PSL Quarterly Progress Report

Project Title: Process Specification and Simulation
Date: March 31, 2001
Principal Investigator: Kincho H. Law, Stanford University
Period Covered: January 1, 2001 – March 31, 2001

Project Objectives

The proposed project is intended to be a feasibility study to evaluate the process specification language (PSL) and a simulation query language (SimQL) with application to project and workflow management. This proposed joint research effort includes:

- to evaluate the potential of the Process Specification Language (PSL) for the planning and modeling activities in a project
- to evaluate the adequacy of the Process Specification Language (PSL) as an interchange definition language to support process-oriented simulation
- to evaluate the applicability of SimQL, a simulation query language, for practical engineering problems
- to develop an integration framework using PSL and SimQL for process-oriented simulation

Our long-term goal is to develop a distributed network-based framework to integrate process specification and modeling and virtual simulations of project activities. To facilitate this research, we use Vite, which is originally developed at Stanford's Center for Integrated Facility Engineering as a benchmarking application for the evaluation of PSL and SimQL.

Progress and Results

Our first gaol in this project is to evaluate PSL as process specification interchange standard using Vite as a benchmark application. 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. It takes traditionally qualitative organizational management theory and builds a model that incorporates rough quantitative measures. As for this investigation, we have built a sample demonstration using PSL as an interchange format to exchange information between Primavera's P3 and Vite. (Primavera is a project scheduling software widely used in the construction industry.) Presently, the demonstration application includes four translators: Vite to PSL(KIF) translator, PSL(KIF) to P3 translator, P3 to PSL(KIF) translator and PSL(KIF) to Vite translator. The translation process between P3 and Vite using PSL is summarized as shown in Figure 1.

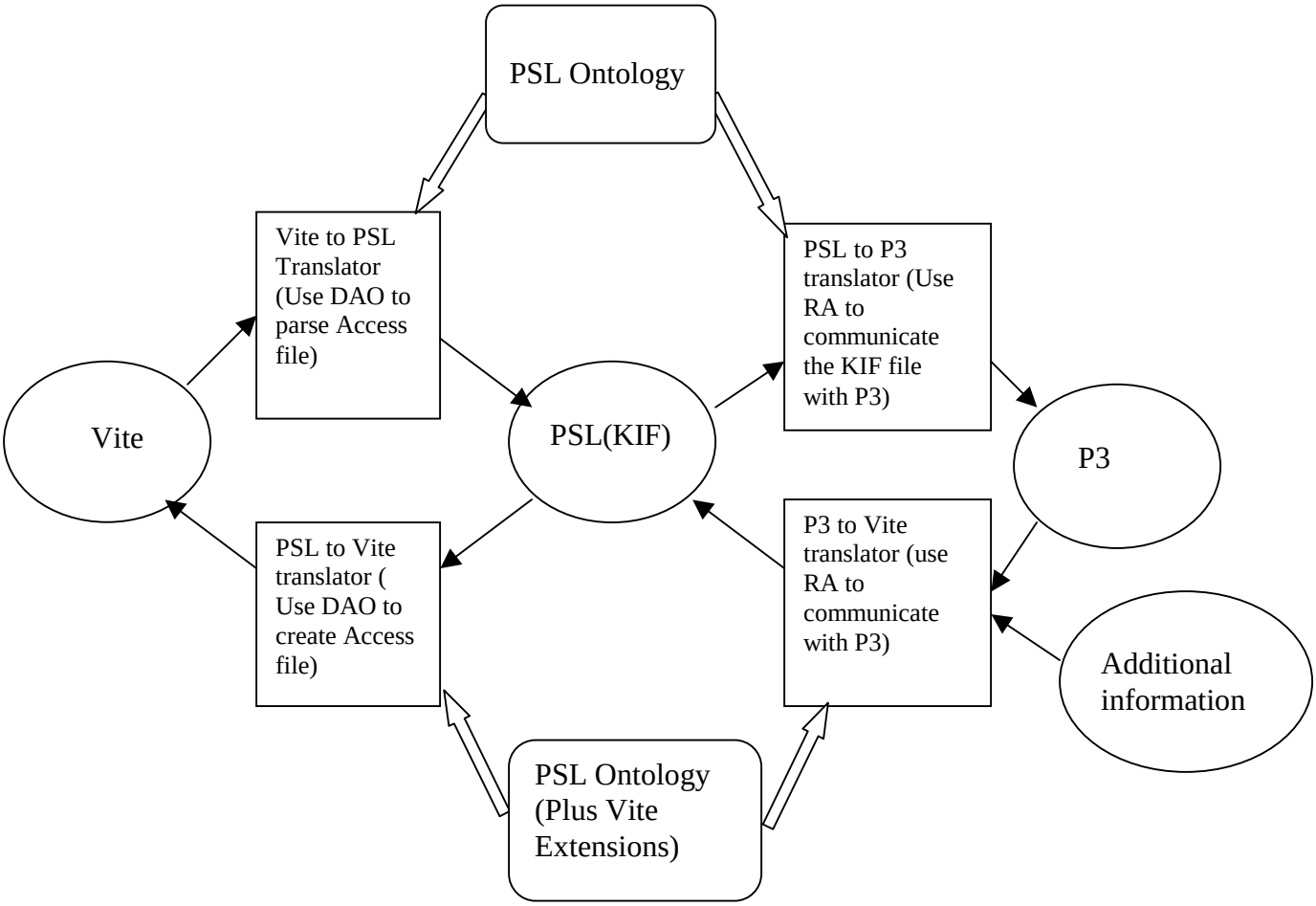

Figure1: Information Exchange between Vite and P3 Using PSL

- The information to be transferred from Vite to P3 consists primarily activity schedules. As discussed in the previous report, activity scheduling information is adequately covered in the current PSL ontology definitions.
- Transferring information from P3 to Vite, however, needs additional information besides activity scheduling, for example, organization, communication and skills, etc. These concepts are not available in the current PSL ontology definitions. As discussed in this report, extensions to the current PSL ontology are added to achieve the information exchange process.

In the following, we discuss the extensions of PSL ontology to capture some of the Vite's concepts to facilitate the process information exchange between Vite and P3.

Extensions to PSL Ontology

An ontology is an explicit specification of some topic. In this work, ontology is defined as a formal and declarative representation which includes the vocabulary (or names) for referring to the terms in a subject area and the logical statements that describe what the terms are, how they are related to each other, and how they can or cannot be related to each other. Ontologies therefore provide a vocabulary for representing and communicating knowledge about some topic and a set of relationships that hold among the terms in that vocabulary.

Current Status of the PSL ontology

The development of PSL has proceeded primarily on an as-needed basis. The initial PSL ontology was developed using a single EDAPS (Electromechanical Design and Planning System) scenario. Later PSL ontology was further expanded to incorporate the concepts introduced in various manufacturing software applications when PSL was used to exchange process information among these packages. Currently PSL ontology could be summarized in Figure 2.

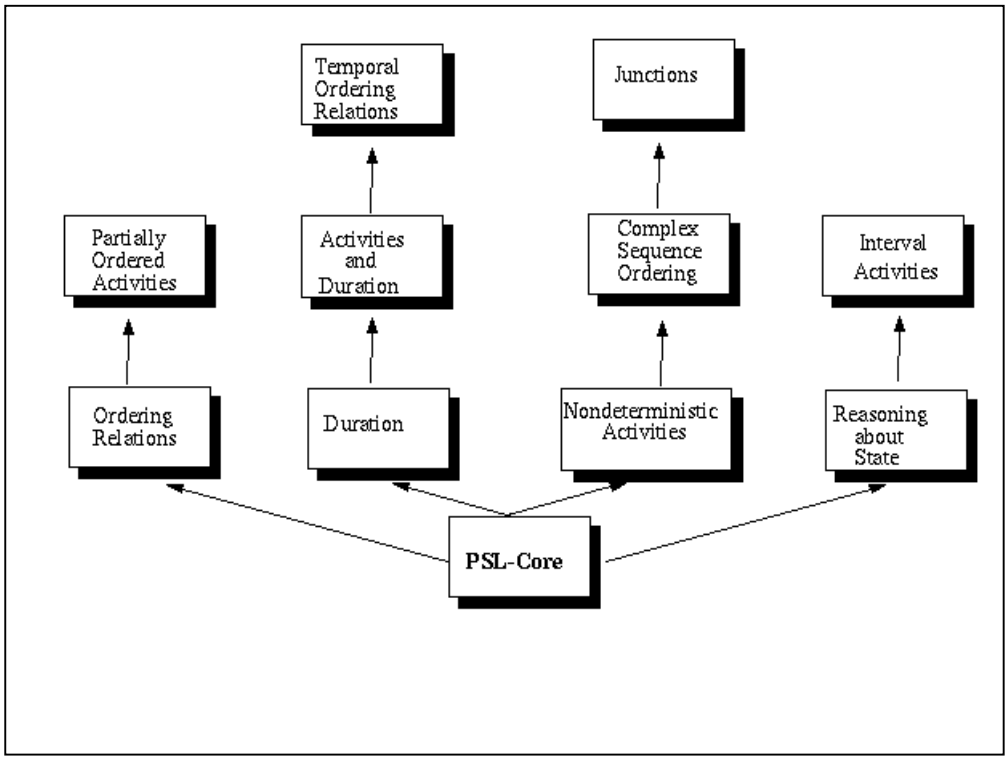

Figure 2: Current PSL Modules

Extension on current PSL Ontology for Information Exchange between Vite and P3

PSL is designed to facilitate communication of process information among the various applications. When two programs need to exchange process information, they not only need to agree on a representation language for the interaction (such as PSL), but also need to agree on an ontology in their domain. Ontologies do not change during problem solving, so they are intended to support multipile tasks and methods and are particularly suited for "compiling" into tools. The key idea here is that PSL (KIF) needs to provide a common ontology that P3 and Vite both agree with. The ontology needs to cover the key concepts (ontology) both in P3 and in Vite.

After analyzing the concepts in Vite and building the sample demonstration application, the following extension to the current PSL ontology has been added to facilitate information exchange between Vite and P3:

- Organization (which includes Actor and Group)
- Vite Activities
- Project

Figure 3 shows the additional ontology modules with relation to existing PSL ontology. The extensions are described in Appendix A.

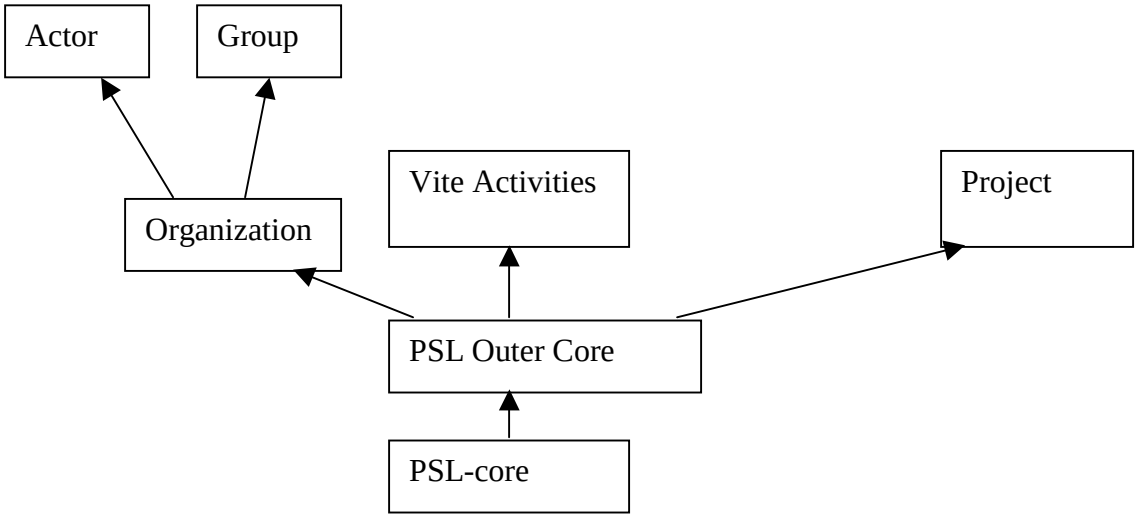

Figure 3: Extensions to Current PSL Ontology

Information Exchange between P3 and Vite

A Vite project is composed of a traditional CPM diagram, additional links showing failure dependence and reciprocal information dependence, a management structure diagram, and responsibility links between the management structure and activities. In

this example scenario, we focus on interchanging activity information between Vite and P3. A typical CPM activity diagram is shown in Figure 4. Activities are named and given durations, and groups of activities are linked together using relationships such as finish-start, start-start, or finish-finish. Milestones may be included in the activity diagram, such as "ship tapes to foundry" in the example project. Begin and end milestones are used to denote the start and finish of the project ("Start Project" and "Fab, Test and Deliver.")

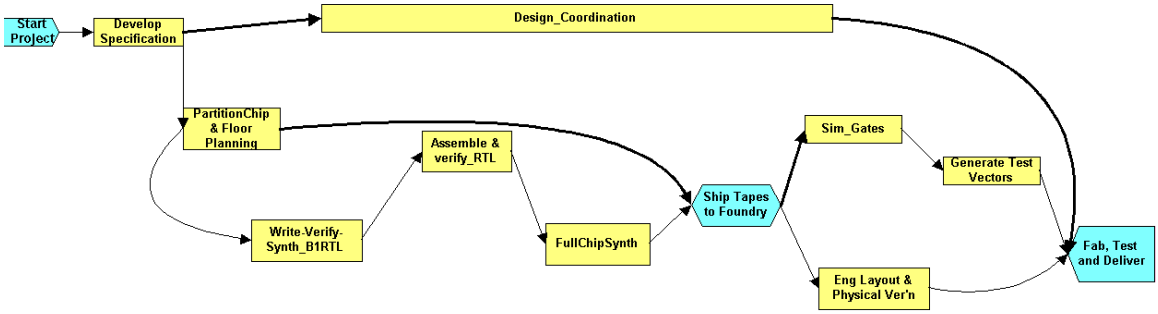

Figure 4: Traditional CPM diagram in Vite

Currently, Vite does not provide any API. The simulation results of Vite are stored in an Access database file format. Vite exports project information to an Access database, which contains the following relational tables: ViteActivities, ViteActivityStatistics, ViteActorCrafts, ViteActors, ViteActorStatistics, ViteActorTrace, ViteAssignments, ViteDependencies, ViteMeetings, ViteProjects, ViteProjectStatistics, ViteProjectSuccessors, ViteReciprocals, ViteScenariors, ViteScenarioStatistics, ViteSuccessors, ViteSupervisions, and ViteTeams.

- For the Vite to PSL(KIF) translator, we map the concepts in Vite into the formal ontology described in PSL, that explicitly and unambiguously defines all terms introduced within the language. Then we parse relevant information stored in the Access database using DAO (Data Access Object), translate the information into PSL(KIF) according to a set of rules, and create a PSL(KIF) file.

- For the PSL(KIF) to P3 translator, we use RA (Primavera Automation Engine) to communicate with P3. 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 parse the activity information stored in PSL(KIF) format and to communicate with P3. Finally, P3 re-constructs the project schedule, based on the PSL(KIF) formatted information.

- Although P3 has much more information about activity scheduling, it does not contain other information such as project organization needed by Vite. For the P3 to PSL(KIF) translator we provide additoinal information in a plain text file. The translator maps the concepts and additional information into PSL ontology (including ontology extensions), and then generates the PSL file.

- For the PSL(KIF) to Vite translator, the information in the PSL file is parsed and rewritten into VNB (Access database) file format. Vite could open the VNB file and start simulation. With the extended PSL ontology to cover the concepts in P3 and Vite, the PSL file can easily be translated into Vite readable format.

The sample demo is illustrated as shown in Figures 5, 6 and 7. Figure 5 shows the Ghant chart for the Vite project activtities shown in Figure 4 (without considering the effects of communication, activity failure, and etc.). Vite to PSL(KIF) translator generates a PSL file, and the scheduling information is produced by P3 as shown in Figure 6. Using the P3 to PSL(KIF) translator, the project information in P3 (as shown in Figure 6) is parsed and combined with additional information to generate a PSL file. Finally, Vite starts simulation, taking into consideration the additional communication and activity failure information, and produces the refined Ghant chart shown in Figure 7.

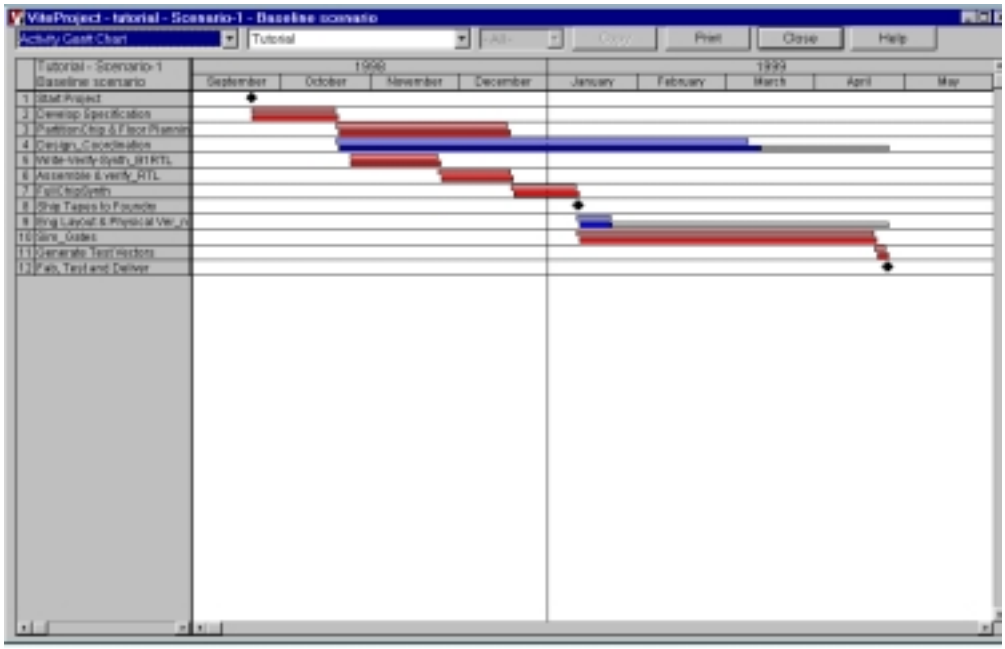

Figure 5: Vite's Activity Ghant Chart (Initial Chart)

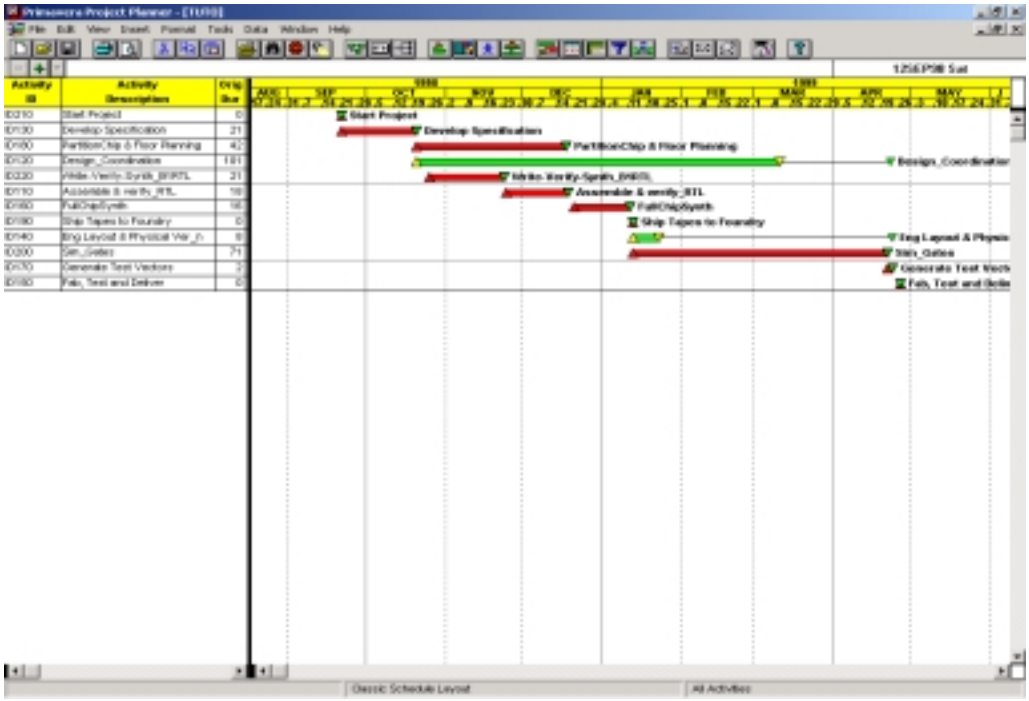

Figure 6: Schedule Chart Re-Produced by P3

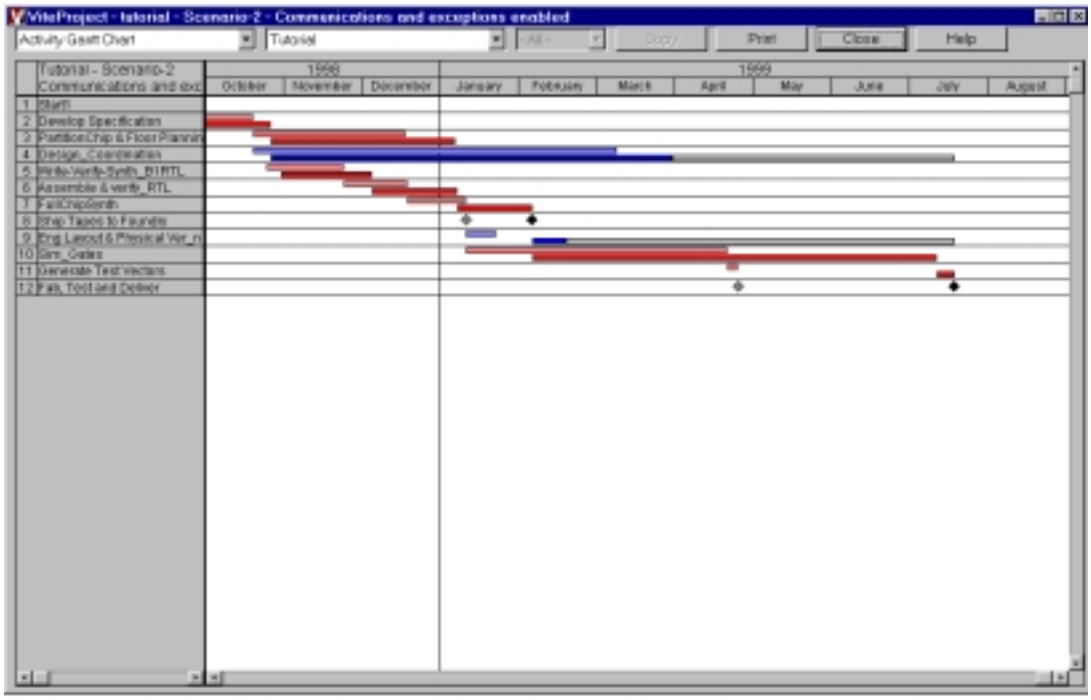

Figure 7: Vite's Activity Gant Chart (After Re-Simulation)

Future Work

Our next step is to investigate the potential of SimQL. SimQL is a new and promising simulation access language to wrap existing simulation programs. We have done some initial investigation on SimQL and we are in the process of re-compiling the software. Our plan is to use SimQL to wrap some typical software in construction industry such as P3 and Vite. The application of SimQL in conjunction with PSL could be very useful. In essence, Vite is a program that could be wrapped using SimQL, provided all the necessary input data could be gathered in advance. Using SimQL to wrap some sample programs in construction industry such as P3 and Vite, we could understand more about the applicability of SimQL in practical engineering problems. Also if we could use SimQL to wrap P3 or Vite successfully, it could further evaluate the potential of PSL as an interchange language.

Appendix A

Ontology is described by a vocabulary of non-logical symbols and a set of axioms. Here we try to develop an ontology in construction process domain based on the analysis of concepts in Vite and our sample demonstration application. Some of the ontology extensions are essential to PSL, while others are specific to Vite. Here we add an asterisk (*) before the Vite specific extensions.

1. Organization Ontology

The organization ontology focuses on organization structure, roles, authority and empowerment. An organization can be individual or group of individuals to which organizational attributes and relations are associated.

1.1 Actor

Actor is an individual person in a construction organization associated with certain attributes. It has the following attributes:

- Role: The specific position the actor acts within the organization
- Experiences : The practical experience one actor has.
- Skill : Skills possessed by the actor
- SkillLevel: (low, medium, high) skill level for a particular skill possessed by the actor
- *FTE : The actor FTE sets the number of full-time equivalent (FTE) people that this actor has available to perform activities.
- Cost : cost per unit time for the actor's work

Actors have the following Arguments and Relation:

Relation	Arguments	Informal Definition
Role	actor, object(role)	The actor acted as a particular role in the organization

Experience	actor, object(experience)	The actor has certain experience.
Skill	actor, object(skill), object(skill-level)	The actor has a skill with certain skill-level.
*FTE	actor, number	The actor has certain FTE to perform activities.
Cost	actor, number	The actor's cost is certain number per unit.

1.2 Group(team)

A Group is a subclass of Organization-Entity and is itself composed of a set of Organization-Entities. A Group consists of Organization-Members (e.g. manager, senior engineer, etc.) which are actors themselves:

- Member: Individual actor within one Group
- Experience: The experience the group has as a whole.
- Director: The member who is responsible for the group.
- Size: The number of individual members in the group

Relation	**Arguments**	**Informal Definition**
Member	group, acotr	The actor is a member of the group.
Experience	group, object(experience)	The group has certain experience as a whole.
Director	group, actor	The actor is the director of the group.
Size	group, number	The group has certain number of members

2. Vite Activity Ontology

Vite Activity is an activity in a construction process associated with certain attributes. It has the following attributes:

- Priority: The relative importance of this activity to other activities.
- Uncertainty : The number of other activities to which this activity has information dependency relationships determines the Uncertainty value. It reflects the effect that other activities can have on this one.
- *RequirementComplexity : Complexity rating that depends on the number of internal requirements this activity must satisfy
- *SolutionComplexityc : Complexity rating that depends on the number of solutions to which the activity contributes.
- Required skill – Skill required to complete the activity quickly
- Dependency : Activity dependency indicate when failure in an indicated successor activity might result in rework for the specified activity.

- Reciprocal : Activity reciprocal indicates when communication is required between the managers of the related activities.

Relation	Arguments	Informal Definition
Priority	activity, number	The activity has a specific priority
Uncertainty	activity, number	The activity has a specific uncertainty.
*RequirementComplexity	activity, object(Requirement Complexity)	The activity has a specific requirement complexity rating.
*SolutionComplexity	Activity, object(SolutionComplexity)	The activity has a specific solution complexity rating.
Dependency	activity, activity, number	One activity is depended on another activity with certain strength.
Reciprocal	activity, activity	Communication is required between the two activities.
RequireSkill	activity, object(skill)	The activity requires a specific skill.

The following axioms hold for dependency and reciprocal relation.

Axiom 1: The dependency relationship is transitive.

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

Axiom 2: Reciprocal relationship is symmetric.

```
(forall (?a1 ?a2)
        (=> (reciprocal  ?a1 ?a2)
                (reciprocal ?a2 ?a1) ) )
```

3. Vite Project Ontology

Project ontology extension covers general project information in vite.

- *Centralization : (low, medium, high) Characterization of centralization of decision-making responsibility within the project.
- *Formalization : (low, medium, high) Characterization of formalization of communications in memos or organized meetings
- Information Exchange Probability: The probability of generating a communication request while processing a sub activity.

- Noise Probability: (percentage) Probability that communication is non-activity related “noise”
- Internal Rework Probability: The probability that a sub activity will fail and generate rework within an activity
- Dependent Rework Probability: The probability that a sub activity will fail and generate rework for failure dependent activities.
- Work Day: (hr) Number of working hours per day (i.e. typically 8)
- Work Week: (days) Number of working days per week (i.e. typically 5)

Relation	**Arguments**	**Informal Definition**
*Centralization	project, number	The project has specific centralization level.
*Formaliztion	project, number	The project has specific formalization level.
InfoExchProb	project, number	The project has specific information exchange probability.
NoiseProb	project, number	The Project has a specific noise probability.
InternalReworkProb	project, number	The Project has a specific internal rework probability.
DependentReworkProb	project, number	The Project has specific dependent rework probability
ActivityAssignment	project, activity, actor, number(FTE)	Assign an activity to a specific actor with certain FTE.
WorkDay	project, number	The project typically works on certain hours per day.
WorkWeek	project, number	The project typically works on certain days per week.