Automation of Process Information Exchange between AEC software Applications using PSL

G. Tesfagaber[1] and N. M. Bouchlaghem[2]

ABSTRACT | This paper presents the result of an academic research study conducted to explore the potential use of the Process Specification Language (PSL) for the automation of process information exchange between Architecture, Engineering, Construction and Facilities management (AEC/FM) computer applications. PSL, one of the many ontology based specification and exchange languages, was developed and standardised as ISO 18629 at National Institute of Standards and Technology (NIST) to bridge the semantic gap between process information representations adopted within the manufacturing applications. The underlying foundation of PSL, the ontology, a formal semantic definition, provides a neutral representation of manufacturing processes' descriptions. Although PSL was initially developed for manufacturing it is established that the language can be applied in construction provided extensions are developed for some of the construction process concepts that cannot be captured within the current PSL ontology. Extensions are proposed, and translation of process information representations between three construction applications is demonstrated.

KEYWORDS | information exchange automation; interoperability; PSL: AEC/FM applications; ontology; cost estimating, scheduling, design

1 Introduction

There is a wide range of computer applications within the construction industry, which deal with the manipulation and management of construction processes descriptions, which is referred to as construction process information. The term "process information" refers to information describing the operations needed to complete a project, including the descriptions of activities, data requirements, ordering relations, temporal constraints, resources requirements, time-duration, abstraction, cost etc [1]. The last decade has seen considerable attention by both researchers and practitioners on understanding construction processes. This combined with the growing attention on supply chain management has resulted in the recognition of a need for improvements in the exchange of process information between different application software packages. Efficient and effective exchange with minimal human interaction is not easy to achieve. Construction organizations typically purchase their applications software from different suppliers. There is no industry 'standard' package that covers different functions. This is exacerbated by the fact

1. PhD, candidate, Dep. of Civil and Building Engineering, Loughborough University, Loughborough, LE11 3TU, tesfagabers@yahoo.co.uk
2. Professor of Architectural Engineering, Department of Civil and Building Engineering, Loughborough University, Loughborough, LE11 3TU

that the construction industry is highly fragmented. Furthermore each application tends to have unique representations and terminologies.

The Process Specification Language, (PSL) project at the National Institute of Standards and Technology (NIST) has addressed the same problem by creating a neutral, standard language for process specification to server as an Interlingua to integrate multiple process related applications throughout the manufacturing life cycle [2]. To date, PSL has developed a core set of theories that can express a wide range of basic processes together with a set of extensions to provide resources for expressing more complex processes relating to planning and scheduling Research work at Loughborough has investigated the applicability of the language in AEC/FM, and identified the need for additional extensions to the current PSL ontology to incorporate the process specification requirements of the construction industry. The extensions are based on the common process concepts identified by the research. Having established these concepts, multiple processes related software applications are able to inter-operate effectively with minimal human interaction. This provides a formal specification of process concepts for three software applications: computer aided drawing (AutoCAD), cost estimating (CCS), and project planning and scheduling (Microsoft Project). This, it is argued, proves the feasibility of correct process information translation that may be achieved electronically once a complete formal specification of construction process concepts is developed.

This paper presents an overview of construction process specification requirements (process information representations) within the pre-construction stages with particular attention to the three core stages: architectural design, pre-tender cost estimating, and planning and scheduling. An overview of the PSL language is provided. The application of PSL to construction is examined through implementation of the language in a scenario, within the pre-construction stages, involving the representation and translation of process related design, cost estimating, and scheduling information between three well-known application software packages. The translation process adopted provides an effective procedure and identifies the need for new extensions to be introduced into the current PSL ontology to incorporate concepts for construction-specific information. Whilst there were extensions applicable to define the planning and scheduling semantic concepts of project management, there were no extensions capable of defining the semantic concepts for either the design or cost estimating of construction works. Extensions were subsequently proposed for these concepts.

2 Process Specification Requirements within the Pre-construction Stages

The pre-construction process stages are the primary areas in which computer applications are widely used to support the description, and planning of construction processes. The requirements related to the pre-construction stages for specifying construction processes can be identified through the analysis of the software applications supporting these stages. This section briefly looks at the architectural design, pre-tender cost estimating and scheduling application areas of the pre-construction stages and the requirements are gathered through the identification of the data types and data structures/file formats adopted within the software applications selected to support these stages.

Programs supporting architectural design, cost estimating and scheduling used in this research study are: AutoCAD, Construction Computer Software (CCS) and Microsoft project. These applications describe construction processes from the design, cost estimating, and scheduling and planning points of view and were analyzed to identify the data types that are used in each of these applications for process information representations. These packages were selected as test applications for the applicability of PSL in construction, because AutoCAD and Microsoft Project application are currently the most widely used applications in UK construction organizations, and CCS has complex data structures to represent complex construction processes and cost information. The construction scenario within the pre-construction process stages was chosen as a test ground for the trial implementation of PSL in construction for the following relearns: (1) it is in these stages that most of the construction processes are described, (2) these stages require complex process information representation and exchange and, (3) most of the computer applications supporting theses stages use different terms and representations to specify the same information.

3 Process Specification Requirements related to Architectural Design

Architectural design is a process of specifying projects facilities/products such as buildings and their parts, by means of drawings and textual specifications that complements the drawings. This process involves several phases spanning from brief to schematic and detailed design. Schematic design is the phase where the gross features of the product or facility are defined including the overall shape, size, structure, adjacencies, circulation, and the materials of construction, and each feature undergoes refinement and elaboration in subsequent the detailed design phases. Taking the view that projects are unique assemblies of discrete, mostly standardized components, the design and representation of projects information is divided into two components: the production of project objects, using drawings that represent information about various project elements using geometrical representations; and the production of specification, text files, which contain description of these elements. These representations must include the necessary information that can support the assessment of the project from different points of view such as production/construction, cost estimating, and scheduling and planning among many others. The results of this process are: hierarchy of complex products and components to be constructed using hierarchically interconnected and complex construction processes.

AutoCAD is widely used for architectural design within the pre-construction stages, and is well known as a product related application. However in this research study it is analyzed as a process related application as it deals with project information representations including process specification requirements. These requirements include construction products, and data required during the construction/development of the project products; and data required in the pre-construction stages to produce other process related information such as cost estimating and construction plan information. AutoCAD is a general-purpose computer aided drafting tool, initially developed in the early 1980's by AutoDesk Inc [3]. This application consists of a set of AutoLISP applications or routines that provide drafting objects such as text and symbol fonts, line types, shapes, hatch patterns, fill patterns, and geometric characteristic symbols. These objects enable users to create AutoCAD drawing objects such as buildings and their parts, and attach attributes for textual specification, which characterizes the drawing objects. AutoCAD has intelligent representation, which enables the representation of all the different components that make up a project, along with the manner in which they come together. The representation must be geometrically clear and textually informative complementing geometrical representation. AutoCAD uses drawing objects, block object, textual attributes and spreadsheet files for the representation and manipulation of projects' information. This includes design for construction information, and information for cost estimating and planning.

DWG object: AutoCAD drawing objects are the main entities by which a designer captures a compound (composite) and component (detailed) objects or products in a project. **Block-objects** provide the capabilities of organising and manipulating components into compound objects that can be exploded into its separate components for detailed information. The geometrical representations may convey information such as dimensions and location of products in a project. However the construction of the project products requires more information than that can be conveyed through AutoCAD drawing objects. **Attribute objects** are textual definitions and characterises that can be attached to drawing or block objects for more and detailed information of the drawing objects.

Spreadsheet files: AutoCAD allows designers to extract attribute information from drawing objects in CDF (Comma Delimited Format), SDF (Space Delimited Format) or DXF (Data Exchange Format) for use in other programs such as databases to produce information such as parts list, bill of materials (BOM), or bill of quantities (BoQ). The CDF, SDF and DXF file formats can be imported into program that accepts ASCII files, these include, Microsoft Excel and Microsoft Access files.

4 Process Specification Requirements related to Cost Estimating

The intent of cost estimating is to determine the monetary value of projects based on various resources needed. Construction Computer Software (CCS) is one of the cost estimating applications that can be used in the construction industry today. CCS is a Project Management System specifically designed for construction [4]. CCS was started in 1978 as a division of a large South African construction company. Today it is an independent organization and most of the software team has been with CCS since its inception. Over 250 contractors in more than 20 countries use the CCS System. The system is continually being extended to accommodate changing conditions in the industry and new user requirements.

The CCS estimating application uses ASCII text files to specify the tasks and resources, details of the project and materials, equipment and labour and other related information. In addition to the native file format CCS can read from and write to the following external file formats: - CCS can writes to MPX files that may be opened in Microsoft Project or Primavera P3, writes to and reads from Microsoft Excel files, writes to multiple Excel spreadsheets files for use with other applications, reads from a Microsoft Project export file (MPX file), reads from a Primavera export file, reads from an ASCII (text) file, reads from Lotus 1-2-3 through Microsoft Excel, reads from Microsoft word and Save current selection as Bitmap file.

CCS uses bills of quantities, macros, operations and worksheet resources as the main cost estimating entities and a large number of data fields associated with these entities to describe construction processes as input files and uses different output files for cost data. Construction Computer Software (CCS) uses the following data types and structures:

- Bill of Quantity items has the following data fields associated with each macro operation
  - Item Number, Op Code (Operation Code), Op Code Description,
  - Op Code Unit, Billed Quantity, Nett Rate, Nett Amount
- Macros have the following data fields associated with each macro operation
  - CCS Page Number, Item Number, Op Code (Operation Code), Op Code Description,
  - Op Code Unit, Billed Quantity, Nett Rate, Nett Amount
  - Trade code
  - Operations
  - Nett Split Rates
  - Op Code Attributes (Attr)
  - Page / Item: page number and item number combined
  - Gross Rate, Billed Gross Amount
  - Selling Rate, Billed Selling Amount
- Operations: have the following data fields associated with each operation
  - Item Number, Op Code (Operation Code), Op Code Description, Op Code Unit, Quantity, Nett Rate, Nett Amount
  - Work sheet resources and cost rates
  - Nett Split Rates, Op Code Attributes (Attr)
  - Page / Item, Gross Rate, Billed Gross Amount
  - Selling Rate, Billed Selling Amount

- Resources: have the following data fields associated with each resource
  - Resource Type, Resource Code, Resource Description, Final Rate, Resource Unit, Resource Attributes (Attr), Base Rate
  - Resource group codes
  - Resource class codes, Production Factor, Production Code
  - Usage, Usage Value
  - Resource amount

The output cost data is organized into the following files, which contain the following basic entities each with a large number of attributes:

- Spool file Reports for Resources by bill items and Billed quantity (Nett and gross amount), Worksheets and Tax analysis
- Trade summary sheet for cost summary per trade
- Cash flow data file for Bill items/OP code (Bill of quantity based); YAC job Reports (Activities based); Cash spread over time (duration based); Work code (bill items group based); and Resources cost Report
- CSV file Reports for Selling Cost amount per trade/task/work/bill items; Nett split Cost amount per resource; and Gross split cost amount for trade/task/work/bill items

5 Process Specification Requirements related to Scheduling and Planning

Scheduling is the planning and organizing often of work, which involves the estimate of time and durations and the sequencing of operations, required to construct a facility/building based on resource usage and relationships between the operations. There are number of computer applications that can be used for construction planning and scheduling such as the Microsoft Project (Microsoft INC) [5] and Primavera Project Planner (Primavera 1991) [6] among the others. Microsoft Project is considered in this paper due to its wide use in the construction industry.

Microsoft Project is a scheduling tool developed by Microsoft inc., currently widely used in the construction project management domain. Microsoft project scheduler is a C++ library of classes and functions that implement concepts for constraint based scheduling. The library enables the representation of scheduling elements and constraints such as activity duration, precedence constraints, resource assignments and resource sharing. The scheduler uses MPP, MPX and templates to specify and represent project information. In addition to the native file formats Microsoft Project 98 can read from and write to the following external file formats:

- MPD/MDB files (these are both Microsoft Access 8.0 format).
- Microsoft Excel 97 XLS files (read but not write).
- Microsoft Excel 5.0/95 XLS files.
- Microsoft Excel 5.0/95 PivotTables (write but no read).
- TXT files.
- CSV files.
- HTML files (write but not read).

Microsoft Project uses the *activity* or *task* (essentially a process entity), *project, resources, resource utilization* (the allocation of specific resources to specific activities), *precedence logic* (the inter-activity sequencing constraints), and *work calendars* (descriptions of the length and timing of the work week available for different types of processes) as the main entities in the process of producing time and resources related process information. A large number of attributes are associated with these basic entities. Microsoft Project uses the following data structures to represent the construction process plan information.

- Tasks have the following data fields associated with them:
  - durations, start and finish dates (early and late), float or slack time, and task delays,
  - work hours, task progress, and earned value analysis,
  - predecessor, successor, and scheduling constraints,

- task costs,
- hierarchical breakdowns (outlines), summary and rollup schedules, subprojects, and work breakdown structures,
- resources,
- task ID's, names, user-defined text and flag fields, notes, contact names, creation dates, and OLE linked objects,
- milestone, criticality, priority, confirmation, and update indicators,
- multiple versions of many of the above data items are stored, including *normal, actual, baseline, remaining,* and numerous user-defined versions.

• Resources have the following data fields associated with them:
  - costs, work hours, work calendars, variances, accrual basis, overtime,
  - rates,
  - resource ID's, names, groups, units, notes and OLE linked objects,
  - progress, peak use, and over-allocation indicators,
  - multiple versions of many of the above data items are stored, including normal, actual, baseline, remaining, and overtime.

6 Electronic Information Exchange between Construction Computer Applications

With the wide spread use of computer applications in the AEC/FM environment to date, information have become increasingly available in electronic format for use in construction operations [7]. As a result there has been an increasing need for an electronic form of information exchange between construction applications in the sector.

The whole set of information, required for a construction project, is produced by the various professional participants who usually come from different organizations and use their individual technology applications. This paper has considered the representation of construction process related information in the pre-construction stages using

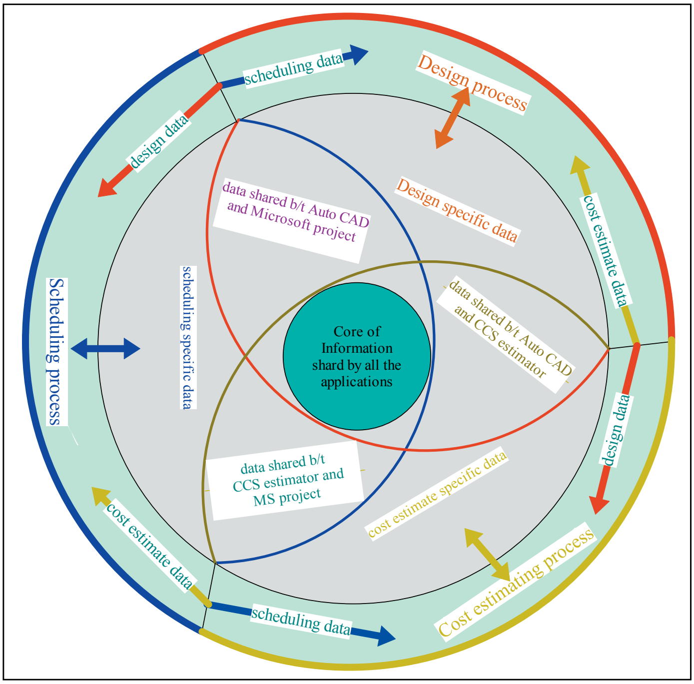

Figure 1: outline of the fragmented and shared process information within the pre-construction stages

Table 1: process specification data types adopted by each of these applications to represent the shared concepts between AutoCAD, CCS and Microsoft project applications

Shared process concepts	AutoCAD representations	CCS representations	Microsoft project representations
Complex Construction product (e.g. door)	Compound Drawing object or block	Bill of quantity Macro operation	Summary task
Simple construction product (e.g. door frame)	Component drawing object	Macro operation	Summary task
Construction activity	Method of construction	Operation	Simple task
Construction product material (e.g. wood)	Material of construction	Material	Resource

AutoCAD, CCS and Microsoft project software applications The different professionals produce their respective information using their individual computer applications and contribute to the whole set of information. For example the information of figure 1 is generated at the architectural design, cost estimating and scheduling stages using AutoCAD, CCS and Microsoft project scheduling packages and their respective information representation terminologies and data structures or file formats described in the previous sections.

As shown in figure 1, this information is fragmented and yet may have to be shared between the different pre-construction stages and software applications. The term "fragmented information" refer to the process information generated and represented at the various stages using different software applications with varying terminologies and data structures. We use the term "shared information" to refer to the process concepts in different representations or terminologies shared between construction stages. Tables 1 and 2 show the concepts shared between these applications and their respective representations and data types within each of the applications in fig 1. Tables1 & 2 also show the process specification data types representing shared process concepts between AutoCAD, CCS and Microsoft project applications; and CCS and Microsoft project applications respectively. The data types representing process concepts shared between AutoCAD and Microsoft project; and between AutoCAD and CCS applications are as shown in table 1.

There are a number of approaches for direct electronic data exchange between software applications. Transferring physical data files, accessing a central database through standard access methods, and procedural calls through an application-programming interface are some of the alternative mechanisms for exchanging information [8]. These alternatives hold great promise for improved communications, information exchange and interoperability of computer applications. However these mechanisms may not support accurate information exchange due to the differences in the meaning/semantics of process information representation terminologies used in the different applications. Simply sharing terminology is insufficient to support interoperability and may lead to inconsistent interpretations and uses of information. Obstacles to interoperability arise from the fact that the applications that support the stages do not share the semantics of their information representations or terminologies.

The ability to automate exchange of process information between computer applications is important for computer-based collaboration in construction. A primary requirement for such collaboration is the development of information representation standards that are needed to provide a unifying neutral language for specifying the information representations used within different computer applications. Hence there is a need for the standardization of process information representation structures for the construction industry to support automation of process information exchange

Table 2: data types represent process concepts shared between CCS and Microsoft project applications

CCS Cost estimating Concepts	Process Concepts	MS Project Scheduling Concepts
Bill of Quantity item	Complex construction product or process (high level)	Summary Task
Macro	Complex construction product or process (lower level)	**Summary Task**
Operation - Material	Simple construction process Material consumed in the process	Simple **Task** - Resource
Billed Quantities Amount - Op code description - AMOUNT - LABOUR cost - PLANT cost - MATERIAL cost - TEMP.MATERIAL cost - SUBCONSTRACT cost	 Construction process at any level Total construction cost Total labour use cost Total plant use cost Total material consumption cost Total temp material use cost Total subcontract work cost	**Gantt chart cost** - Task - Total task cost - Resource cost/use - Resource cost/use - Resource cost/us - Resource cost/use - Resource cost/use
Bill Resource Analysis - RESOURCE NAMES - Rate/per utilization - UTILIZATION - AMOUNT=rate*utilization - LABOUR - PLANT - MATERIAL - TEMP.MATERIAL - SUBCONSTRACT	 Name of resource used or consumed Rate per duration or quantity .... Total cost of resource for a process Man power used in the process Machinery used in the process Material consumed in the process e.g. form work used in the process Organization performing processes	**Resource sheet entry** - Resource name - Standardrate/perduration - … - Cost/use - Resources - Resources - Resources - Resources - Resources
Resource by Bill Item - OP CODE DESCRIPTION - QTY - RATE - RESOURCE NAME - QUANTITY - RATE - AMOUNT	 Construction process an any level Quantity of work in the process Cost per quantity of work in process Name of resource used or consumed Quantity or duration of resource Cost rate pre duration or quantity Total cost of resource	**Gantt chart entry** - Task … … - Resource names - …… - Resource rate - Cost/use

between diverse software applications. That is there is a need for a standard language that can interpret the meaning of set of messages exchanged.

As a result there have been several efforts in information standardization in the field of industrial automation and integration in general [9]. Most of the standardization efforts in the AEC/FM sector have focused on facilitating product data and product related information exchange between software applications based on product data technology. However among these, the IAI's IFCs effort target applications include CAD systems and project management applications: cost estimating and scheduling and planning systems [10]. The following section reviews literature on the Project Management (PM) Domain Group of the IAI North American Chapter (IAI NA PM), and the results of the implementation of the project management portions of the IFCs standards.

7 The IAI's IFC Standard Development: the Project Management Domain Effort

The International Alliance for Interoperability: IAI's IFCs, the largest current effort within the building construction industry, is an international industry-led initiative aimed at defining a specification called the Industry Foundation Classes (IFCs) for AEC/FM to enable interoperability between industry processes of all different professional domains by allowing the computer applications used by all project participants to share and exchange project information [10]. The IFCs are data classes defined in a neutral computer language that support the representation and exchange of building project information between different types of AEC/FM computer applications. Development of IFC is guided by versions and releases, which extend the scope successively. The processes supported by the current IFC2x specifications are outline conceptual design, full conceptual design, co-ordinated design, procurement and full financial authority, production information, construction, operation and maintenance.

In addition to physical information about buildings, the Industry Foundation Classes (IFCs) tend to represent project management information such as cost estimating and scheduling data. The IFCs have included some project management-related objects since their initial Release, in addition to the core concepts relating to the project management portions of the IFC, a significant number of new and revised objects have been included in the IFC Release 2.0 [11]. However these have received no implementation until the Project Management (PM) Domain Group of the IAI North American Chapter (IAI NA PM) held a workshop to conducted trials of the project management portions of the IFCs in February 1999[12], to test how well the IFCs represent the cost estimating and scheduling information and evaluate the current IFC models as they relate to the project management tasks of estimating and scheduling. Form this exercise it was identified that changes of class names between versions (from IFC Release 1.5.1 to IFC Release 2.0 Beta) were inevitable resulting confusion and required redundant work during the implementation of the PM test case. The results identified many areas for potential improvement, but also confirmed that the overall approach taken by the models worked well for representing and integrating product, work process, estimating, and scheduling information.

The application of the Process Specification Language (PSL) in construction is explored as an alternative to the IFC standard for AEC/FM software applications interoperability.

8 Overview of the Process Specification Language (PSL)

To address the issue of software interoperability problems in the manufacturing industry the Manufacturing Systems Integration Division (MSID) at National Institute of Standards and Technology (NIST) developed PSL, a neutral, standard language for process specification to serve as an interchange language for process information exchange between multiple process-related applications throughout the manufacturing life cycle [13]. For many years MSID has been involved in the definition of a neutral representation of product data that was realized recently through the STEP standard [14] As the notion of process also underlies the entire manufacturing cycle, the representation of manufacturing process became another candidate area of focus. PSL is being standardized within Joint Working Group 8 of Sub-committee 4 (Industrial data) and Sub-committee 5 (Manufacturing integration) of Technical committee ISO TC 184 (Industrial automation systems and integration) as ISO 18629. More information about the evolving PSL Standard can be found on [14].

The underlying foundation of PSL is ontology, a formal semantic definition of the process-related terms in the language. Therefore PSL is ontology or a data model, which incorporates a syntax and grammar specification to make it a language [14]. In providing ontology the notions of Language, Model theory, and Proof theory are specified for PSL [2]:

Language: A language is a lexicon (a set of symbols) and a grammar that is a specification of how these symbols can be combined to make well-formed formulas or formal sentences. The underlying grammar used for PSL is that of KIF (Knowledge Interchange Format) [8] which 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.

Model theory: The model theory of PSL provides a rigorous, abstract mathematical characterization of the semantics, or meaning, of the language hence defines meanings for the terminology and a notion of truth for sentences of the language.

Proof theory: Finally the proof theory of PSL consists of three components: PSL Core, the foundational theories (outer Core), and PSL extensions:

PSL core: PSL core is a set of axioms written in the basic language of PSL. The purpose of PSL-Core is to axiomatize a set of intuitive semantic primitives that is adequate for describing the fundamental concepts of manufacturing processes. These axioms provide a syntactic representation of the PSL model theory and semantics for the terms in the language.

Foundational theories: these theories provide an expressive power sufficient to give precise definitions or axiomatization for the primitive concepts of PSL.

Extensions: the third component of PSL ontology provides expressive resources that are adequate for expressing more complex process involving concepts that exceed those of the PSL core. The extensions are groupings of related process concepts that are common to some but not all applications that together provide an added functionality. The terminology and their definitions in the PSL extensions are designed to capture the intuitive meaning of the manufacturing process concepts that are common to some applications.

To date PSL has developed set of core theories for expressing a wide range of basic processes and a set of extensions to provide resource for expressing more complex processes relating to planning and scheduling that are grouped into the PSL parts. For detailed information of PSL parts and their process concepts refer to PSL home page [14].

9 PSL from Construction Processes Perspective

PSL was mainly developed and used in the manufacturing sector for the specification and representation of manufacturing processes information to integrate multiple process related manufacturing applications. The PSL process concepts were analyzed from the perspective of the process specification requirements intrinsic to the pre-construction stages that are related to the AutoCAD, CCS and Microsoft project applications.

To date PSL has developed core set of theories that can express wide range of basic processes and a set of extensions to provide resources for expressing more complex processes [2]. The core theories can define the process specification data types that are adopted to represent basic process concepts shared between AutoCAD design, CCS cost estimating and Microsoft project scheduling applications. While the planning and scheduling concepts of project management are well defined within the PSL extensions, there are no extensions applicable for either the architectural design or cost estimating concepts of construction works. Based on the analysis of the semantic concepts of AutoCAD, CCS, and Microsoft project applications' process specification data types and the manufacturing process concepts' definitions within the PSL ontology, it was identified that the "Processor Activities" of part 46, *"Sub-Activity Theory"* of part 12, "Theory of *Resource Requirements*" of part 14, and *"Reasoning* *about Resource Divisibility"* of part 44 of PSL ontology contain some definitions applicable for some of the data types that represent shared concepts between Microsoft project and the other two applications. Detailed information of the process concepts of the PSL parts can be found in [14].

PSL was implemented in a construction scenario within the pre-construction stages for an in depth investigation of the applicability and impact of the language in construction. The implementation is aimed at exploring a construction scenario in which PSL can be used to facilitate the translation of process information representations between AutoCAD, CCS and Microsoft project packages. In this implementation a hand written translation process was conducted that involves process related architectural design; cost estimating; and scheduling and planning information for construction of a case study project; the installation of a metal door in a metal opening of a building project where PSL is used to as an interchange language between the three software applications. The translation processes provide a procedure for the identification of the concepts requiring incorporation into the PSL ontology new extensions that are necessary for the applications' complete interoperability. The following sections describe and present an example of the translation process.

10 Translation Process: Information Exchange Between the Scenario Applications using PSL

In this process as an interchange language PSL is used to facilitate the translation of process information representations between AutoCAD, CCS and Microsoft Project. As described earlier each of the applications has its own internal representations and use different terminologies. Integration of the applications requires a neutral definition of the terms used by the different applications and translation of the representations using PSL. The information translation between the applications' files requires a two-stage process: syntactic and semantic translations. The syntactic translator is a parser between the PSL syntax (KIF notation) and the native syntax of each of the applications; this parser keeps the terminology of the application intact. The semantic translation substitutes terminology of each of the applications with the definitions written using PSL terminology. The information exchange between the three applications is carried out through translation of information representations between each of the software application's native ontology and PSL ontology using parsing and pattern matching (concepts mapping) procedures for syntactic and semantic translations respectively. Hence in order to facilitate the mapping of information between the applications and PSL, the first step in the translation process required the development of each of the applications native ontology. The ontology of each of the applications is hidden and is known by the developers only. However, ontology of such applications can be developed from the user point of view that is, how an application represents information using the terminologies and file formats, defined by the developers is the ontology of the application.

11 Syntactic and Semantic Translation

Information exchange using PSL requires syntactic and semantic translators to translate information from the applications to PSL and back to the applications. While the syntactic translators are parsing procedures for syntactic translation between the applications information representations and PSL syntax that is Knowledge Interchange Format (KIF) [15], the semantic translators are semantic translation definitions. These translation definitions between an application ontology and PSL are driven by the ontological definitions that were written using the same foundational theories. These are definitions for the terminology of the application ontology, using only the terminology from the PSL Ontology, as well as definitions for the terminology of the PSL Ontology using only the terminology of the application ontology. Based on PSL ontology and the applications' native ontology: the application's terminologies (data types) and file formats, and representations of process data for the case study project, hand written semantic translators (semantic translation definitions) were developed between PSL ontology and each of the applications. As an example some of semantic translation definitions for the three software applications are illustrated in the following section.

12 Semantic Translation Definitions for the AutoCAD, CCS and Microsoft project Applications:

Translation Definition for AutoCAD Application (A)

```
(<=> (design-object component@A ?r)
     (resource-created@PSL  ?r))
(<=> (material_of_construction@A ?r)
     (consumable@PSL  ?r))
(<=> (construction@A ?a)
     (processor_activity@PSL  ?a))
```

Translation Definition for CCS Application (C)

```
(<=> (macro_op@C  ?r)
     (resource-created@PSL  ?r))
(<=> (operation@C ?a)
     (processor_activity@PSL  ?a))
(<=> (material@C ?r)
     (consumable@PSL  ?r))
```

Translation Definition for Microsoft Project Application (S)

```
(<=> (summary_task@S  ?r)
     (resource-created@PSL  ?r))
(<=> (task@S ?a)
     (processor_activity@PSL  ?a))
(<=> (resource@S ?r)
     (consumable@PSL  ?r))
```

13 Translation Process Example:

In the translation process, construction process information for a case study project was exchanged between AutoCAD, CCS and Microsoft project, using hand written parsing procedure, and the semantic translation definitions developed between PSL and each of the applications. The procedure begins with a design file written in the AutoCAD application's syntax using its terminology (AutoCAD application's native ontology). Figure 2 shows the mapping of process related AutoCAD design, CCS cost estimating and Microsoft scheduling concepts to PSL ontology, and exchanges of process information between the three applications using the PSL ontology definitions as an interchange language. Figure 2 particularly, depicts the translation of process information representations between the three software applications using parsing procedure for syntactic translation of the applications' representations to and from the PSL syntax (KIF notation); and the semantic translation definitions developed between the applications and PSL as a common language for semantic translation of process information representation terminologies between the applications. As an example, the following section presents, part of the case study project construction process information exchanged between the three scenario applications to illustrate the translation and information exchange procedure, and the role of the PSL language. Detailed information about the translation procedure can be found in PhD Thesis [16].

14 Translation Process Example: information exchange between AutoCAD and CCS Architectural Design Data File in AutoCAD Application's Representations

Architectural Design data (textual file) Written in AutoCAD's textual attribute representation format and terminologies.

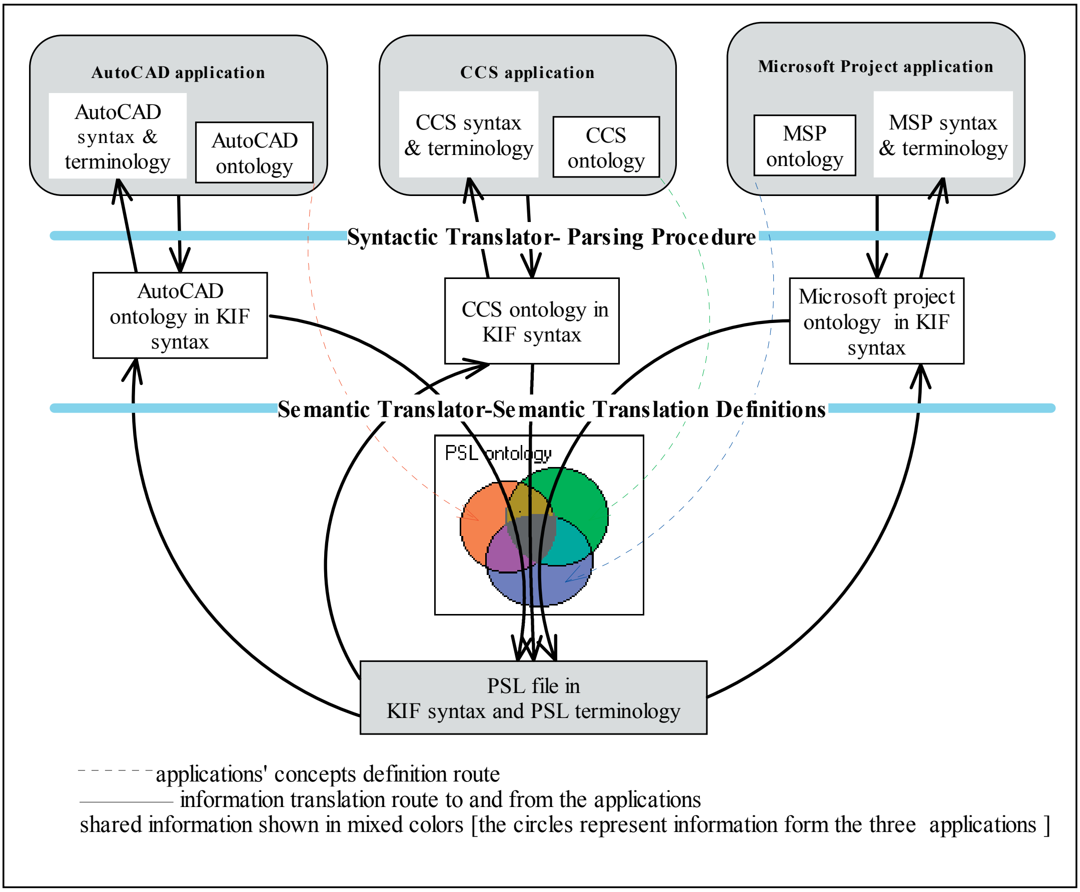

Figure 2: Syntactic and Semantic translation of process information between the three Applications and PSL representations.

```
{DWG-object     metal bldg opening (x)}
{components     wood frame; steel girt;  wall panel }

{Component:     97"*102 ½" wood frame
                metal bldg opening (x)}
{Material:      2*12 Douglas Fire Wood (x)}
{construction:  wood-frame construction as design (x)}
```

15 Architectural Design Data File in PSL syntax (KIF Notation)

The syntactic translator reads the architectural design data file in AutoCAD application's syntax and terminology and produces a corresponding file using PSL syntax (KIF notation), but still persevering the AutoCAD application's terminology

```
(forall ?x
(=>(metal_bldg-opening  ?x)
     (DWG_object  ?x)))
(<=>(=>(metal_bldg-opening ?rx)
               (DWG_object ?x))
     (exists (?r1  ?r2 ?r3)
               (and (=> (wood_frame ?r1)
                    (component   ?r1 ?x)))))

(forall ?a
          (=>(wood-frame_constructed_as_design  ?a)
              (construction   ?a  ?r))
     (forall ?r1
                    (=>(2*12 Douglas Fire Wood  ?r1)
                 (material_of_construction   ?r1 ?a)))))
```

16 Architectural Design Data File in PSL Representations

In this translation, using the translation definitions for AutoCAD the architectural design data file in PSL syntax (KIF) is mapped to PSL definitions that contain only PSL terminology, and produces a corresponding file containing only PSL terminology in PSL syntax (KIF) by substituting the definition of all the AutoCAD application's terminologies with definitions in PSL.

```
(forall ?x
(=>(metal_bldg-opening  ?x)
     (DWG_object  ?x))) --------New product concepts
(<=>(=>(metal_bldg-opening ?rx)
               (DWG_object ?------New product concepts
    (exists (?r1  ?r2 ?r3)
             (and (=> (wood_frame ?r1)
                 (resource_created   ?r1 ?x))

(forall ?a
        (=>(wood-frame_constructed_as_design  ?a)
           (processor_activity  ?a  ?r))
    (forall ?r1
               (=>(2*12 Douglas Fire Wood  ?r1)
             (material_consumable ?r1 ?a)))))
```

17 Architectural Design Data File in PSL syntax and CCS terminology

In this translation a reverse step is followed. Using the semantic translator (semantic translation definitions) for CCS the AutoCAD application's architectural design data file containing only PSL terminology in PSL (KIF notation) syntax is translated into CCS application's terminology but still preserving the PSL (KIF notation) syntax intact.

```
(forall ?x
(=>(metal_bldg-opening  ?x)
      (bill_of_quantity_item  ?x))) ---New product
                                                concepts
(<=>(=>(metal_bldg-opening ?rx)
       (bill_of_quantity_item ?x)) ---New product
                                              concepts
    (exists (?r1  ?r2 ?r3)
             (and (=> (wood_frame ?r1)
                 (macro_op   ?r1 ?x))

(forall ?a
        (=>(wood-frame_constructed_as_design  ?a)
           (operation   ?a  ?r))
 (forall ?r1
             (=>(2*12 Douglas Fire Wood  ?r1)
    (material  ?r1 ?a)))))----new  resource category
                                 concept
(forall ?x ?a ?r1
             (=>(and (bill_of_quantity_item  ?x )
           (material  ?r1)
                      (operation ?a ))
           (exists ?N
                  (net_amount  ?C  ?x ?r1 ?a)))))
                                          -----new cost concept
```

18 Architectural Design Data File in CCS Application's input File

In this translation the syntactic translator maps the architectural design data file containing only CCS application's terminology in PSL (KIF notation) syntax back into CCS application's syntax and terminology. This contains only the architectural design data as input for cost estimating application.

(bill of quantity item	metal building-opening (x))
(Macro Op	97"*102 ½" wood frame (x))
(Operation	wood-frame construction as design(x))
(material	2*12 Douglas Fire Wood (x))
(Billed Quantity	(x))
(Nett Rate	(x))
(Nett Amount	(x))

19 Cost Estimating Output File in CCS Application's Representations

This represents the output file of CCS cost estimating application that is generated based on the input file translated from AutoCAD data representation to CCS using PSL.

```
[BILLED QUNATITIES NETT AMOUNT]
(Item Number              )
(Op Code                  )
(Op Code Description construct wood frame)
(Op Code Unit             )
(Billed Quantity          )
(AMOUNT           1000)
          --------task total cost in MSP
(LABOUR           500)---&
(PLANT            250)----&
(MATERIAL         250)----- are cost/
            use of resources for a task
[BILL RESOURCE ANALYSIS for all trades]
(RESOURCE CODE    )
(RESOURCE NAMES   2*12 Douglas Fire
                  Wood)--simple resource
(UNIT             )
(GROUP            )
(RATE             x)
(UTILIZATION      y)
(AMOUNT           250)-------=x*y &
              amount=material use/cost
(LABOUR           )
(PLANT            )
(MATERIAL         250)
[RESOURCE by BILL ITEM]
 (ITEM            )
 (OP CODE         )
 (OP CODE DESCRIPTION
                  construct wood frame)
 ( QTY            )
 ( UNIT           )
 ( RATE =         )
RESOURCES INFORMATION
  (RESOURCE GROUP )
(RESOURCE CODE   )
(RESOURCE NAME   2*12 Douglas Fire
                 Wood)
(QUANTITY        )
(RATE            )
(AMOUNT          250) ----=QTY*RATE or
                 cost/use for task
 (PRODUCTION     )
```

20 Proposal for New Extensions to the PSL Ontology

Based on the translation process conducted to exchange information between AutoCAD, CCS and Microsoft project applications using PSL the need for the new extension to incorporate construction process concepts related to these applications were identified. These concepts do not include every aspect of the applications but they were identified from the shared concepts between the three applications. For detailed information about the translation procedure and how the need for the new extensions, and their process concepts were arrived refer to PhD Thesis [9]. The proposed new extensions including the concepts, the relationships between them, and the theories required from other existing PSL extensions are as follows:

(a) Construction or Project Product

Extension Name: Construction or Project_Product.def
Primitive Lexicon: None
Defined Lexicon:

- (design_elements ?x)
- (components ?x1… ?x)

Theories Required by this Extension: Processor_Activity.def, psl_core.th

(b) Cost

Extension Name: Construction or cost.def
Primitive Lexicon: None
Defined Lexicon:

- (resource_rate ?x)
- (activity_rate ?x)

- (product_rate ?x)
- (activiy_total_cost ?x)
- (activiy_fixed_cost ?x)
- (product_cost ?x)
- (resource_cost/use ?x ?a)
- (resource_standad_rate ?x ?a)
- (resource_category_rate ?x)
- (resource_category_cost ?x)
- (overhead_cost ?x)
- (mark_up ?x)
- (split_rate ?x)
- (plug_rate ?x)

Theories Required by this Extension: Processor_Activity.def, psl_core.th, requires.th, Project_Product.def

(c) Activity and Product Quantity

Extension Name: Activity_&/or_Product _Quantity.def

Primitive Lexicon: None

Defined Lexicon:

- (activity_billed_quantity ?x)
- (product_quantity ?x)

Theories Required by this Extension: Processor_Activity.def, sub_activiy.th, psl_core.th, Project_Product.def,

(d) Resource Category

Extension Name: Resource_Category.def

Primitive Lexicon: None

Defined Lexicon:

- (plant ?x)
- (labour ?x)
- (material ?x)
- (temporary_material ?x)
- (sub_contract ?x)

Theories Required by this Extension: requires.th , psl_core.th, res_divisible.def

21 Conclusion

This paper has reported on a research project aimed at exploring the applicability of a standard process specification language (PSL) for the automation of process information exchange between semantically disparate AEC/FM software applications. PSL was initially developed for the interoperability of process-related manufacturing applications. The trial implementation of PSL in the construction scenario has proven the potential of the language to resolve the semantic differences that hampered electronic information exchange between construction applications. The ontology based approach to PLS compliance is different from the traditional approach to standards compliance where rather than forcing the adoption of exactly the same terminology, an application is PSL-compliant if there exist definitions for its terminology (semantic definitions for the application's process specification data types) in the PSL ontology using PSL language and syntax. Once these applications become PSL compliant by defining the semantic concepts of their process information representations (or data types), they would be able exchange information with each other and with every other application that is PSL complaint.

In addition to the built and design information about buildings the IAI's IFCs efforts include project management information for estimating and scheduling applications interoperability. However from the results of the trial implementation of the IFCs project management parts it was identified that changes of class names between versions (from IFC Release 1.5.1 to IFC Release 2.0 Beta) were inevitable resulting confusion and required redundant work, and the need for potential improvement. Contrary to the IAI's long-term efforts PSL provides a mechanism for rapid technology acquisition and deployment for AEC/FM systems integration.

PSL was designed for process information. However from the implementation of PSL in the construction scenario which involved information exchange between CCS cost estimating, and Microsoft project scheduling including AutoCAD design, it has shown that PSL is applicable for process related product data as well. The translation process has proven that PSL can be used for process related product data exchange between the design and project management applications provided extensions are developed to incorporate the concepts of construction products. Product is related to process definitions through the processes that are carried out to develop the product. However PSL ontology to date has not defined the concepts of construction products, hence a new extension is required to incorporate the semantic concepts of product/construction product/ project product/ or project design elements. Cost information of construction processes is the main issue in the pre-construction stages and cost related information is exchanged between cost estimating and scheduling applications. Additionally a designer may be required to produce design information for a project to a specified budget hence the designer may need to exchange cost information with the other two applications. There are no semantic concepts for cost in general in the current PSL ontology. Hence there is a need for further extensions to the PSL ontology to incorporate the concepts of cost. Based on this, new sets of extensions to the PSL ontology were proposed to incorporate the construction process concepts identified by the research.

A hand written translation was conducted in the implementation of PSL for information exchange between the applications using parsing procedures for syntax translation, and mapping and pattern matching of concepts based on the semantic translation definitions developed between PSL and each of the applications. The Development of electronic translation programs would provide better basis for evaluating the potential use of PSL in the automation of the process information exchange between construction computer applications. However the hand written translation process shows the underlying details of the information exchange that can be hidden below the program.

Although the research study has focused on integration of AEC/FM processes and software interoperability in the construction sector, the benefit that can be obtained from PSL would not be restricted to the industry. For example PSL would also integrate construction with the manufacturing and other PSL compliant sectors, which contribute to the success of construction practices. This would allow the construction professionals to exchange electronic information with manufacturers and suppliers efficiently.

REFERENCES

[1] Cutting-Decelle A.F., Michel J,J., Schlenoff C. 2000. *From Manufacturing to construction: to wards a common representation of process information:* the PSL approach, concurrent Engineering Conference., Lyon.

[2] Schlenoff, C., Gruninger M., Tissot, F., Valois, J., Lubell, J., Lee, J., *"The Process Specification Language (PSL): Overview and Version 1.0 Specification,"* NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD., 2000.

[3] Omura, G., Mastering AutoCAD 14, 2002, sybex, San Francisco.

[4] WWW.ccsuk.com

[5] Microsoft (1994), *Microsoft Project, Version 4.0,* Reading, WA.

[6] Primavera Systems, Inc. (1991). *Primavera Project Planner 5.0, Reference,* Bala Cynwyd, PA.

[7] Jagbeck A. 1998., *IT Support for Construction Planning. A system design based on integrated information* (Thesis). Royal Institute of Technology, Construction management and economics, Stockholm, ISBN 91-7170-269-5.

[8] T. Froese and B. Paulson, "OPIS, An Object-Model-Based Project information System," Microcomputers in Civil Engineering, Special Issue on Innovative Computer Technology in Construction, Vol. 9, February 1994, pp. 13-28.

[9] ISO (1994) . "Industry Automation Systems-Product Information Representation and Exchange - Part 1, Overview and Fundamental Principles", ISO TC184/SC4 Project Management Advisory Group.

[10] IAI (1996), End User Guide to Industry Foundation Classes, Enabling Interoperability in the AEC/FM Industry. International Alliance for Interoperability (IAI), Washington, DC.

[11] IAI, IFC release 2x, IFC technical Guide (TG), 2000

[12] Froese, T., et al. (1999) “Industry Foundation Classes For Project Management—A Trial Implementation”, Electronic J. of Information Technology in Construction, Vol.4, 1999, pp.17-36. Available online at http://itcon.org/1999/2/

[13] Craig Schlenoff, The Process Specification Language (PSL) Overview and Version 1.0 Specification

[14] http://www.mel.nist.gov/psl/.

[15] Genesereth, M., Fikes, R.: Knowledge Interchange Format (KIF) Version 3.0 - Reference Manual, 1992, Computer Science Dept., Stanford University, Stanford, CA.

[16] Tesfagaber, G., “Application of PSL to Construction Process Information Specification and Exchange “, University of Loughborough, PhD report, 2004