THE IASDO MODEL FOR INFORMATION MANUFACTURING SYSTEMS MODELLING

(Completed academic paper)

Thanh Thoa Pham Thi
Dublin City University
Thoa.Pham@computing.dcu.ie

Markus Helfert
Dublin City University
Markus.Helfert@computing.dcu.ie

Howard Duncan
Dublin City University
Howard.Duncan@computing.dcu.ie

Abstract: One of the criteria that relates to information quality in information manufacturing systems is the method or model used for modelling these systems. In essence, an information manufacturing system includes information, processes that consume/produce information and organisational responsibilities for the management of processes (responsibility) and management of information (privilege). Therefore, a model or method for information manufacturing system modelling firstly has to be able to describe all these elements, and secondly describe them in an accurate and consistent manner. In this paper, we review some actual models and methods for modelling information manufacturing systems and outline some limitations of current approaches. Then we propose a model for modelling information manufacturing systems, the IASDO model, which satisfies the above requirements and thus helps to improve information quality.

Key Words: Information Quality, Information Product, information manufacturing system, modelling methods, dynamic specialization.

1. INTRODUCTION

Over the last decade, academics have addressed challenges in information quality and have proposed several methods to measure and improve information quality [9]. Companies have developed various tools to cleanse and analyze data in databases and data warehouse systems [8]. One innovative method is to manage information as an information product (IP) and to view the manufacture of an IP as a sequence of processes that must be represented accurately [10]. In this view, manufacturing an IP is akin to the manufacture of a physical product in a supply chain network. Raw materials, storage, assembly, processing, inspection, rework, and packaging (formatting) are all applicable to an IP.

An essential element of manufacturing an IP are the underlying processes. However information and organizational aspects should not be neglected. Controlling and managing a process requires accurate information. In addition, information manufacturing processes are far more complex to manage than industrial manufacturing processes, which often include robots or automation. Processes in information manufacturing may need organizational responsibilities for its execution.

In order to describe such systems, it is common to describe the required system elements, their structure and the process dynamics through various models. These are commonly referred as enterprise engineering
or Information System Architecture [4]. Typically, to develop and maintain a consistent and adequate (enterprise) model, thus capturing user requirements and dynamic changes in the enterprise environment, is challenging. In the context of an information manufacturing system, a model should address at least three aspects:

- Static aspect that concerns information and information structure in the form of raw data, component data or information product.
- Dynamic aspect that concerns information processing.
- Organizational aspect that concerns the management of the information system at an organizational level.

Although attempts have been made to provide models for information manufacturing systems, certain critical aspects of the stages that an IP goes through within the manufacturing system have not been addressed. Most current approaches lack the ability to represent the dynamic changes involved in manufacturing (or creating) an IP in a systematic way that allows information to be tracked and to represent these changes in relation to process management and organizational aspects. Besides, they often do not take into account the interrelation between these aspects to ensure a consistent specification.

In order to overcome the shortcomings of current approaches, we propose in this paper an object-oriented model for IMS modeling; the IASDO model (Integrational Aspects of Static, Dynamic and Organization). The IASDO model can explicitly describe concepts of the three aspects mentioned above. In addition it allows consideration of the interrelation between different views, and thus by integration of these aspects the IASDO model ensures an accurate and consistent specification.

The rest of the paper is organized as follows. Section 2 presents an overview of some actual methods and models for IMS modelling and an illustration of these methods and models with modelling information manufacturing in library management. A discussion on their shortcomings is also presented. Section 3 aims to describe the IASDO model including its basic concepts. Section 4 illustrates the IASDO model with modelling of a library management case and then we show how information quality is improved within this model. Section 5 presents a conclusion.

### 2. RELATED WORKS

Information manufacturing system modelling has attracted several researches. Numerous methodologies or models have been proposed to model the process by which the information product is manufactured such as IP-UML [18] or IP map [3]. In addition, some other approaches are related to IMS modeling in a broader sense. In this article we selected two such approaches, Data Flow Diagrams [17] and Event Driven Process Chains - EPC [11].

IP Map focuses on representing details associated with the manufacture of an IP. The model uses graphic constructs to describe process, input and output information of processes, customer receiving output information, decisions, IS boundary, organization boundary and department/role concept associated with each construct representing organizational responsibility. IP Maps are typically used to describe the composition of one information product, and thus relations between the manufacture of several information products are often not considered. Besides, they do not describe dynamic changes of information.

An adaptation of the IP-Map concept is the IP-UML approach, which combines IP map and UML by proposing a data analysis model, a quality analysis model and a quality design model. The data analysis model using UML aims to identify interesting data and their composition by proposing stereo-types of class such as raw data, component data and information product. An information product is composed of raw data or component data or both. The quality model identifies quality requirements on each data item.
specified in the data analysis model. The quality design model combines the IP map and activity diagram of UML and describes data and process together in order to verify the satisfaction of quality requirement. However, it is a local description of the manufacture of an IP product, in that it does not represent relations between different information manufacturing processes and relations between information products. Furthermore, neither dynamic changes of information nor the privileges of organizational units for information access (security of data) have been described.

Data Flow Diagrams (DFD) have been used for many years. They focus on describing processes as well as their inputs and outputs. External entities and data stores are also constructions used in the model. External entities may send input data to processes and receive output data from processes. Because the model focuses on processes, relations between data are not described and we do not know how the data structure is changed by processes. Furthermore, the model does not take into account organizational structure, so neither responsibility for processes nor control of data access are mentioned.

Event Driven Process Chains are typically used to describe business processes. They describe business process, information exchanged, and functions of organization. This model includes graphic constructs to model functions, events, input data, output data, organizational units and the interconnection between them. The model does not systematically take into account consistent modelling within interrelations of all these constructs. In the model, neither relations between data nor dynamic changes of data through processes are described. Furthermore, the model does not mention control of data access.

In the following, we illustrate the IP Map and EPC model, those represent two modelling approaches: IP oriented and business process oriented, by modelling a library management application using the following scenario. A library holds copies (books) that may be borrowed by readers. Copies are units representing physical books that belong to one logical document. A logical document has one or many correspondent copies. Readers request a loan by indicating the documents needed. The librarian can make a loan or a reservation for the reader depending on the status of the corresponding copies of the document. In the case of a reservation, when there is a copy available the librarian will notify the reader and put a block on that copy. The reader can then borrow the blocked copy which becomes a borrowed copy, or the reader can cancel the reservation made and the copy is unblocked. The workflow is finished when the reader returns the copy or cancels his reservation.

Figure 1 is a specification of the process Create a loan for creating Loan with IP Map. Select document, Request a Loan, Create a loan, Create a reservation are processes. Loan and Reservation are information products. Reader is a source of data as well as a consumer because it provides information for creation of information product Loan and Reservation.

When a loan is created, as a consequence a borrowed copy is created as well. With IP Map, relations between Document and Copy; Copy and Borrowed Copy; Reservation, Loan and Request are not described. The designer may add integrity rules to ensure a consistent specification, for example that the borrowed copy of a loan belongs to the same document as that requested at the beginning. Because there is no relation between information (for example Copy and Borrowed Copy), the specification does not describe how information changes step by step through manufacturing process.
Figure 1. Process *Create a loan* for creating Loan described with IP-Map

Figure 2. Business process *Make a loan* described with EPC
Figure 2 presents a description of the business process *Make a loan* with EPC model. *Reader requests a loan, Copy available, No copy available* are events. Librarian is an organizational unit who is responsible for execution of processes: *Make a request, Make a loan and Make a reservation*. After execution of processes *Make a loan* and *Make a reservation*, there are some modifications of data such as change state of request and copy (or creation of new data) on request inventory and copy inventory. As in the specification in Figure 1, this specification lacks relations between information products in order to describe a precise specification and rules to ensure a consistent specification. Organizational concept is introduced in the model; it describes responsibility for the process execution; however the security on information accessibility of organizational units is not mentioned.

A summary of descriptions of models mentioned above and some remarks is given in Table 1.

<table>
<thead>
<tr>
<th>Methodology/Model</th>
<th>Data</th>
<th>Process</th>
<th>Organization</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Map</td>
<td>-Source (input) -Data storage</td>
<td>-Information processing -Check for data quality -Decision</td>
<td>-Department/Role -Business boundary -Information system boundary</td>
<td>-Focus on manufacturing of one IP product -Do not describe dynamic changes of information -No relation between information products</td>
</tr>
<tr>
<td>IP-UML</td>
<td>-Data analysis model is based on class diagrams of UML -Quality analysis model is based on class diagrams of UML</td>
<td>-Quality design model is a combination of IP Map and Activity diagrams of UML</td>
<td>-Actor is responsible for activity execution</td>
<td>-Focus on manufacturing of one IP product -No relation between information products -Do not describe dynamic changes of information -Privilege on data access is not mentioned</td>
</tr>
<tr>
<td>EPC</td>
<td>-Input data -Output data</td>
<td>-Business function and its logic connectors -Event</td>
<td>-Organization unit is responsible for business function execution</td>
<td>-No relation between data -Do not describe dynamic changes of data -Privilege on data access is not mentioned</td>
</tr>
<tr>
<td>Data Flow Diagram (DFD)</td>
<td>-Input data -Output data -Data store</td>
<td>-Process</td>
<td>-External entity send/receive data</td>
<td>-No relation between data -Responsibility for process execution is not mentioned -Privilege on data access is not mentioned</td>
</tr>
</tbody>
</table>

Table 1. Constructions of some methods and models for IMS modelling
3. IASDO MODEL
The IASDO model [15] covers concepts of the static, dynamic and organizational aspects of information manufacturing systems. Basic concepts of the model are as follows.

Static concepts
Static concepts present a static view of the system that includes the structure of information and relations between information entities. Information may be raw data (material data), component data or information product. All of these kinds of information are modeled using the class concept. The existential dependency (ED) is the only kind of relation between classes. Using only the existential dependency relation a much precise semantic of data description can be obtained than with other kinds of relation and it allows the model to support evolution [12, 13].

An ED relation (or imperative ED relation) between a class C1 and a class C2 means that the existence of any object o1 of the class C1 depends on one and only one object o2 of the class C2. This dependency is permanent through the life of object o1. For example, there is an ED relation between a class Copy and a class Document. A copy cannot exist if it does not link to a corresponding document. The ED relation is transitive.
A non-imperative ED is an optional existential dependency, i.e. an object o1 in class C1 depends on zero or one objects o2 in class C2.

Dynamic specialization (DS) [2, 15] is a particular case of ED relation. The DS is described by the following features: DS graph, active/inactive status, access-view, and loops.
- DS graph: A DS graph is defined by a set of classes (nodes) linked by DS links (edges). If a class C1 is linked to a class C2 by a DS link, then C1 is a direct sub-class of C2, or C2 is a direct super-class of C1. For any object o1 (sub-object) of class C1, its existence depends on existence of one and only one object o2 (super-object) of C2.
  A DS graph has one root class. An object can belong to several classes in the same DS graph. If an object belongs to a class C then it belongs to all super-class of C (i.e. direct and indirect super-classes).
- active/inactive status: An object that belongs to a class can be active (default status) or inactive in that class. When an object changes from a class C2 to its sub-class C1, it may remain in C2 or leave it. This is indicated by the specialization link. In the first case, the object remains active in the super-class C2, while in the second case it becomes inactive in C2. An inactive object in a class can not be updated, but can be consulted, i.e. one can get its attribute values.
- access-view: An access-view of a class C consists of all the properties (attributes and methods) defined on class C, and selected (by the designer) properties of the super-classes of C. Thus an access-view of C allows the properties of the super-classes of C to be show or hidden. This differs from classical inheritance.
- loop: As we mentioned before, an object can migrate from a class C to its sub-classes. There are some situations in a DS where an object may migrate from a sub-class C1 to one of its super-classes; this is a loop situation and it requires keeping the trace of the object. Figure 3 presents a loop situation; class Copy is a super-class of AvailableCopy which is a super-class of BorrowedCopy. This in turn is a super-class of Returned copy. An Available copy object becomes Borrowed copy by a loan, a Borrowed copy becomes available after being returned. Returned copy is a looped class of Available copy. It allows a trace of copy to be kept at beginning of a borrowing and the end of a borrowing.

Figure 3. Example of loop situation in dynamic specialization
A dynamic specialization relation may be imperative or non-imperative. In case of mono-specialization (i.e. a sub-class has only one direct super-class) the specialization link must be imperative; in case of multi-specialization (i.e. a sub-class may have several direct super-classes) a sub-object must be linked to at least one super-object.

**Dynamic concepts**
Dynamic concepts represent information processing. At the local view, the object life-cycle belongs to dynamic concepts that describe changes of an object’s (or information) state caused by process execution. A process corresponds to a business function or a decision.
At the global view, dynamic concepts concern interaction between objects; business processes and information consumed or produced by processes that cause a change of the system’s state. Our model takes into account both the local view and the global view.
The object life-cycle is presented in our model with state and process concepts. A process execution changes objects in input states into output states. The differences between our model and the others like UML state diagrams [5] and object life-cycle with Petri net are follows:
- An object can remain in or abandon a state when it changes to another state. So at any time an object may reside in several states. For example, when an available copy changes to borrowed copy state, it abandons available copy state; but if a European country changes to UE member state, it keeps on being European country.
- A process is associated with execution rules that call pre- and post-conditions. A process is valid (able to execute) if objects in its input states satisfy its pre-condition. After the process execution, its post-condition is satisfied. These conditions are logical expressions including logical connectors (AND, OR, XOR) and combinations of them, which are defined by the designer. There is no convention on logic connectors as in Petri nets.

**Organizational concepts**
The principal concept in the organizational aspect is the concept of an organizational role. An organizational role corresponds to an organizational unit, which is assigned responsibilities and functions that should be carried out to achieve the objectives of organization. A process may be under the responsibility of zero (in case it is an automatic process), one or several organizational roles. An organizational role may be responsible for zero, one or several processes. An organizational role has privileges on information accessibility such as creation, suppression, modification, and query.

**IASDO model definition**
The IASDO model presents all concepts mentioned and the interrelation between them in the same model. At this global view, we consider a state in the object life-cycle of a class C as a sub-class of the class C because a state of C concerns more specific objects of C that satisfy certain conditions. Furthermore the order of changes of state in the object life-cycle must correspond with the order of object migration in dynamic specialization to ensure consistency between these two aspects. In this view, an object life-cycle in dynamic aspect corresponds to a dynamic specialization graph in static aspect. Consequently, the input and output states of a process become information consumed and produced by the process with respect to execution rules.
Interrelations between the organizational aspect and the other two aspects are described by the fact that an organizational role has responsibility for process and privileges on information. This allows the management of process and information in information manufacturing.
A complete specification in the IASDO model is defined with:
IASDO model = \(<\text{CL}, f_{ed}, f_{ds}, \text{back-inactive}, \text{access-view}, \text{loop}, \text{P}, f_{in}, f_{o}, \text{R}, \text{Pr}, f_{rcl}, f_{rp}>\)

- \text{CL}: a set of classes \(\{\text{cl1, cl2, \ldots, cln}\}\).
  A class \text{cli} = \langle\text{Name, Att, Met}\rangle, \text{cli.Att} is the set of attributes of \text{cli}; \text{cli.Met} is the set of methods of \text{cli}.
- \(f_{ed}\): an existential dependency function, \((\text{CL x CL}) \rightarrow \{0,1,2\}\).
  \(f_{ed}(\text{clj}, \text{cli}) = 1\) if there exists an imperative existential dependency from the class \text{clj} to the class \text{cli}; The value 2 indicates that the existential dependency is non-imperative, and the value 0 indicates the absence of existential dependency from \text{clj} to \text{cli}.
- \(f_{ds}\): a dynamic specialization function, \((\text{CL x CL}) \rightarrow \{0,1,2\}\).
  \(f_{ds}(\text{clj}, \text{cli}) = 1\) if there exists an imperative dynamic specialization link from the class \text{clj} to the class \text{cli}, i.e. \text{clj} is a direct sub-class of \text{cli}. The value 2 indicates that the dynamic specialization link is non-imperative, and the value 0 indicates the absence of dynamic specialization from \text{clj} to \text{cli}.
- \text{back-inactive}: \((\text{CL x CL}) \rightarrow \{0,1\}\), if \text{back-inactive}(\text{clj}, \text{cli}) = 1 then when an object in class \text{cli} becomes an instance of class \text{clj} then the object becomes inactive in class \text{cli} (\text{clj} is an ancestor of \text{clj}).
- \text{access-view}: \text{CL} \rightarrow \{\text{CL.Att, CL.Met}\}, \text{access-view}(\text{clj}) = \{\{\text{clj.Att} \cup \text{clj.Accessible.Att}\}, \{\text{clj.Met} \cup \text{clj.Accessible.Met}\}\} where 
  \text{clj.Accessible.Att} \subseteq (\cup \text{cli.Att}) and \text{fds(clj, cli)}=1, i=1..n, j=1..n, i\neq j
  \text{clj.Accessible.Met} \subseteq (\cup \text{cli.Met}) and \text{fds(clj, cli)}=1, i=1..n, j=1..n, i\neq j
- \text{loop}: a function describes the end and start classes of a loop, \text{loop}: \text{CL} \rightarrow \text{CL}.
  if \text{loop}(\text{clj}) = \text{cli}, then \text{clj} is the end of a loop which has its start at \text{cli} (\text{clj} is an ancestor of \text{clj}); \text{clj} is called looped class of \text{cli}. This has the following consequence: when an object is created in \text{clj}, it is created again in \text{cli}.
- \text{P}: set of processes, a process has pre and post conditions.
- \(f_{i}\): \((\text{CL x P}) \rightarrow \{0,1\}\), if \(f_{i}(\text{cl}, \text{p}) = 1\) then \text{cl} is an input class of the process \text{p};
- \(f_{o}\): \((\text{P x CL}) \rightarrow \{0,1\}\), if \(f_{o}(\text{p}, \text{cl}) = 1\) then \text{cl} is an output class of the process \text{p};
- \text{R}: set of roles.
- \text{Pr}: set of privileges on information, \text{Pr} = \{\text{create, modify, delete, query}\}\).
- \(f_{rcl}\): assigns privileges on a class to a role, \((\text{R x CL x Pr}) \rightarrow \{0,1\}\);
  if \(f_{rcl}(\text{r,cl,pr}) = 1\) then the role \text{r} has the privilege \text{pr} on the class \text{cl}.
- \(f_{rp}\): assigns responsibility for a process to a role, \((\text{R x P}) \rightarrow \{0,1\}\);
  if \(f_{rp}(\text{r,p}) = 1\) then the role \text{r} is responsible for the process \text{p}, so a role \text{r} may be responsible for many processes and vice-versa.

A specification with the IASDO model includes graphic specifications and text description for detailed information about the attributes and methods of classes, active/inactive objects, access-view information and privileges information.

Concerning the interrelation between three aspects, the model defines additional rules for a consistent specification.

**R1**: An output class of a process has to link to at least one input class of this process by ED or DS link. This rule ensures the consistency between static and dynamic aspect: output information of a process cannot exist if there is no input information to the process.

**R2**: Let \text{C1, C2} be two input classes of a process \text{P}; \text{C3, C4} are output classes of the process \text{P}. If an organizational role \text{R} is responsible for \text{P} then \text{R} has at least the privilege of information creation on \text{C3} and \text{C4}, the privilege of information consultation on \text{C1, C2, C3 and C4}. 
4. MODELLING INFORMATION MANUFACTURING SYSTEMS

In this section, we illustrate our model by modelling the information manufacturing system of the library management case. Then we show how our model allows information quality to be improved in IMS modelling.

Information is manufactured concerning Request, Loan (Borrowing), Reservation, and changing state of Copy.

**Specification with the IASDO model**

Figure 4 presents the information specification of the library management case. There is a dynamic specialization graph whose root class is Copy and another one whose root class is Request. Between Copy and Document is an ED relation.

---

**Figure 4. Information and relations between them in the library management case**

Textual description of the specification is as follows. Back-inactive(BlockedCopy, AvailableCopy) = 1 meaning that an object that migrates from class AvailableCopy to BlockedCopy becomes inactive in AvailableCopy. Similarly, we have back-inactive(BorrowedCopy, AvailableCopy) = 1, back-inactive(NotifiedReservation, Reservation) = 1, etc. The access view of class AvailableCopy is access-view(AvailableCopy) = \{available copy number, copy code, document number, available date, Borrow, Block\}. The access view of class BorrowedCopy is access-view(BorrowedCopy) = \{available copy number, copy code, document number, borrowing date, Return, Lose\}^1, etc.

---

^1 Borrow, Block, Return, Lost are methods of classes
By using dynamic specialization, the specification allows information about Copy and Document to be traced after each manufacture; relations between Borrowed Copy and Borrowing, Blocked Copy and Notified Reservation, etc. allow a precise specification. UnblockedCopy and ReturnedCopy are looped classes of the class AvailableCopy i.e. an unblocked or returned copy becomes again an available copy. The specification imposes rules on the order of state changes (e.g. it is impossible to change a borrowed copy to blocked copy because Borrowed Copy is a subclass of Blocked Copy), and on the parentage of sub-objects: a borrowed copy of a loan must belong to the same document as that requested by the request corresponding to the loan. These rules could not be defined in specifications with other methods or models.

Figure 5. Information processing in the library management case

Figure 5 illustrates information processing in the library management case. At the local view, the yellow part corresponds to the object life-cycle of Copy; the dotted part is the object life-cycle of Document. At the global view, this figure presents the information processing of information products. A process consumes input information and produces output information, in accordance with its execution condition. For example, the pre-condition of process Process request is ((Request and Available Copy) or Request); the post-condition is (Reservation xor (BorrowedCopy and Borrowing)), etc.

By integrating static and dynamic concepts, an object life-cycle corresponds to a dynamic specialization
graph, and so we obtain the result depicted in Figure 6. Organizational roles are described after each process’s name. In this specification, Librarian, Logistic service and Control service are organizational units. For example, Librarian is responsible for Request a loan process, Borrow process, Notify process, etc. Librarian has creation privileges on Request, Loan, Reservation, BorrowedCopy, BlockedCopy, etc. and query or consultation privilege on AvailableCopy, etc.

**Figure 6. Complete specification of information manufacturing of the library management case**

**Improving information quality with the IASDO model**
Comparing specifications made by the IASDO model with those of other methods and models mentioned above, we see that the IASDO model allows description of information products, processes and organizational concepts. Furthermore, relations between information products as well as relations between information products, processes and organizational roles are also explicitly described in this model.

Dynamic specialization is an integration of static and dynamic concepts. It can describe information
change step by step while keeping track of information. Execution rules are attached to processes; it provides a rigorous process description. Integration of organizational concepts with static and dynamic concepts allows organizational responsibility and control of information access to be described, as a result of the concrete definition of privilege on information. Therefore, the model allows information management and process management to be improved. Furthermore, when interrelation between different aspects is taken into account, it allows a consistent specification of information manufacturing.

Table 2 presents a summary of the IASDO model

<table>
<thead>
<tr>
<th>Data</th>
<th>Process with pre/post condition</th>
<th>Organization role</th>
<th>Relations</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information product (Class)</td>
<td>Process with pre/post condition</td>
<td>Organizational role</td>
<td>-Relation between IP: ED and DS relations</td>
<td>-Describing information products and relation between them with ED relation and DS relation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Relation between static and dynamic concepts thanks to dynamic specialization</td>
<td>- It not only focuses on process of one information product but also several information products to give a global view on processes and information of the system as well as relations between them</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Relation between organization and data, process: privilege on information, responsibility for process execution</td>
<td>- Allow to keep track of information</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Allow a rigorous description of process with pre/post condition</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Describing responsibility for process execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Describing security on information accessibility</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Allow a precise and consistent specification because of taking into account relations between these three static, dynamic and organizational aspects</td>
</tr>
</tbody>
</table>

Table 2. Summary of the IASDO model descriptions

5. CONCLUSION

Information quality in IMS modelling relates to the models and methods used. An information manufacturing system basically concerns processes, information consumed and produced by processes and organizational concepts for process and information management. The IASDO model defines constructions that allow information, information processing, organizational aspect and the interrelation between them to be described accurately, to ensure a consistent specification. Furthermore, this model also allows information to be traced through information processing. Therefore it is proposed for improving information quality in IMS modelling. In the future, we aim to apply this model into practical environments.

ACKNOWLEDGEMENT

We would like to acknowledge Prof. Dr. Michel Léonard, University of Geneva, Switzerland for his guidance on the development of the IASDO model.
REFERENCES