Towards a unified specification of the construction process information : the PSL approach

A.F. Cutting-Decelle
University of Savoie/ESIGEC/LGCH, Chambéry, France

C.J. Anumba, A.N. Baldwin, N.M. Bouchlaghem
University of Loughborough, Loughborough, UK

M. Gruninger
University of Toronto, Toronto, Canada

ABSTRACT: As the use of IT in manufacturing or construction has matured, the capability of software applications to interoperate has become increasingly important. Standards-based translation mechanisms have simplified integration by requiring only a single translator. This challenge is especially apparent for process information, used by many software applications, each in a different way. The primary difficulty is that they sometimes associate different meanings with the terms : both their semantics and their syntax need to be considered when translating to a neutral standard.
The Process Specification Language (PSL) creates a neutral, standard language for process specification to serve as an interlingua to integrate multiple process-related applications. This interchange language is unique due to the formal semantic definitions (the ontology) that underlie the language.
The aim of this paper is to analyse the « *applicability* » of PSL to the construction sector through the example of a generic construction process.

1 INTRODUCTION

In all types of communication, the ability to share information is often hindered because the meaning of the information can be drastically affected by the context in which it is viewed and interpreted (Cutting1 2000). This is particularly true in construction, since, in addition to the complexity of information and problems of information exchange, there are additional problems due to the number of the actors involved in the construction process and the diversity of the information handled during the building's life cycle (design, construction, operation of the building). This results, at the engineering design level, in a poor understanding of the contents of the messages between the actors and with the site.

At the same time, there is an increasing need for improvement of the conventional design and construction process in the construction industry, mainly related to the poor performance commonly associated with building projects (Cooper 1998).

Recently, there has been an increased need for the development of process modelling concepts, particularly in terms of how *Information Technologies* (IT) are used and could be used to support the overall life-cycle process (Karhu 1999). A generic process modelling method would be suitable for describing the building construction process from the different points of view whose integration defines the whole project.

A first stage in the integration of the construction information lies in the development of a common language enabling all the actors of the construction process to share the same semantic concepts intrinsic to the capture and exchange of information used in the process.

Motivated by this growing need to share information process in the manufacturing environment, the PSL project is aimed at providing a generic *Process Specification Language,* as a language focused on the description of process, building on existing modelling methods. Originally developed for a manufacturing environment, and given the similarities between this sector and the construction sector, there is a need for testing the applicability of the PSL concepts to the representation of the construction process.

2 TOWARDS A GENERIC MODEL OF THE CONSTRUCTION PROCESS

2.1 *The concept of "process"*

The concept of process lacks a commonly agreed definition. A typical definition is : « *a set of partially* *ordered steps intended to reach a goal* » (Humphrey 1992, referred in : Koskela 1995).

There are four common perspectives to processes (Curtis 1992) :

- Functional : representing what process elements are being performed, and what flows connect these elements ;
- Behavioral : representing when process elements are performed, and how they are performed through feedback loops, iteration, decision making conditions, etc. ;
- Organisational : where and by whom process elements are performed ;
- Informational : a perspective of the informational entities produced or manipulated by the process.

In the functional view, processes consist of activities, that together achieve the purported goal. In addition, auxiliary concepts such as artifacts (products of activities) can be used for process representation.
In a behavioral perspective, processes may consist of precedence relations or information and material/ information flows, with the time explicitely represented. Flow process concepts focus on what happens to material and information in timeline.
In an organisational perspective, processes may consist of agents (performing activities) and roles (set of activities assigned to an agent). Also, the process may be viewed as composed of a supplier-customer partnership.
In an informational perspective, processes consist of data, objects, documents, etc.
In principle, these perspectives, when combined, produce a complete model of a process. However, in current practice of process modelling, the functional perspective (as provided by SADT method) often dominates : activity is seen as the basic construct, and this process concept only achieves one goal : « how to obtain the result ».
Of course the answer to this question is sufficient for achieving the process ; however, it does not exhaust all improvement potential. There are two other relevant goals, that should generally be tackled : how not to consume unnecessary resources (Koskela 1995) and How to ensure that the result corresponds to requirements. In order to achieve these goals, contributions from behavioral and organisational perspectives are needed. We will see in this paper that, as a language, PSL is capable of expressing these different points of view, once they have been represented using modelling methods dealing with these multiple views (such as IDEF3, for example).

2.2 *An approach of the construction process*

Information handled during the construction process can be divided into several categories (Björk 1992) :

- First, information must state facts : such as design documents, which are the results of design decisions. Information to be transferred between computing systems in the construction process is mostly of this type. This information has also to define goals and requirements which a particular project must fulfil. The third category of information states rules which restrict facts, but which apply in general and are not tied to a particular project. These three categories of information can be called « facts », « constraints » and « knowledge ». From a programming language point of view, facts can be constructed using assignment statements, requirements are mainly represented by inequality operators (or algorithms) and knowledge through knowledge based systems ;
- The second point provides a semantic approach dividing information into project-specific and more general information. Facts can be both project specific and general. Constraints are mainly project-specific and knowledge is usually general in nature ;
- The third point of view concerns the presentation and categorises the types of documents used to present the information for human interpretation. Some typical presentation formats used in construction are : drawings, schemas, realistic visualisations, written specifications, calculation results, bills of materials, contracts, orders and various tendering documents.

We will here limit our study to project-specific information, focusing on the semantics of the information. The reason of this choice comes from our primary concern to study information management within construction projects. The information to be communicated to other parties in the construction process mostly consists of factual information. Clearly constraints are very important in the early briefing stages of projects and in quality assurance applications. Knowledge mainly resides in application programs and its effect on the actual transfer of data between project participants will need to be examined further.

2.3 *Modelling of the construction process*

Several process models have been developed in the domain of construction, among which the MoPo model (Cooper 1998), covers the whole construction life cycle. Other models mainly focus on the design stage such as the ADePT model described in (Austin 1996). Some process models introduce concurrent engineering features, such as the model presented in

(Anumba 1996), or client requirements (Kamara 2000).

It is interesting to make a synthesis of the common features of these models, thus leading to a generic process representation. One of these generic representations of the construction process, as provided by (Björk 1992), consists in three main categories, which are : *activities*, *results*, *resources*. An *activity* uses *resources* to produce *results*. Traditional construction classification systems often tend to equate results to buildings and their parts. This is due to a desire to distribute total construction costs over building parts, which is useful for cost analysis purposes. It is, however, evident that information (mostly delivered as documents) and services are other important sub-types of results.

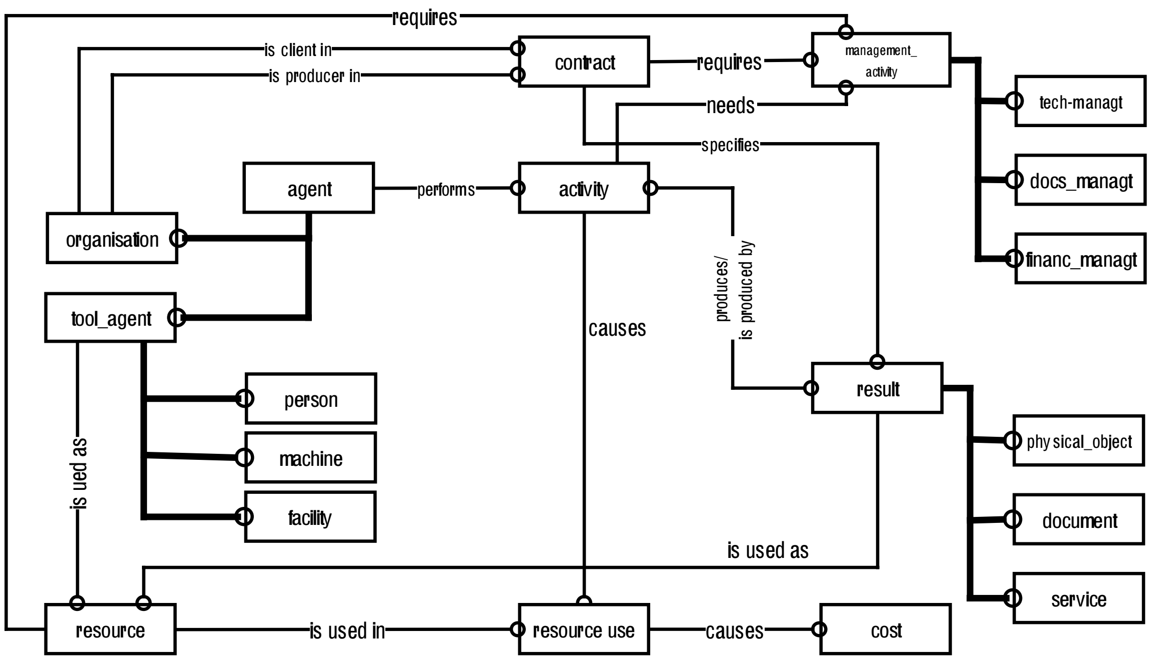

Figure 1 : EXPRESS-G diagram of a generic representation of the construction process (Björk 1992)

The schema of the Fig. 1 (adapted from Björk 1992) gives an EXPRESS-G (ISO10303-11 1994) representation of some objects of a generic construction process. These include :

- **Activity** : the kernel of the model, with relationships with most of the other objects of the model : an activity may have relationships with the result it produces, the resources it uses and the agents performing it.
- **Result** : an example of an entity type needed for classification purposes, but which is intermediate in nature since most of the relevant information about results will be defined in the class descriptions of the sub-types of results.
- **Physical _object** : any physical object with shape and location. Both characteristics may be dynamic.
- **Service** : results of activities which are not physical objects or documents (information), such as « guarding the site ».
- **Agent** : any organisation, person, machine, or facility which participates in the activities of the project. An agent performs some activity in the construction process. A fundamental aspect of an agent (distinguishing him from a product) is that he/it has an existence outside the project and usually participates in several projects.
- **Resource use and cost** : each activity in the project demands a number of inputs in the form of resources which are « consumed » or used. The actual use may be measured in manhours, tons, squaremeters, etc. A clear distinction is made between resource use and the resources entity in itself, which may be documents, materials, machines or persons. The use of any resource involves a cost, which in most cases can be measured by the amount of the resource consumed, but in some cases by the opportunity cost of that resource, that is the cost the customer accepts to pay for the resource in question.
- **Management_activity** : a super-type of the different types of management related to a construction project, such as technical management (planning, logistics, QS activity), document management (drawings, bills of quantities, calculation notes) and financial management.

This model of the construction process will provide the basic example on which the PSL concepts will be mapped in this paper.

3 THE PSL (PROCESS SPECIFICATION) LANGUAGE

The objective of PSL (initially developed by the National Institute of Standards and Technology, NIST, US) is to create a process interchange language 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. This representation would facilitate communication among the various applications because they would all have a common understanding of concepts to be shared. This language is currently being standardised at the international level, by the ISO TC184/SC4 committee (« *Industrial Data* »), now at the level of Provisional Work Item ISO 18629 (ISO 18629-1 2000).

PSL specifies a language for the representation of process information, limited to the realm of discrete processes related to manufacturing, including all processes in the design/manufacturing life cycle. Business processes and manufacturing engineering processes are included in this work both to ascertain common aspects for process specification and to acknowledge the current and future integration of business and engineering functions.

The goal of the project is to create a process *specification* language, not a process *characterization* language. A process specification language is a language needed to specify a process or a flow of processes, including supporting parameters and settings. This may be done for prescriptive or descriptive purposes. The language is composed of an ontology and one or more presentations.

This is different from a process characterization language, which can be defined as a language describing the behaviors and capabilities of a process independent of any specific application (process *modelling* language). For example, the dynamic or kinematic properties of a process (tool chatter, a numerical model capturing the dynamic behavior of a process or limits on the process performance or applicability), independent of a specific process, would be included in this characterization language.

PSL is a neutral language for process specification serving as an interchange language to integrate multiple process-related applications throughout the manufacturing process life cycle (from initial process conception all the way through to process retirement). This project is related to, and in many cases working closely with, many other efforts (Schlenoff 2000). These include individual efforts (single company or academic institutions) such as A Language for Process Specification (ALPS) Project, the Toronto Virtual Enterprise (TOVE) Project, the Enterprise Ontology Project, and the Core Plan Representation (CPR) Project. In addition, the PSL project is in close collaboration with various projects (involving numerous companies or academic institutions) such as Shared Planning and Activity Representation (SPAR) Project, the Process Interchange Format (PIF) Project, and the WorkFlow Management Coalition (WfMC). Most of these efforts have been taken into consideration in the development of PSL, the language benefitting from the experience gained through the different projects analysed in the study (Knutilla 1998).

The primary component of PSL is an ontology designed to represent the primitive concepts that, according to PSL, are adequate for describing the basic manufacturing, engineering, and business processes. An *ontology* is lexicon of specialized terminology along with some specification of the meaning of terms in the lexicon. Note that the focus of an ontology is not only on terms, but also on their meaning. We can include an arbitrary set of terms in our ontology, but they can only be shared if we agree on their meaning. It is the intended *semantics* of the terms that is being shared, *not simply* the terms.

The challenge is that a framework is needed for making the meaning of the terminology for ontologies explicit. Any intuitions that are implicit are a possible source of ambiguity and confusion. For the PSL ontology, we must provide a rigorous mathematical characterization of process information as well as precise expression of the basic logical properties of that information in the PSL language. In providing the ontology, we therefore specify three notions: language, model theory, proof theory.

3.1 *The language*

A language is a lexicon (a set of symbols) and a grammar (a specification of how these symbols can be combined to make well-formed formulas). The lexicon consists of logical symbols (such as boolean connectives and quantifiers) and nonlogical symbols. For PSL, the nonlogical part of the lexicon comprises expressions (constants, function symbols, and predicates) chosen to represent the basic concepts in the PSL ontology. The underlying grammar used for PSL is that of KIF (Knowledge Interchange Format). KIF (Genesereth 1992) is a formal language based on first-order logic developed for the exchange of knowledge among different computer programs with disparate representations. KIF provides the level of rigor necessary to define concepts in the ontology unambiguously, a necessary characteristic to exchange manufacturing process information using the PSL Ontology.

3.2 *Model theory*

The model theory of PSL provides a rigorous, abstract mathematical characterization of the semantics, or meaning, of the language of PSL – an abstract representation of the primitive concepts of PSL. This representation is typically a set with some additional structure (e.g., a partial ordering, lattice, or vector space). The model theory then defines meanings for the terminology and a notion of truth for sentences of the language in terms of this model. Given a model theory, the underlying theory of the mathematical structures used in the theory then becomes available as a basis for reasoning about the concepts intended by the terms of the PSL language and their logical relationships, so that the set of models constitutes the formal semantics of the ontology.

3.3 *Proof theory*

The proof theory of PSL is perhaps its most important component. It consists of three components : PSL Core, one or more foundational theories, and PSL extensions :

- PSL Core : PSL Core is based upon a precise, mathematical first-order theory, a formal language, a precise mathematical semantics for the language and a set of axioms that express the semantics in the language. There are four primitive classes, two primitive functions, and three primitive relations in the ontology of PSL Core. The classes are OBJECT, ACTIVITY, ACTIVITY_OCCURRENCE and TIMEPOINT. The four relations are PARTICIPATES-IN, BEFORE, and OCCURRENCE-OF. The two functions are BEGINOF, and ENDOF. ACTIVITIES, ACTIVITY_ OCCURRENCES, TIMEPOINTs (or POINTs for short), and OBJECTs are known collectively as entities, or things. These classes are all pairwise disjointed.

- Core Theories : The purpose of PSL Core is to axiomatize a set of intuitive semantic primitives that is adequate for describing basic processes. Consequently, its characterization of them does not make many assumptions about their nature beyond what is needed for describing those processes. The advantage of this is that the account of processes implicit in PSL core is relatively straightforward and uncontroversial. However, a corresponding liability is that the Core is rather weak in terms of pure logical strength. In particular, the theory is not strong enough to provide definitions of the many auxiliary notions that become needed to describe an increasingly broader range of processes in increasingly finer detail. For this reason, PSL includes one or more *core theories*. A core theory is a theory that axiomatizes new primitive concepts not found in PSL-Core, but which are needed to provide rigorous semantics for other terms in PSL.

- Extensions : The final component of PSL consists of PSL *extensions*. Roughly speaking, a PSL extension gives one the resources to express information involving concepts that are not part of PSL core. Extensions give PSL a clean, modular character. PSL core is a relatively simple theory that is adequate for expressing a wide range of basic processes. However, more complex processes require expressive resources that exceed those of PSL core. Rather than clutter the PSL core itself with every conceivable concept that might prove useful in describing one process or another, a variety of separate, modular extensions have been (and continue to be) developed that can be added to PSL core as needed. In this way a user can tailor PSL precisely to suit his or her expressive needs. To define an extension, new constants and/or predicates are added to the basic PSL language, and, for each new linguistic item, one or more axioms are given that constrain its interpretation. In this way one provides a « semantics » for the new linguistic items. A good example of such an extension is the theory of timedurations below. The PSL core itself does not provide the resources to express information about timedurations. However, in many contexts, such a notion might be useful or even essential. Consequently, a theory of timedurations has been developed which can be added as to PSL core, thus providing the user with the desired expressive power.

3.4 *Current structure of the language*

To date, the language is built on the following two categories of extensions :

- PSL core and outer-core (small set of extensions that are so generic and pervasive that they have been put apart), introducing primitive concepts of the language : core, activity occurrences, atomic activities, complex activities, occurrence trees, activity performing, subactivity occurrence ordering, integer and duration, resource requirements theory, resource sets ;
- PSL extensions, introducing new definitions : ordering relations (complex sequences), nondeterministic activities, ordering relations over activities, junctions, duration, reasoning about states, interval activities, states, temporal ordering relations, reasoning about resource divisibility, resource roles, reasoning about resource usage, capacity-based concurrency, substitutable resources, fixed resource sets, homogeneous resource set, inventory resource sets, resource pools, resource set-based actions, resource paths, processor actions.

All the extensions are written using the KIF syntax, under the form of basic axioms and related definitions, in the following way : (excerpt from Occurrence trees extension : occ_tree.th)

Occurrences form a tree.

```
(forall (?a1 ?a2 ?occ1 ?occ2)
     (=> (=   (successor ?a1 ?occ1)
              (successor ?a2 ?occ2))
          (and   (= ?a1 ?a2)
                 (= ?occ1 ?occ2))))
```

4 APPLICATION OF PSL IN CONSTRUCTION

To date, PSL is mainly used in the manufacturing environment. One of the aims of our research work is to provide elements of convergence between the two sectors, manufacturing and construction, through a common representation of process information.

4.1 *Method of work*

The application of PSL to the construction sector requires several stages described here on the basis of the example in the Fig. 1. These stages are not necessarily sequential (Cutting2 2000).

- *Stage 1 : Identification of extensions relevant to the processes to be represented in PSL* : since the beginning, the development of extensions has proceeded on an « as-needed » basis ; the initial PSL ontology was developed using a single scenario, the EDAPS (Electromechanical Design and Planning System) scenario developed by Steve Smith at the University of Maryland (CIM). The concepts introduced were defined and modelled within PSL and later extended as other scenarios were explored, such as the pilot implementation in the domain of scheduling between ProCAPP (KBSI) and the Scheduler 4.3 software (ILOG). If there is not any extension corresponding to the concepts to be expressed with PSL, it may be necessary, either to adapt the process model and/or its representation, or else to develop new extension(s).

- *Stage 2 : Elaboration of a synthesis model of the construction process* : generic enough to encompass the different stages of the construction and the diversity of the actors involved, but precise enough to provide elements of information suited to the interoperability among categories of software corresponding to different stages of the building life cycle and usually considered as incompatible in terms of data representation.

- *Stage 3 : Expression of the synthesis model using a formalism already identified as such within PSL* : this stage is not mandatory, however the work of translating a process model into PSL is largely simplified. This is the case when using the IDEF3 process representation (Mayer 1995), since several PSL extensions make use of elements coming from this method. However, a research work conducted by (Cioccoiu 1998) provides the possibility of a nearly direct translation from IDEF3 to PSL.

- *Stage 4 : Translation of the concepts expressed in the process model into the PSL process language.*

Applied to the example in the Fig. 1, considered as a first approach of a generic process model, the IDEF3 representation of the process model shows the following features, depicted in the Fig. 2 below. This paper presents only the early stages of the work.

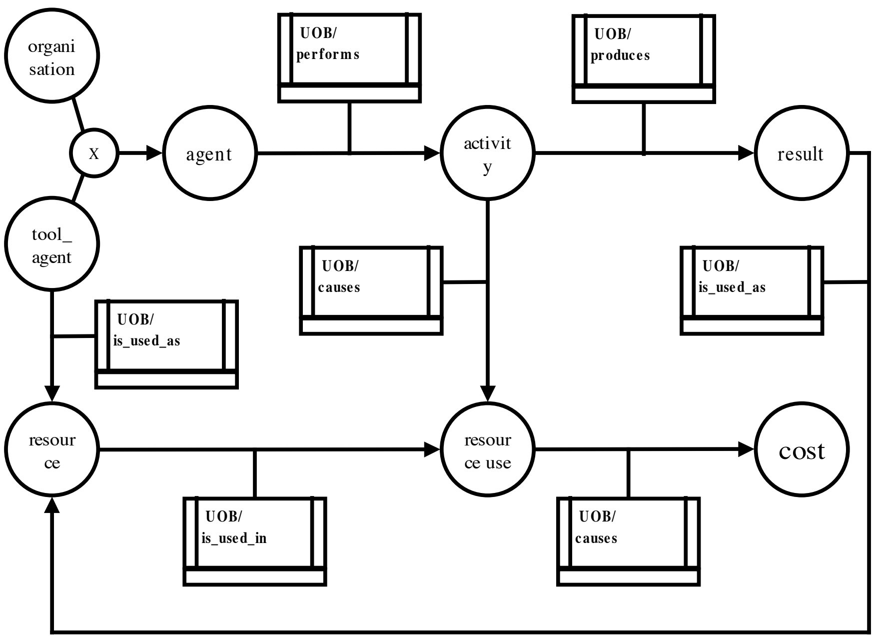

Figure 2 : Enhanced transition schematic representation of a part of the Fig. 1 using IDEF3

Note : in order not to get a too complex schema, all the elements in the Fig. 1 have not been represented in the Fig. 2.

Then, the following stage will consist of expressing the concepts of the schema into the IDEF3 language, corresponding to the graphical representation : each item of the graphics can be expressed in the IDEF3 language, such as : UOB (Unit of behaviour) and UOB-use declarations, processes, links, junctions.

The PSL representation of an IDEF3 schematic is a set of KIF sentences that define a PSL theory. The translation process can be described by a set of meta-theoretic *compilation rules* that associate KIF sentences with the IDEF3 constructs. Writing such compilation rules can also be seen as providing a formal, declarative semantics into PSL to IDEF3 constructs.

Once the compilation rules are written, implementing the translator becomes possible, using for example the `lisp` macros provided by the compilation. It can also be possible to provide the compiler with a KIF expression simplifier, leading to a simple syntax using typeless quantifiers and standard operators.

4.2 *Benefits of the PSL approach in construction*

- An important feature of this approach is related to *the role of PSL as « interlingua »* : through the ontology on which the language is built, it becomes possible to find, for as many concepts as mentioned in the extensions, a common generic representation, thus enabling exchanges of information among software applications traditionally non-interoperable.

The PSL approach thus contributes to the definition of interoperable process models, providing rules enabling this interoperability. Then, the integration of these rules may lead to a general methodology of development of synthesis process models.

This point is very important, since many process models have been developed, some of them are already implemented in software tools, but none of them are, to date, able to directly exchange process-related information.

- It is also interesting *to compare the PSL approach with other synthesis approaches* : here we will base our analysis on the example of the GEPM (generic process model) provided by the MoPo project, since it is one of the more developed to date.

First of all, the final objective is not the same : « the MoPo project aims at providing new methods, reference process models and (IT) tools to support construction process improvement through systematic process analysis and design/planning using construction modelling approach » (Karstila 1999). The expected results of the project are, among others, to develop a « generic construction process information model capturing information requirements for construction process modelling, reference construction process models to be re-used in e.g. company specific modelling efforts for process descriptions ».

In this kind of project, the main efforts concern the development of (synthesis) process models.

However, while MoPo is aimed at providing process *modelling tools*, an initiative such as PSL targets the *litteral expression* to be given to these models (with the final objective of computer-based exchanges among different categories of software, with different and most of the time incompatible data representations).

In other words, it is also possible to say that there is no concurrency in their use : PSL, as other process *specification languages* will act downstream with respect to modelling methods/tools such as the MoPo initiative.

Besides, the use of generic process modelling tools can be considered as a mandatory step of a process representation approach using PSL, since the genericity of these tools contributes to guarantee the validity of the results provided by PSL !

The added value of the PSL approach comes from its « *interlingua* » feature, however, it also comes from the synthesis models on top of which the language is built !

5 PERSPECTIVES OF THE WORK

One of the stages of the on-going work is to make a synthesis of the different approaches in terms of process modelling currently available, these include : the IRMA (Information Reference model for AEC) initiative (Froese 1993), the Unified approach model (Björk 1992), Process Protocol II (PPII 1999), ADePT methodology and tools (Austin 2000), MoPo project (Karstila 1999), DFD model of a construction company operation (Fisher 1992), COSMOS and STAR project (Hannus 1999), Model of construction process information (Froese 1994), various papers dealing with process models, *Organisation of information about construction works, Part 2 : Framework for classification of information, ISO DIS 12006-2, 1999*.

The following stage consists of making a synthesis of these models, in order to create a generic process model applicable to several stages of the life cycle of a building (design, construction, operation, maintenance, decommissioning, facility management), identify the milestones of process models for construction (mandatory features, common elements, etc.) and identify the commonalities/differences with manufacturing industry : comparison of requirements of manufacturing/construction industries.

Then, we plan to propose an IDEF3 representation of the model coming from the synthesis work, in terms of process schematics and object schematics.

We shall then analyse the results of the mapping to IDEF3 representation, in terms of loss of information, incompleteness of the model, impossibilities, etc. in order to propose a translation of the IDEF3 representation into PSL, with new extensions if they are necessary.

The validation of the work will be conducted using an example of a real construction project, in order to test the concepts of the example of a real test case.

6 SUMMARY AND CONCLUSIONS

This paper has discussed the applicability of the Process Specification Language (PSL) to the construction industry and used an example to illustrate the representation schema. There is much scope for the use of PSL to facilitate the specification and exchange of process information in the construction industry. The research project on which this paper is based is exploring this and intends to deliver appropriate PSL extensions that will facilitate its deployment in the construction sector.

AKNOWLEDGEMENTS :

The work described in this paper is the subject of a research contract funded by the Engineering and Physical Sciences Research Council (UK) and being conducted in collaboration with Loughborough University (UK).

REFERENCES

Anumba C.J., Evbuomwan N.F.O., 1996. A concurrent engineering process model for computer-integrated design and construction, *in Information Processing in civil and structural engineering design*

Austin S. Baldwin A.N., Newton A., 1996. A data flow model to plan and manage the building design process, *Journal of Engineering Design*, Vol 7, N° 1

Austin S. Baldwin A.N., Baizhan L., Waskett P. 2000. Analytical design planning technique (ADePT) : a dependency structure matrix tool to schedule the building design process, *Construction management and economics* (2000) Vol. 18

Björk B.C. 1992. A unified approach for modelling construction information, *Building and Environment*, Vol. 27, N° 2

CIM http://www.isr.umd.edu/Labs/ CIM/cimcontent.html

Cioccoiu M. 1998. Translating IDEF3 to PSL, *Technical Report,* University of Maryland, 98-63

Cooper R., Kagioglou M., Aouad G., Hinks J., Sexton M., Sheath D., 1998. The development of a generic design and construction process, *European PDT Days*

Curtis B., Krasner H., Iscoe N. 1992. Process modeling, *communications of the ACM*, 35(9)

Cutting-Decelle A.F., Michel J.J. 2000. Representation of industrial information through the joint use of ISO 15531 MANDATE and PWI ISO 18629 PSL : a contribution to the factory of the future, *PDT Europe Conference*, Noordwijk

Cutting-Decelle A.F., Michel J.J., Schlenoff C. 2000. From manufacturing to construction : towards a common representation of process information : the PSL approach, *Concurrent Engineering Conference*, Lyon

Fisher N. & Yin S.L. 1992. Information management in a contractor, *Thomas Telford Ed.*

Froese T. 1994. Developments to the IRMA model of AEC projects*, Computing in Civil Engineering : proceedings of the First congress*, Khalil Khozeimeh Ed. ASCE, Vol.1 June 20-22

Genesereth M., Fikes R., 1993. Knowledge interchange format (Version 3.0) – Reference manual, *Computer science department, Stanford University*, Stanford

Hannus M. 1999. http://cic.vtt.fi/projects/star/star1/brochure.html

Humphrey W.S. & Feiler, P.H. 1992. Software process development and enactement : concepts and definitions. *Tech.Rep. SEI-92-TR-4, Pittsburgh : Software Engineering Institute*. Carnegie Mellon University

ISO 10303-11 1994. Industrial automation systems and integration - Product data representation and exchange, ISO IS 10303-11, Part 11 : Description methods : EXPRESS language reference manual, 1994

ISO 18629-1 2000. Industrial automation systems and integration - Process specification language, ISO WD 18629-1, Part 1 : General overview

Kamara J. M., Anumba C. J. & Evbuomwan N. F. O. 2000. A Process Model for Client Requirements Processing in Construction, Business Process Management Journal (in press).

Karhu V. 1999. Formal languages for construction process modelling, *CEC 99 (Concurrent Engineering in Construction) Conference,* Helsinki

Karstila K. 1999. Models for the construction process - The MoPo-project, *CEC 99 (Concurrent Engineering in Construction) Conference,* Helsinki

Knutilla A., Schlenoff C., Ray S., Polyak S.T., Tate A., Cheah S.C., Anderson R.C. 1998. Process specification language : an analysis of existing representations, *NISTIR report n° 6160*, NIST

Koskela L. 1995. On foundations of construction process modeling, *CIB W78,* Stanford

MayerR.J., Menzel C.P., Painter M.K., de Witte P.S., Blinn T., Perakath B., 1995. Information integration for concurrent engineering (IICE) IDEF3 process description capture method report, *KBSI Inc, AL-TR-1995*

Process Protocol II 1999. http://PP2.dct.salford.ac.uk

Schlenoff C., Gruninger M., Tissot F., Valois J., Lubell J., Lee J., 2000. The process specification language (PSL) Overview and version 1.0 specification, *NISTIR 6459,* NIST