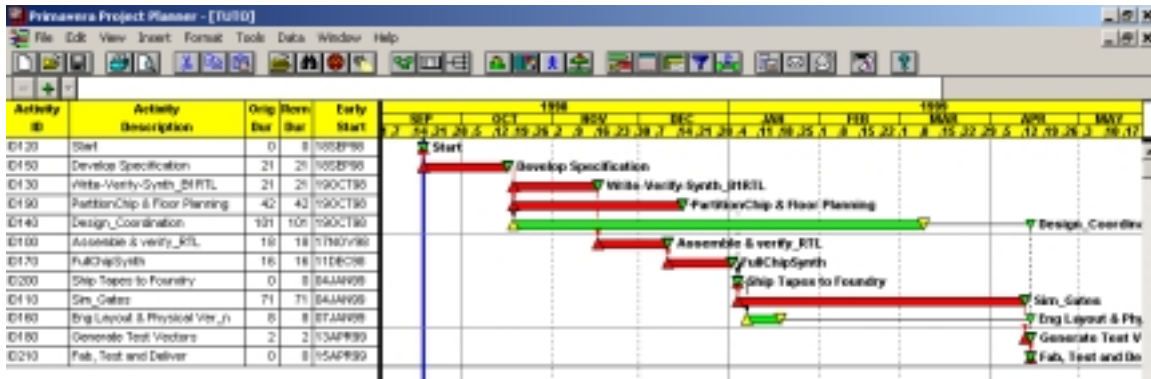


Figure 11: Regenerated Schedule in Primavera Project Planner using PSL

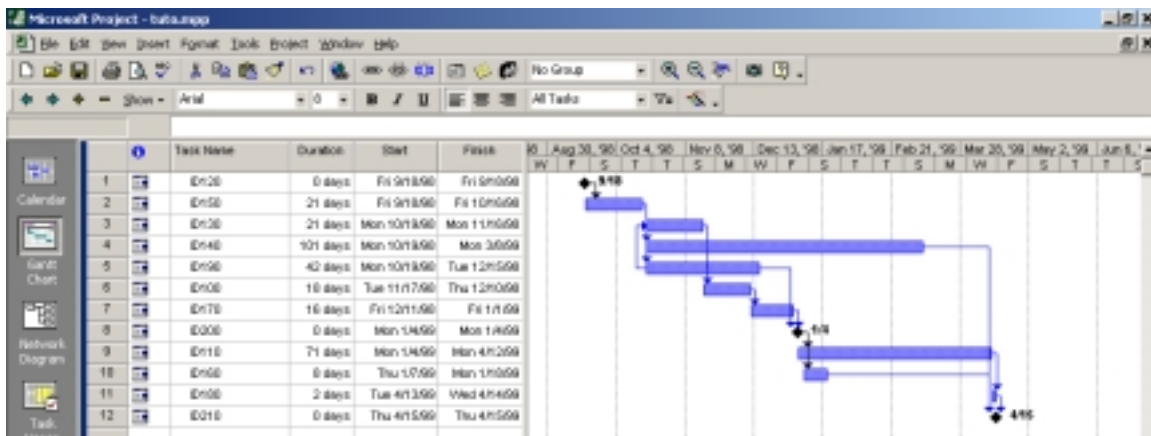


Figure 12: Regenerated Schedule in Microsoft Project using PSL

## 6.2 EXAMPLE 2: MORTENSON CEILING PROJECT

To test the scalability and applicability of PSL as an interchange standard, the Mortenson Ceiling Project is employed to illustrate the information exchange process. The Mortenson Ceiling Project is a portion of the construction of the Walt Disney Concert Hall, built by Mortenson Construction, and designed by Frank O.Gehry & Associates.

We use PSL as the data standard to exchange project information among P3<sup>TM</sup>, Microsoft Project<sup>TM</sup>, and 4D Viewer (McKinney and Fischer 1998). There are 191 activities and 459 dependency relationships in the project. In the PSL file of this project, there are more than 2000 logic sentences.

Figures 13 to 15 show selected results of this example demonstration. Figure 13 is the original Gantt chart of the ceiling project in P3<sup>TM</sup>. Figure 14 is a snapshot of construction progress in 4D Viewer on March 25, 2001. Figure 15 is the modified Gantt chart regenerated in Microsoft Project<sup>TM</sup>, where the duration of activity 18T1-33201 has been changed from 1 day to 40 days. As shown in Figure 15, the scheduling information originally in Primavera Project Planner (P3)<sup>TM</sup> is successfully regenerated in Microsoft Project<sup>TM</sup> using PSL.

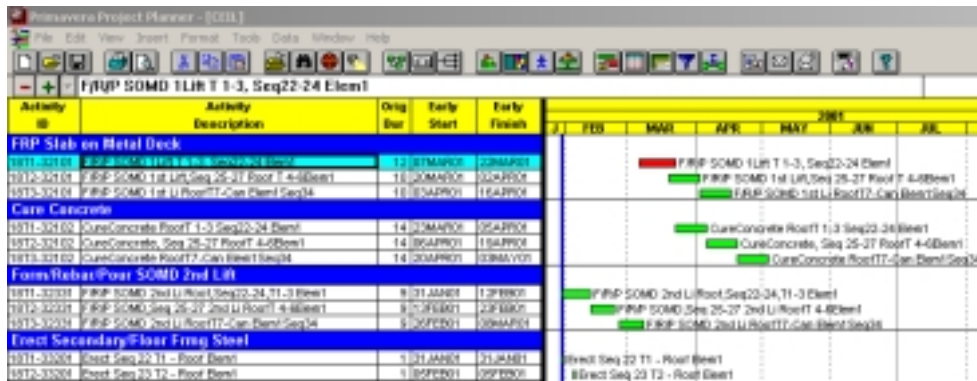


Figure 13: Original Schedule in Primavera Project Planner

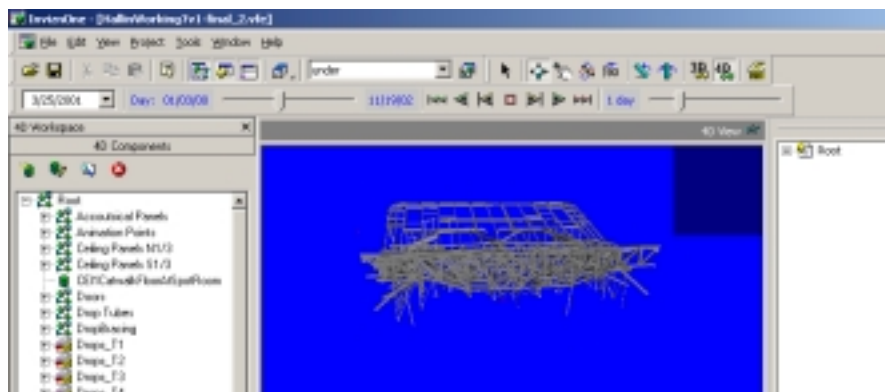


Figure 14: Original Model in 4D Viewer Taken on March 25, 2001

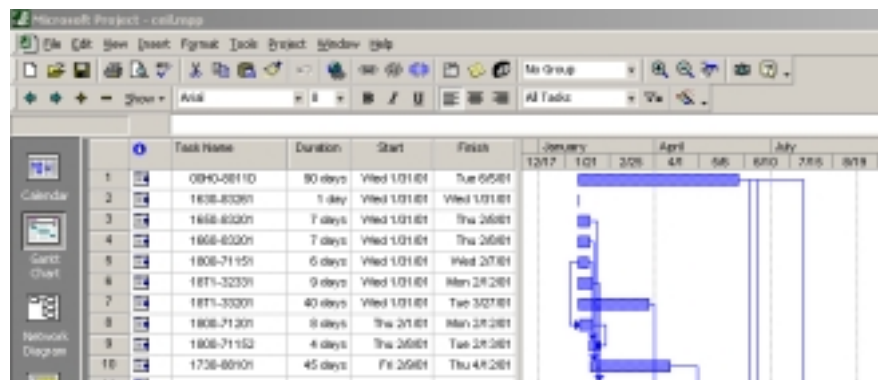


Figure 15: Regenerated Gantt Chart in Microsoft Project using PSL

## 7. CONCLUSIONS

Many construction applications can be employed in a construction project. To exchange project information among different construction applications poses a challenge for collaborative team members of a project. Although PSL has originally been designed

specifically for process information related to manufacturing applications, we have successfully extended the ontology to model essential construction project information. We have developed PSL wrappers for some typical construction applications, and successfully exchange project information among those applications, such as P3<sup>TM</sup>, Microsoft Project<sup>TM</sup>, Vite<sup>TM</sup> and 4D Viewer (McKinney and Fischer 1998). Our research shows that PSL, an emerging interchange standard for manufacturing applications, is also a promising candidate interchange standard for construction project management applications.

Our successful demonstration also shows that PSL can be used for information exchange among different applications, thus facilitating concurrent engineering processes in construction from different team members. Process information can be shared among various applications by different teams of different disciplines.

## 8. ACKNOWLEDGEMENTS

This work is partially sponsored by the Center for Integrated Facility Engineering at Stanford University, a Stanford Graduate Fellowship, and the Product Engineering Program headed by Dr. Ram D. Sriram at NIST. The Product Engineering Program gets its support from the NIST's SIMA (Systems Integration for manufacturing Applications) program and the DARPA's Radeo Program. The 4D Viewer and the 4D model of the Mortenson Ceiling Project are provided by Professor Martin Fischer and his research group at Stanford University.

## 9. REFERENCES

- Fowler J. (1995), *STEP for data management, exchange and sharing*, 214 pp, Technology Appraisals Ltd., UK, 1995.
- Fox, M.S., Barbuceanu, M., and Gruninger, M. (1996), "An Organisation Ontology for Enterprise Modelling: Preliminary Concepts for Linking Structure and Behaviour." *Computers in Industry*, 1996, Vol. 29, pp. 123-134.
- McKinney, K. and Fischer, M. (1998) "Generating, evaluating and visualizing construction schedules with 4D-CAD tools." *Automation in Construction*, 7(6), 433-447.
- Genesereth, M.R. and Fikes, R. (1992), "Knowledge Interchange Format Version 3.0 Reference Manual." Computer Science Department, Stanford University.
- Guarino N. (1997), "Understanding, building and using ontologies." *Int. J. of Human-Computer Studies*, Feb-Mar 1997, Vol. 46, no 2-3, pp. 293-301.
- IAI (1997), "Industry Foundation Classes." Specification Volumes 1-4, International Alliance for Interoperability, Washington DC 1997.
- IAI (2002), "AecXML." International Alliance for Interoperability, <http://www.aecxml.org> (Accessed: 15 March 2002).
- ISO (1994), 10303-1:1994, "Product data representation and exchange: Part 1: Overview and fundamental principles."
- Schlenoff, C., Gruninger M., Tissot, F., Valois, J., Lubell, J., and Lee, J. (2000), "The Process Specification Language (PSL): Overview and Version 1.0 Specification," NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD.