Towards the understanding of the requirements of a communication language to support process interoperation in cross-disciplinary supply chains

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Citation: DAS, B. ... et al., 2007. Towards the understanding of the requirements of a communication language to support process interoperation in cross-disciplinary supply chains. International Journal of Computer-Integrated Manufacturing, 20 (4), pp.396-410.

Additional Information:

- This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Computer-Integrated Manufacturing on 11/04/2007, available online: http://dx.doi.org/10.1080/09511920600873741.

Metadata Record: https://dspace.lboro.ac.uk/2134/26587

Version: Accepted for publication

Publisher: © Taylor & Francis

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

Please cite the published version.
Towards the understanding of the requirements of a communication language to support process interoperation in cross-disciplinary supply chains

B. DAS†, A. F. CUTTING-DECELLE‡, R. I. M. YOUNG*†, K. CASE†, S. RAHIMIFARD†, C. J. ANUMBA† and N. BOUCHLAGHEM†

†Loughborough University, Loughborough, LE11 3TU, UK
‡University of Evry/IUT-Dept Organisation and Production Management EVRY, France

Many manufacturing organizations while doing business either directly or indirectly with other industrial sectors often encounter interoperability problems among software systems. This increases the business cost and reduces the efficiency. Research communities are exploring ways to reduce this cost. Incompatibility amongst the syntaxes and the semantics of the languages of application systems is the most common cause to this problem. The process specification language (PSL), an ISO standard (18629), has the potential to overcome some of these difficulties by acting as a neutral communication language. The current paper has therefore focused on exploring this aspect of the PSL within a cross-disciplinary supply chain environment.

The paper explores a specific cross-disciplinary supply chain scenario in order to understand the mechanisms of communications within the system. Interoperability of processes supporting those communications are analysed against PSL. A strategy is proposed for sharing process information amongst the supply chain nodes using the ‘PSL 20 questions wizard and it is concluded that, although there is a need to develop more effective methods for mapping systems to PSL, it can still be seen as a powerful tool to aid the communications between processes in the supply chain. The paper uses a supply chain scenario that cuts across the construction and manufacturing business sectors in order to provide a breadth to the types of disciplines involved in communication.

Keywords: Interoperability; Cross-discipline; Supply chain; Manufacturing; Construction; Communication-language

1. Introduction

The purpose of the current paper is to explore, from a system developer’s viewpoint, the requirements of a communication language that would enable process interoperation in a cross-disciplinary supply chain (SC) system. Many manufacturing organizations do business either directly or indirectly with other industrial sectors where interoperability problems are aggravated by different cultures and disciplines as well as by the broad range of suppliers and subcontractors having different business functions. This has prompted the present authors to undertake this research study within a cross-disciplinary SC environment.

The term ‘interoperability’ refers to the ability to share technical and business data, information and knowledge seamlessly across two or more software tools or application systems in an error-free manner with minimum manual interventions (Ray and Jones 2003). For example, a weekly production scheduling package would require customer order details, production capacity details of machinery at the shop floor and their availability, production process details etc., which would eventually generate weekly production plans as well as product delivery details for
of semantic support for SC communication such as ebXML and RosettaNet (ebXML 2006, RosettaNet 2006). However, a critical issue is the level of rigour involved in the semantic definition. If interoperability is to be checked and confirmed by computer analysis, it is essential that sufficient mathematical rigour underlies the semantic definitions being used. It is for that reason that the current paper has focused on the application of the process specification language (PSL), which is based on first-order logic, to the SC interoperability problem.

PSL (Schlenoff et al. 1996, ISO TC184/SC4 2001) has been developed to provide a common shared language to support process interoperability and is now an ISO standard, ISO 18629. PSL is formal and based on first-order logic, a mathematical set theory (Barwise and Etchemendy 1999). The work reported in this paper has focused on exploring the potential of PSL as a shared communication language within the context of a cross-disciplinary SC. The particular aim of the work is to explore the use of PSL as a formal route to the comparison of potentially interoperable processes in order to identify their level of compatibility.

We have pursued a scenario study approach to developing and understanding the requirements of a cross-disciplinary SC system in terms of likely communication processes. This scenario is introduced in section 2. Within the context of this scenario, the paper goes on to analyse a particular cross disciplinary transaction, the 'purchase order transaction', in order to explore the detailed functions, processes and tasks that may occur in the chain. Section 3, then highlights the communication processes that take place through a set of forms across the chain by following the lifecycle of the purchase order from start to finish.

The analysis of the 'purchase order transaction' scenario, clearly suggests that the processes within the system are the main routes to facilitating the communication across the SC, which occurs through physical movement of sets of forms containing embedded information generated by those processes. It is therefore logical as the next phase of the work to explore characteristics of those processes in order to identify their interoperability potential. The PSL_20 questions the wizard tool developed by the National Institute of Standards and Technology (NIST) (2003), and provides a mechanism for identifying the interoperability characteristics of processes. How this has been applied to our scenario is explained in section 4. Section 5 then provides a general discussion of our work and the conclusions that can be drawn from it.

2. A cross-disciplinary SC scenario

This section introduces a cross-disciplinary SC scenario with a set of processes and information flows, which are
required to support a typical set of business functions. The scenario encompasses a construction company, a construction site, a manufacturer, a retailer and a transporter. The scenario is then used as the basis for the exploration of specific SC processes and the applicability of PSL to process interoperability analysis.

2.1. The SC nodes

Figure 1 shows the SC network as devised for this research study. The main actor of this network is the construction company, which initiates most of the activities.

2.2. Functions of the SC nodes

Various functions (tasks, activities, actions, processes, operations) normally performed by each node of the network within the context of processing a client’s order by the construction company are briefly described below.

1. Construction company. First receives an order from its client for implementing construction project. The project manager in-charge of this project would ask its various departments (see figure 2 for organization structure) to design, cost-estimate and prepare a construction schedule for the project.

Based on the large list of items that need to be ordered for the construction schedule to be completed in time, the project manager would place orders through the materials management department to the manufacturer and the retailer for the goods to be delivered at the construction site on a particular date. Once the orders have been accepted by the supplier and confirmed by the supplier, the project manager would request its accounts department to prepare payment to the supplier for the set of goods he has ordered. However the payment will not be released until the project manager has received confirmation of the delivery of goods at the construction site from the construction site manager and the satisfactory quality report from the quality department. Many other functions take place within the company but they are out of the scope of this scenario.

The main functions involved within the construction company in this business scenario are: order processing, project planning and management, designing of building, cost-estimating, construction scheduling, procuring, quality management and accounting.

Figure 1. A typical cross-disciplinary supply chain network.
2. **Manufacturer.** Based on the order received from the construction company, the manufacturer would process the order. If feasible, the goods would be produced and shipped to customer’s construction site via the transporter. Accounts department prepares payments to suppliers and obtains remittance from the customer. The main functions involved within the manufacturer’s business scenario are: order processing, manufacturing, procuring, goods handling and accounting.

3. **Transporter.** This handles all the tasks relating to transport and delivery of finished goods to customer’s selected location. It receives the shipment order from the manufacturer; collects the goods and delivers them at the construction site; accounts department would obtain remittance from the customer for the service. The main functions involved within the transporter’s business scenario are: order processing, distribution planning and scheduling, goods handling and accounting.

4. **Retailer.** Based on the order received from the customer (manufacturer and construction company) the retailer would analyse the order. If feasible to supply, goods would be delivered. If items are not in stock, the retailer places an order to a distributor and delivers them to the customer when they are received from the distributor. Accounts department prepares payment to the distributor and releases it when goods are received and also obtains remittance from the customer for the goods supplied. The main functions involved within the retailer’s business scenario are: order processing, distribution planning and scheduling, procuring, goods handling and accounting.

5. **Construction site.** This is a part of the construction company’s business until it is handed over to the client. The construction site starts building work according to a project plan and the building work schedule, which would be prepared by the construction company at its head office. The construction site receives notification from the construction company, manufacturer, transporter and the retailer; construction site notifies appropriately the construction company, manufacturer, transporter and the retailer when goods are received; all damaged goods would be sent back to relevant supplier and undamaged goods would be released to the relevant users. The main functions involved at the construction site’s business scenario are: project management and goods handling including quality management.
It is essential to mention here that it is assumed that each actor uses separate workflow engines, as each represents an autonomous company, and there is therefore bound to be a problem with interoperability.

Detailed organization structure and function analysis of all the five nodes have been carried out within the scope of the project/research work. The information presented in this sub-section represents a brief overview.

The functions described here for each node are basically inter-business processes and it is essential to understand how they are related to each other within the node as well as across the SC nodes towards supporting process interoperability in cross-disciplinary SCs. These issues are addressed in the next two sub-sections (2.3 and 2.4).

2.3. Processes to support the SC node functions

This sub-section introduces a set of processes, which are assumed to be needed to support the various functions of the SC node described in sub-section 2.2. For this research work, the term ‘process’ means a structured collection of activities/tasks that have sequential relationships, while an activity/task means the transformation of a set of inputs into a set of outputs.

Table 1 shows the overall list of processes that may occur within the Construction company node to facilitate the functions described earlier. Many other processes occur within the same node, but they are not considered here.

Similarly four other sets of processes supporting the functions of the other nodes have been identified and developed within the scope of the project work. These are: manufacturer, transporter, retailer and construction site.

2.4. Information flow across the SC nodes

An overall information flow diagram across the SC to support an order processing transaction (‘Client’s order’) as initiated by the construction company is shown in figure 3. Table 2 shows the overall list of information as used in figure 3. The numbering procedure used on the information flow line of figure 3 is as follows: first digit indicates the node number, which is followed by a dot and then the next two digits indicate information list number of that particular node.

The diagram shows the nature of interactions of information that may occur across the SC network. Some of this information would be going-out from the node and some will be coming into the node from another external node. Therefore visualization of the flow of information within and across the system boundaries are important particularly from the point of view of software development as well as in understanding the requirements for better communication/interoperability etc. It would also highlight the variations of system requirements.

This brief introduction of the functions, processes and information flow across the SC nodes clearly indicates the nature and the complexity of the system and emphasizes that a clear understanding of the system is essential for managing the SC efficiently (Cutting-Decelle et al. 2004, Young et al. 2004, Cutting-Decelle et al. 2005).

3. Purchase order transaction scenario

The ‘Purchase order transaction scenario’ is now considered as a basis for the detailed exploration of the communication processes across the SC, the understanding of the related information structure, the exploration of process connectivity and also the applicability of the PSL_20 questions wizard. For this work, the term ‘communication process’ means the flow of information from the sender to the receiver and vice versa in order to execute an activity/task. The propagation of such information takes place either electronically or by some other means such as postal systems etc. In this particular scenario as shown in figure 1, the Purchase Department of the construction company of the SC places an order for supplying a set of doors to a manufacturer. Doors should be delivered to its construction site on a particular date. The order may contain the following information: ‘Supply ten pieces of 6.5 ft x 3.5 ft (2 in thick) plane...
varnished ply-wood door complete with lock and handle at our Loughborough construction site on 10 October 2004’. The manufacturer would supply the goods at the construction site through a transporter. The SC schematic as presented in figure 1 is still applicable for this particular scenario, although in this case the retailer is not involved. Similarly the functions and the processes as described in sub-sections 2.2 and 2.3 for each SC node to support the business of those nodes are also valid for this scenario.

3.1. The lifecycle of a purchase order form

3.1.1. How forms are generated. Whenever a business transaction (for example ‘Receive client’s order’ by the construction company) takes place in a SC node, then based on the functions of that node, a number of processes take place within the node. These processes eventually lead to generation of set of data such as: client’s order details, client number, project order number, accounting code number, start date of the project, end date of the project, location of the project etc. Ultimately, depending on the business practice of that node, these data are collated either electronically or manually and a set of information would be produced. Again these sets of information would be collated and eventually a required Form would be generated containing some of this information, for example a *project order form* that would be ready for the next process. The next process may simply be *send completed project order form* to project management department or to a sub-contractor. Similar scenarios would exist within other participating nodes of the SC. However, those participating nodes would not activate/initiate their appropriate functions and processes until they receive from the sender the appropriate information through such forms. Based on this scenario, it may therefore be said that all information flow between SC nodes are expected to be present on such forms. Figure 4, shows the mechanism of formation of a typical form through a set of processes.

3.1.2. Lists of forms relating to the purchase order transaction scenario. Based on this understanding, the set of forms supporting the ‘Purchase order transaction scenario’ have been identified by following the ‘lifecycle’ (start–end) of a purchase order form as shown in figure 5. This figure represents the construction company’s point of view. It shows how the purchase order form progressed through the system across the participating SC nodes, from
the moment it is sent out by the construction company to the relevant node (in this case the manufacturer), all the way until the ordered item relating to this form reaches the construction site through the transporter. The diagram also shows the other associated forms generated by various processes of other nodes needed to react to the information content of the purchase order form. The long solid lines with arrows and numbers on top of them indicate the sequence and direction of movement of these forms from node to node.

As explained earlier (see figure 4), these forms could be linked to the particular processes of the nodes of the SC involved in the transaction. They are also shown in this diagram. For example, in the diagram, the process ‘1–3 Send_order’ relates to the purchase order form sent out by the construction company. Similarly, the process ‘3–5 Notify_customer’ relates to ‘Goods supply date form’, sent out by the manufacturer to the construction company.

Table 3 shows the overall connectivity between the processes and the forms generated by the SC nodes to affect the ‘purchase order transaction scenario’.

### 3.2. Information definitions of the forms

In the previous sub-section, we have just discussed the list of forms that are generated by the ‘purchase order transaction scenario’ and their connectivity with the relevant processes of the node. However to share the information as represented in these forms, we need to know the information content of the forms. For example, ‘items on order’ information may contain data on: quantity, item name, item type, item size, accessories, special requirements, etc. This would enable us to analyse which parts of the information content of the form are shareable and vice versa. The information definitions of these forms are derived from detailed analysis of functions, processes and activities/tasks of the SC nodes relevant to the ‘purchase order transaction scenario’.

Figure 6 shows the possible information of the ‘Purchase Order Form (PO form)’ relating to the ‘purchase order transaction scenario’. It is sent out by the construction company to the manufacturer. The process linked to this form is: ‘1–3: Send purchase order to manufacturer’.

Similarly, information definitions of the other nine forms (see table 3) have been developed through detailed analysis of functions, processes and activities/tasks of the SC nodes relevant to the ‘purchase order transaction scenario’.

### 3.3. Case for exploring interoperability of processes

The analysis of the ‘purchase order transaction scenario’, suggests that the processes within the system are the main sources in facilitating the communication across the SC.
The next phase of the work explored the interoperability characteristics of those processes before attempting to develop mechanisms to share information amongst the nodes of the chain. This had been done through the use of the PSL_20 questions wizard tool as presented in the next section.

4. Application of PSL_20 question wizard to the understanding of process interoperability

This section introduces a short overview of PSL, the PSL_20 questions wizard tool and then explains its application to our scenario.

4.1. Concepts in PSL

PSL is based on a lexicon or a set of logical and non-logical symbols as well as a grammar, i.e. a specification of how these symbols can be combined to make well-formed formulae. The underlying grammar used for PSL is based on the knowledge interchange format – a formal language based on first-order logic developed for the exchange of knowledge among computer programs with disparate representations. This provides a rigorous foundation for the formal definition of the concepts in the language.

PSL provides an extensive set of some 350 concepts for process description. The key classes of concepts can be listed as:


b. Time and state concepts e.g. ‘timepoint’, ‘between’, ‘exists_at’; ‘state’, ‘holds’, ‘prior’, trigger, launch;

c. Ordering concepts e.g. ‘earlier’, ‘initial’, ‘precedes’, ‘successor’;
Abbreviations as used in above figure are:

PO form = Purchase order form; GSD form = Goods supply date form; GCO form = Goods collection order form; GAD form = Goods arrival date form; GCD form = Goods collection date form; GDD form = Goods delivery date form; GHO form = Goods handover form; GR form = Goods receipts form; GRC form = Goods receipt confirmation form; P Con form = Payment confirmation form.

Figure 5. Life cycle of a Purchase order Form within the supply-chain as viewed from the Construction company node.

Table 3. Overall connectivity between the processes and the forms supporting the ‘purchase order transaction scenario’.

<table>
<thead>
<tr>
<th>Process Id</th>
<th>Process</th>
<th>Name of the form</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-3 Send purchase order to manufacturer</td>
<td>Purchase order form</td>
</tr>
<tr>
<td>2</td>
<td>3-5 Notify goods supply date to customer</td>
<td>Goods supply date form</td>
</tr>
<tr>
<td>3</td>
<td>3-6 Send goods collection order to transporter</td>
<td>Goods collection order form</td>
</tr>
<tr>
<td>4</td>
<td>4-9 Notify goods collection date to customer</td>
<td>Goods collection date form</td>
</tr>
<tr>
<td>5</td>
<td>4-10 Notify goods delivery date to site</td>
<td>Goods delivery date form</td>
</tr>
<tr>
<td>6</td>
<td>3-11 Handover goods to transporter</td>
<td>Goods handover form</td>
</tr>
<tr>
<td>7</td>
<td>4-12 Deliver goods to customer’s chosen site</td>
<td>Goods receipt form</td>
</tr>
<tr>
<td>8</td>
<td>2-5 Notify receipt of undamaged goods to construction company</td>
<td>Goods receipt confirmation form</td>
</tr>
<tr>
<td>9</td>
<td>1-11 Notify payment to supplier</td>
<td>Payment confirmation form</td>
</tr>
</tbody>
</table>

Note: the first digit of the process represents the SC node number, which is followed by a dash and the corresponding process list number of that node.


These provide comprehensive coverage for process relationships and include concepts for concurrency, preserving order and repetitive processes. It also provides concepts to capture state and/or time effects, either those which are preconditions for an activity or those which are the effects of an activity.

By formally defining SC communication processes using PSL concepts it should be possible to identify the extent to which two processes are the same. For example we could
have two processes, both called ‘send’, but can we tell if they mean the same? The following section describes a tool which enables PSL concepts called ‘complex activities’ to be analysed and compared.

4.2. Overview of the PSL_20 questions wizard

The PSL_20 questions wizard has been developed by NIST as a tool facilitating the selection of the most appropriate PSL concepts corresponding to the set of ten generic properties set-up for a process. These concepts are selected by the tool based on the activities inherent within the process and the answers chosen by the user to the questions proposed by the tool in a questionnaire. A screen shot of the PSL_20 questions wizard questionnaire is shown in figure 7.

This questionnaire has ten questions with multiple answers from which the user would choose/tick only one answer for each question. These questions have been set-up around three topics: Constraints on branch structure, Variation of branch structure and Distribution of complex activity occurrences. Question 1 and 2 cover the first topic; question 3–6, cover the second topic and finally question 7–10, cover the third topic. Once the questionnaire has been completed and sent by the user to the system, a response is provided on the screen, as illustrated in figure 8.

Each of the ten PSL concepts chosen by the wizard from a set of concepts, directly links to one of the ten generic properties of the process. That is ‘parent–children’ relationship exists between the generic properties of a process and the corresponding PSL concepts. The list of these ten generic properties of a process is given in table 4.

It must be emphasized here, that these concepts are rigorously/mathematically defined within the PSL (Knutilla et al. 1998, Schlenoff et al. 1999, Gruninger 2003) and hence there should be no scope for misunderstanding their meaning and using them incorrectly. For example, unless all the conditions embedded within the definition are satisfied, the wizard would not select these concepts. The wizard will only select these concepts provided the user has selected the correct answer box from the multiple choice answer lists. The output as shown in figure 8 could be treated as a PSL translation definition of the process or the mapping of the process in terms of the PSL concepts. This mapping could later be analysed to identify the interoperability issues among other processes of the system.

4.3. Application of the PSL_20 question wizard tool

4.3.1. Example. As an example, the PSL_20 question wizard has been applied to the process called ‘Send_purchase_order’ of table 3. Figure 8 shows the response provided by the wizard. This translation definition highlights ten properties of the process in terms of ten PSL concepts such as:

- **Name:** Global Construction Ltd
- **Address:** 12, Exhibition Road, London SW11 5NZ, England
- **Tel no:** 01504-697 8586
- **Contact:** Sales order manager

**PO ref. Number:** XXXX

**Project ref. Number:** 0024

**Accounts ref. Number:** N/Y/0024/AC/005

**Name:** Custom Doors Ltd

**Address:** 23 High St, Loughborough, LE12 5GB, England

**Supplier contact:** Sales order manager

**Supplier Tel no:** 01013-6789

**Quantity:** 10 pieces

**Item:** Door

**Item type:** Plane varnished plywood door

**Item size:** 6.50 length × 3.50 width × 2inch thick

**Accessories:** Complete with handle & door lock

**Special requirement:** To be delivered at our const. site

Figure 6. Typical information in a Purchase order Form.
Repeating the PSL mapping method with other SC processes, allows us to identify where semantically similar processes exist. If the semantic translations of two processes are very similar, then there is likelihood that these two processes are shareable or interoperable.
Table 4. The ten generic properties of a process for which the wizard has been set-up.

<table>
<thead>
<tr>
<th>Property No</th>
<th>Generic properties of a process</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Activity occurrences</td>
</tr>
<tr>
<td>2</td>
<td>Atomic sub-activity occurrences</td>
</tr>
<tr>
<td>3</td>
<td>Constraints on the set of minimal activity trees for the complex activity</td>
</tr>
<tr>
<td>4</td>
<td>Constraints on the set of minimal activity trees for the complex activity based only on the state</td>
</tr>
<tr>
<td>5</td>
<td>Constraints on the branch structure of occurrences of the complex activity based on the time</td>
</tr>
<tr>
<td>6</td>
<td>Constraints on the branch structure of occurrences of the complex activity based both on state and time</td>
</tr>
<tr>
<td>7</td>
<td>Constraints on the occurrence of the complex activity based only on the state</td>
</tr>
<tr>
<td>8</td>
<td>Constraints on the occurrence of the complex activity based only on the time</td>
</tr>
<tr>
<td>9</td>
<td>Constraints on the occurrence of the complex activity based both on the state and the time</td>
</tr>
<tr>
<td>10</td>
<td>Constraints on the occurrence of the complex activity based both on the state and the time</td>
</tr>
</tbody>
</table>

Therefore based on this argument, the PSL_20 questions wizard has been applied to the set of ten processes that are identified in our scenario (see table 3) and the translation results are analysed in the next subsection.

4.3.2. Evaluation of the set of processes as translated by the PSL_20 questions wizard. The analysis of the scenario and the list of ten processes identified and shown in table 3 prompted us to classify these processes into four generic process types. These are:

a. SEND: formulated around activities involving monetary transactions;
b. NOTIFY: formulated around activities involving response by supplier to customer’s requirements;
c. HANDOVER: formulated around handling activities of goods, which are finished, semi-finished or raw materials, and responded by supplier based on customer’s requirements.
d. DELIVER: formulated around logistics/transporting activities of finished goods, semi-finished goods or raw materials and responded by supplier based on customer’s requirements.

The ten specific processes have been allocated under the relevant generic process class as shown in the table 5.

We might reasonably expect that all the processes, which are allocated under any of these four generic processes as shown in table 5, should be semantically similar. Also, the processes belonging to different generic process class would not be semantically similar. This is not always the case, which will be evident from the following analysis. The translation results of the ten processes in our scenario are analysed for semantic similarities by comparing the ten generic PSL properties of each process with the corresponding properties of the other processes. This leads to four groupings of these ten processes as shown in tables 6(a)–6(b).

These groupings can be used to explore the potential for share-ability of the processes. For example, table 6(a) shows the translation results and the grouping of the process 1 and the process 3 under the generic process class ‘SEND’. Both processes are semantically similar in terms of the ten PSL concepts. We can conclude that there is a scope of sharing these two processes which is in line with our expectations. Similarly we can expect the ‘NOTIFY’ processes in table 6(b) to be sharable.

However, when we analyse the ‘NOTIFY’ processes in table 6(c) we find that these have a range of different PSL properties. For example if we examine process 2 and process 4 as presented in table 6(c) it may be seen that these two processes are semantically different, because property numbers 6 and 8 of these two processes are different. Similarly if we consider process 4 and the process 10 of the same table, they are also semantically different, because their properties: 1, 5, 6, 7, 8, 9 and 10 are all different.

There are many reasons for these differences which can be better understood by further investigation. For example, if we look more closely at property number 8 for process
numbers 4 and 10. For process 10, the value assigned to property 8 is ‘trigger’. The reason for allocating this property to this process is because this process is triggered once all the relevant information are received from all the four participating nodes of the SC. On the other hand, the ‘Partial_trigger’ chosen by the PSL wizard for the process number 4, is owing to fact that it waits for information only from within the node itself, in this case the Transporter. In this manner, analytical reasons for selecting the properties of the process by the wizard can be developed.

Another interesting observation from table 6(d) is that all the ten properties of the process 7 and process 8 are same, although they are allocated under different generic process class in this case – ‘HANDOVER’ and ‘DELIVER’. In this case, although the names used are different the processes are semantically the same.

Analysis of this nature indicates the importance of properties of processes in terms of the PSL concepts and also suggests that although we can allocate several processes of a scenario under one generic process term, it is not necessary that they will be semantically similar.

Again, evaluation of the set of processes as translated by PSL_20 question wizard eventually leads to overall groupings of processes as shown in table 7.

From this evaluation, we can say that the PSL tool, has enabled us to analyse and classify these ten processes into semantically similar and non-similar categories as shown in table 7, which, we would not have been able to do by

<table>
<thead>
<tr>
<th>Property No</th>
<th>Process 1: Send purchase order to manufacturer</th>
<th>Process 3: Send goods collection order to transporter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Simple</td>
<td>Simple</td>
</tr>
<tr>
<td>2</td>
<td>Nondet_folded</td>
<td>Nondet_folded</td>
</tr>
<tr>
<td>3</td>
<td>Variegated</td>
<td>Variegated</td>
</tr>
<tr>
<td>4</td>
<td>Conditional</td>
<td>Conditional</td>
</tr>
<tr>
<td>5</td>
<td>Partial_time_conditional</td>
<td>Partial_time_conditional</td>
</tr>
<tr>
<td>6</td>
<td>Partial_mixed_conditional</td>
<td>Partial_mixed_conditional</td>
</tr>
<tr>
<td>7</td>
<td>Restricted</td>
<td>Restricted</td>
</tr>
<tr>
<td>8</td>
<td>Trigger</td>
<td>Trigger</td>
</tr>
<tr>
<td>9</td>
<td>Partial_launch</td>
<td>Partial_launch</td>
</tr>
<tr>
<td>10</td>
<td>Partial_conditional_launch</td>
<td>Partial_conditional_launch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property No</th>
<th>Process 5: Notify goods arrival instruction to site</th>
<th>Process 6: Notify goods delivery date to site</th>
<th>Process 9: Notify receipt of undamaged goods to construction company</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Partial_permuted</td>
<td>Partial_permuted</td>
<td>Partial_permuted</td>
</tr>
<tr>
<td>2</td>
<td>Partial_folded</td>
<td>Partial_folded</td>
<td>Partial_folded</td>
</tr>
<tr>
<td>3</td>
<td>Variegated</td>
<td>Variegated</td>
<td>Variegated</td>
</tr>
<tr>
<td>4</td>
<td>Partial_conditional</td>
<td>Partial_conditional</td>
<td>Partial_conditional</td>
</tr>
<tr>
<td>5</td>
<td>Partial_time_conditional</td>
<td>Partial_time_conditional</td>
<td>Partial_time_conditional</td>
</tr>
<tr>
<td>6</td>
<td>Partial_mixed_conditional</td>
<td>Partial_mixed_conditional</td>
<td>Partial_mixed_conditional</td>
</tr>
<tr>
<td>7</td>
<td>Restricted</td>
<td>Restricted</td>
<td>Restricted</td>
</tr>
<tr>
<td>8</td>
<td>Partial_trigger</td>
<td>Partial_trigger</td>
<td>Partial_trigger</td>
</tr>
<tr>
<td>9</td>
<td>Partial_launch</td>
<td>Partial_launch</td>
<td>Partial_launch</td>
</tr>
<tr>
<td>10</td>
<td>Partial_conditional_launch</td>
<td>Partial_conditional_launch</td>
<td>Partial_conditional_launch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property No</th>
<th>Process 2: Notify goods supply date to customer</th>
<th>Process 4: Notify goods collection Date to customer</th>
<th>Process 10: Notify payment to supplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Partial_permuted</td>
<td>Partial_permuted</td>
<td>Nondet_permuted</td>
</tr>
<tr>
<td>2</td>
<td>Partial_folded</td>
<td>Partial_folded</td>
<td>Partial_folded</td>
</tr>
<tr>
<td>3</td>
<td>Variegated</td>
<td>Variegated</td>
<td>Variegated</td>
</tr>
<tr>
<td>4</td>
<td>Conditional</td>
<td>Conditional</td>
<td>Conditional</td>
</tr>
<tr>
<td>5</td>
<td>Partial_time_conditional</td>
<td>Partial_time_conditional</td>
<td>Rigid_time_conditional</td>
</tr>
<tr>
<td>6</td>
<td>Partial_mixed_conditional</td>
<td>Mixed_conditional</td>
<td>Rigid_mixed_conditional</td>
</tr>
<tr>
<td>7</td>
<td>Restricted</td>
<td>Restricted</td>
<td>Local</td>
</tr>
<tr>
<td>8</td>
<td>Trigger</td>
<td>Partial_trigger</td>
<td>Trigger</td>
</tr>
<tr>
<td>9</td>
<td>Partial_launch</td>
<td>Partial_launch</td>
<td>Launch</td>
</tr>
<tr>
<td>10</td>
<td>Partial_conditional_launch</td>
<td>Partial_conditional_launch</td>
<td>Unconditional_launch</td>
</tr>
</tbody>
</table>
interoperability problems amongst the nodes and leads to fun
similar across SC nodes. However, their requirements and
within SC nodes and the names of processes can be very
development of a software system. Many processes occur
The identification of processes is very important for the

5. Discussion and conclusions

The identification of processes is very important for the
development of a software system. Many processes occur
within SC nodes and the names of processes can be very
similar across SC nodes. However, their requirements and
function may be different, which creates the potential for
interoperability problems amongst the nodes and leads to
the cross-disciplinary communication problems in SC
management.

The current paper has shown an approach to the
evaluation of processes in a SC in terms of the semantic
compatibility. This has been achieved by identifying the
nature of information flows, business nodes and processes.
For a particular node, some information is generated
within the organization and others are received from
external sources. Sometimes the names of information
processes used by different nodes are the same but the
information content may be different. This necessitates the
formal analyse of information processes to identify where
semantic similarities or differences exist.

We have shown in this paper that a communication
language, which can support process interoperation, is
required to have the following features:

a. sufficient mathematical rigour to enable computa-
tional comparisons of the meaning of terms;
b. formal definition of process concepts as well as the
more generally accepted entity information con-
cepts;
c. process concepts should include a broad range of
constraints which may be applicable, such as
sequencing, state, time and resource.

PSL, although developed for discrete manufacturing
processes, appears to provide sufficiently generic and
rigorous foundation ontology from which to compare the
process semantics between SC processes. Although there is
a need to develop more effective methods for mapping
system specific terminology to PSL, it can still be seen as a
powerful tool to aid the communication between processes
in the SC. This is because of the combination of underlying
mathematical rigour in the definition of its concepts with
the comprehensive range of process definitions and
constraints which it offers. The key problem appears to
lie in the effective mapping from SC concepts to PSL
concepts which requires a mapping from engineering
expertise to the expertise of a logician.

The PSL wizard provides a current method available
which aims to support the identification of mappings
between engineering processes and PSL concepts. While
this is very useful it does have some significant draw-
backs in providing an effective and reliable route to
achieving these mappings. First, to answer the questions
presented by the wizard, the user must have a clear grasp
of set theory and logic, as the questions are written in
terms which are not completely familiar with these. A rewording
of the wizard, with supporting explanations of the questions
being asked would be helpful. This is particularly
important as a slight misunderstanding of the question
can lead to different experts providing different answers
and hence nullifying the value of the mapping process.
For the commercial use of PSL in the future, it is
necessary to develop translation rules or guidelines to
help the user to identify the appropriate concepts of the
language. It is likely to be necessary to develop a range
of wizards for different SC domains, e.g. different wizards
for manufacturing SCs and tourism SCs. There is also an
issue in terms of the granularity with which a wizard will
be able to effectively operate e.g. will a manufacturing
SC wizard be able to support all type of manufacture or
will different wizards be needs for automotive, electronic
or food SCs?
In summary, while there is great potential for improved interoperability in the approach taken, there is still substantial research needed before effective commercial solutions are available.

Acknowledgements

The work reported in the present paper has been conducted as part of the X-Slang project. The current authors would like to acknowledge the support of the EPSRC which funded this project through Loughborough University’s ‘Innovative Manufacturing and Construction Research Centre’, under the Grant No. GR/N22526.

References
