

# A Metamodel for Enabling a Service Oriented Architecture

Baba Piprani<sup>1</sup>, Chong Wang<sup>2</sup>, Keqing He<sup>2</sup>

<sup>1</sup> SICOM, Canada

<sup>2</sup> State Key Lab. of Software Engineering, Wuhan University, 430072, China

babap@attglobal.net,

wangchong\_wuhu@yahoo.com.cn

hekeqing@public.wh.hb.cn

**Abstract.** Process modelling initiatives generally develop their process models without much emphasis on data, burying their sequence of operations as a thread within a non-elementary process. More often than not, these buried operations are elementary atomic reusable components. The resulting models are generally not flexible or sufficiently reusable, suffering from update anomalies and redundancies. Addressing “service” as a major deliverable component, an ORM metamodel was developed in line with ISO 19763-5 Metamodel Framework for Interoperability: Metamodel for Process Model Registration, to harmonize atomic component processes using a control sequence and event models to enable the delivery of a totally flexible model set facilitating metamodel interoperability and cooperation between systems via their respective models. The paper provides a limited ORM based review of ISO 19763-5, and uses underlying component processes to develop a metamodel for a deliverable Services Oriented Architecture containing control sequence models, event models, and bridges to associated data models or web services.

**Keywords:** Services, Event Modelling, Service Oriented Architecture, ORM, ISO19763-5

## 1 Introduction

Many businesses suffer from weak IT infrastructures consisting of disconnected databases, applications and services. This is even reflected in the glaring lack of documented business processes and their automatable counterparts in the form of IT Process Models.

A Conceptual Schema (as in ISO TR9007[1]) essentially reflects the static and dynamic rules of the enterprise. Processes address the dynamics and the behavior rules of an enterprise. Process modeling approaches have been around for decades in one form or another, each having their own syntax and semantics.

Process models, involving business processes, workflow, Web services etc., are deemed as a special kind of information resource along with complex structure, rich semantics and behavioral features.

International Standards Organization activities and several industrial consortia have contributed to standardization of domain specific process models using various representation notations and description languages for focused domains, such as BPMN[2] (Business Process Modeling Notation) for business process and OWL-s for Web services[3].

Most process modeling approaches concentrate on the flow of control for operations, weaving a complex scenario that may include several re-usable individual standalone processes in the form of a “service”. It is this inflexible set that is weak in its foundation and is not adaptable to change. Noting that the processes represent the ‘how’ of things to be addressed and dynamic behaviour in the enterprise, focus is lost on the ‘what’ of the enterprise, i.e. the business facts and semantics or the static behaviour of the enterprise.

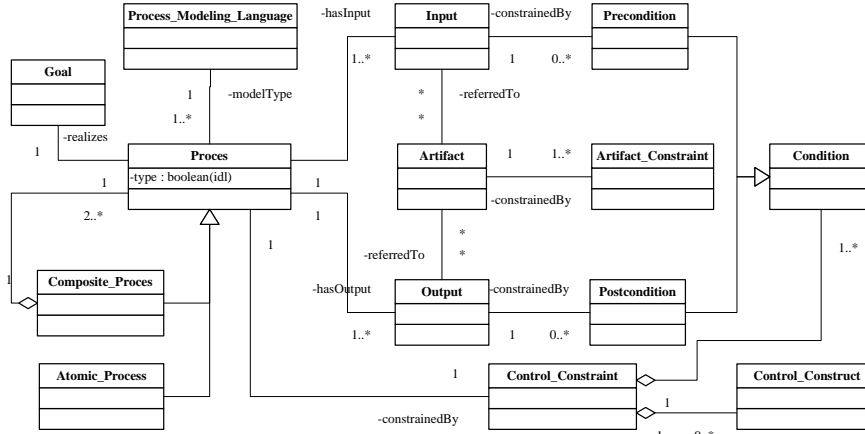
Change in an enterprise essentially is reflected more in the ‘how’ part and much less so in the ‘what’ part i.e. there is more change that is reflected in how the business is done vs. less change on the facts themselves. Take for example the purchase of an airline ticket for a flight. The process has gone through a dramatic change from a manual paper ticket operation without computers, through issuing of paper tickets using computers, through e-tickets using computers, i.e. the ‘how’. But the facts that a person is travelling on a particular flight from an emplaning city to a deplaning city still remain the same.

## **2 The ISO 19763-5 Metamodel**

In order to enable semantic interoperation between process models expressed in different modeling languages and promote further reuse based on them, ISO/IEC 19763-5[6][7] is introduced in this paper to provide a metamodel to register administrative structural information and meaningful semantics of process models, including workflows, business processes, Web services, software processes, etc. As an abstract facility, it focuses on the common structural and semantic content of process models expressed with different modeling languages, rather than their representations. Fig. 1 shows the overall structure of ISO/IEC 19763-5, i.e. Metamodel for process model registration(MPMR).

Concerning the construction of process models, Atomic\_Process and Composite\_Process are proposed to denote two kinds of process model. Atomic\_Process is the simplest process model and corresponds to one-step execution. In contrast to Atomic\_Process, Composite\_Process comprises at least two sub-processes, which can be atomic processes or other composite processes. For either of them, we should designate the modeling language that the registered process model adopts and the purpose that should be achieved by Process\_Modeling\_Language and Goal respectively. Since process model can be identified as the transformation of input to output[8], it is obvious that one process model will have one or more Input to generate one or more Output as desirable products. If each input or output is taken as an information deliverer, then the involved objects or resources can be treated as corresponding information carriers. So in MPMR, all the objects, data and resources used in the process model can be instances of Artifact. Moreover, each artifact might

play different roles specified by different communities in different cases. Therefore, artifacts respectively referred to the Input of one process and the Output of another process can be the same.



**Fig. 1.** Overall structure of Metamodel for process model registration (MPMR).

As for constraints between components within a certain process model, we define **Artifact\_Constraint** to record relationship between Artifacts, which can be derived from knowledge base of domain or ontologies that artifacts are contained in, such as equivalence relation between two concepts. It also can be used to add semantics to referred resources and connect process models semantically or semi-automatically. Relatively, process is restricted with **Control\_Constraint**. Particularly, due to the complexity of registered process models, two types of strategies are considered in MPMR. As for **Atomic\_Proces**, **Condition** is the only mandatory constraint, which has two subclasses, i.e. **Precondition** and **Postcondition**. **Precondition** is referred to **Input** to specify the information state that should be satisfied before execution, while **Postcondition** is restricted to **Output** to represent desirable outcomes when process is completed successfully. Considering **Composite\_Proces**, **Control\_Constraint** becomes more complicated. It comprises **Condition** and **Control\_Construct** because its sub-processes are connected with each other through at least one instance of **Control\_Construct**. Specifically, **Control\_Construct** here can be generalized as **AnyOrder**, **Choice**, **Join**, **Split** and **Sequence**. **AnyOrder** allows sub-processes to be executed in an unspecified order. **Choice** invokes one component of process model from a given collection. **Join** works when all of its components have been completed; **Sequence** means execution in order. **Split** produces at least two branches when the previous process model is executed successfully. Notice that inherent operation semantics of **Control\_Construct** should be considered when specifying **Precondition** and **Postcondition** of **Composite\_Proces**.

Furthermore, Fig. 2 depicts the ORM Schema of ISO19763-5 metamodel as per published transforms from UML to ORM [9]. It is presented here to facilitate verification of the constructs by ISO.

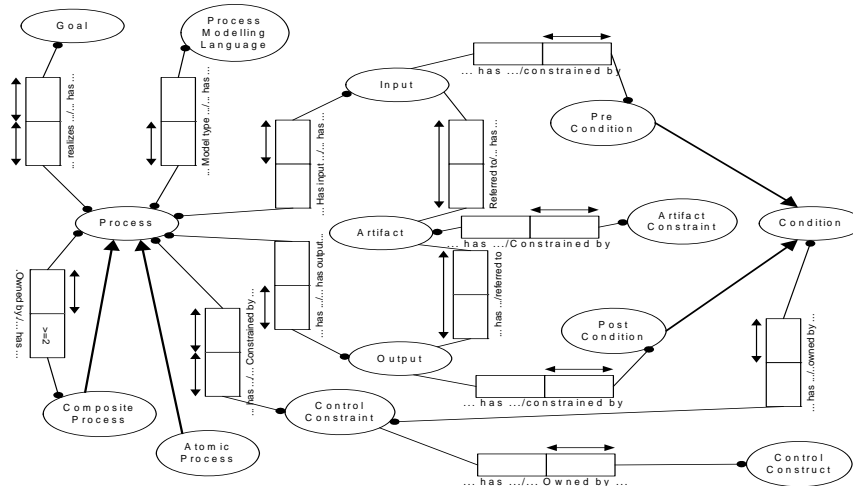


Fig. 2. ORM Schema of ISO 19763-5.

### 3 Positioning the Process Model in SDLC vis-à-vis Services and Data

Business processes form the backbone of an enterprise in terms of its infrastructure to deliver services in association with proper supporting information. In Figure 3 we define a framework for positioning the various components of the involved infrastructure where each component environment is represented through some degree of one or more formalized models. Although ORM has been used as a candidate for modelling data semantics it is important to recognize that this framework also represents a generic model driven environment---including overlaps with OMG's Model Driven Architecture.

#### 3.1 The Business Activity Model to the Semantic Modelling phase

An initial forest-level picture pegging the boundaries of the set of stated requirements is first defined through a form of functional business decomposition to establish the overall scope. A business activity contributes to the achievement of an objective of the business. Each business activity in the hierarchy is decomposed until the lowest level activities (called elementary business activities or atomic processes that only handle a single unit of work and cannot be split further without loss of business meaning). This last (or lowest) level is denoted as level "n". The business decomposition stops at level "n-1", i.e. the level when the activity involves an "automatable part" and still maintaining a "business part", and where a further decomposition results in an automatable process involving primitive computer facilities of input / output.

The procedures for defining business activities and decomposition may be done differently by different people. Decompositions of the same business may be arrived at with different results based on which set of criteria is chosen, like business functions, organizational etc.. This is quite acceptable and it does not matter, as long as all the business activities are being covered.

Why does this not matter? A business activity model is not a formal model i.e. there is not a formal grammar to support the business activity model. What matters is the data or information that is to be identified and formalized in the data usages of information flows from the lowest level process.

It is important that the lowest atomic processes represent a complete stand-alone, re-usable elementary task activity that cannot be split any further without losing meaning, and that these elementary tasks, while they may contain a processing sequence to accomplish that given elementary task, may not be connected or sequenced with other elementary tasks except for its own self to complete its given task---since this sequencing actually is a actually a service, and should be depicted by an independent stand-alone sequence model that controls the sequence of atomic processes.

A semantic data model is derived from the data usages in the information flows of these atomic processes. A semantic data model is a formal model with formal grammar associated with it, and is also known as a Computational Independent Model (CIM).

What this means, is that it does not matter how the business activities are organized, as long as the data usages have been recorded. No matter which alternate approaches of business activity modelling or decomposition is used---be it organizational based, product association based, business functionality based---the data usage information flows from the lowest level processes will ultimately result in “one” formal data grammar or semantic schema. This is because the final implementation is supported by a Services Model to achieve the business deliverables of the enterprise. The decomposition of business activities is only a means to achieve the formalization of the semantic data model required to support the enterprise information.

It is the Services Model that will bring the necessary atomic processes, their necessary sequences along with pre and post conditions and the Control Sequence Model to enable the carrying out of the necessary services for the enterprise as derived from the requirements. The Services Model is essentially driven by an Event Model which in its simplest incarnation depicts Time, i.e. Run the Backup Services at midnight every night, perform certain services at that start of a new year etc.

### **3.2 Bringing the Processes together**

Recall that the elementary business activity or an atomic process, while it may have its own internal sequence to complete the stated elementary task (e.g. change reservation date of hotel guest), it should not be associated with another process in any sequence except if that process is calling another elementary process to complete its given elementary task. For example the atomic process “change reservation date of hotel guest” will require a re-usable atomic process “select hotel guest folder”

which will simply fetch the current reservation and other account details of the given hotel guest.

The sequence of processes to be performed is determined by a Services Model which has a set of processes that is driven by events and in turn uses a control sequence model that determines which process is to be performed for that particular service. Of course, the processes may require data from multiple database sources or URIs.

#### 4 Processes in a Services Oriented Architecture (SOA)

The onset of a Service Oriented Architecture paradigm has resulted in a mixed bag of successes and failures. This is essentially due to the lack of any formalistic approaches being adopted towards the assemblage of processes involved in a service. The more intrinsically woven the atomic processes are, the more inflexible and less adaptable to change the service becomes.

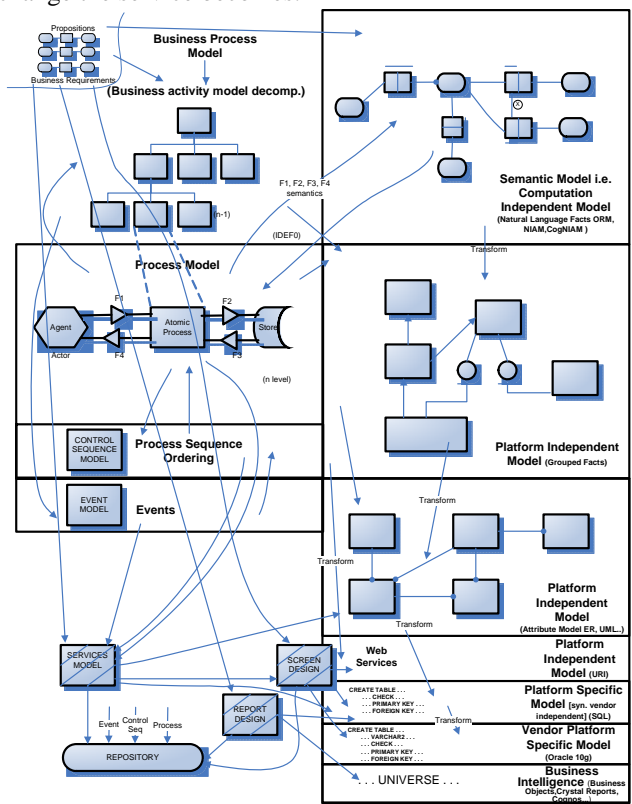


Fig. 3. Positioning the Services and Processes in the overall SDLC.

The secret here is to divorce the control and sequence from within the service process to the level of ensuring that the contained atomic processes are identified,



addition a Service must have one or more Service Providers which could be URIs or other agents including heterogeneous databases. A Service may be contained in a Service Group which may belong to a Service Category like Basic Services, Foundation Services, Management Services, Security Services, Business Services, and Identity Services etc. A Service Broker may access one or more Services Metadata.

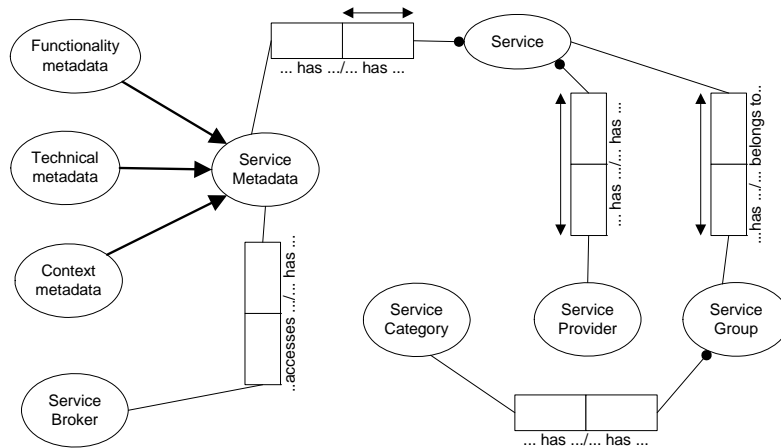


Fig. 5. ORM Schema of Common Services Metadata.

## 6 A Strong SOA Overlay based on Atomic Processes

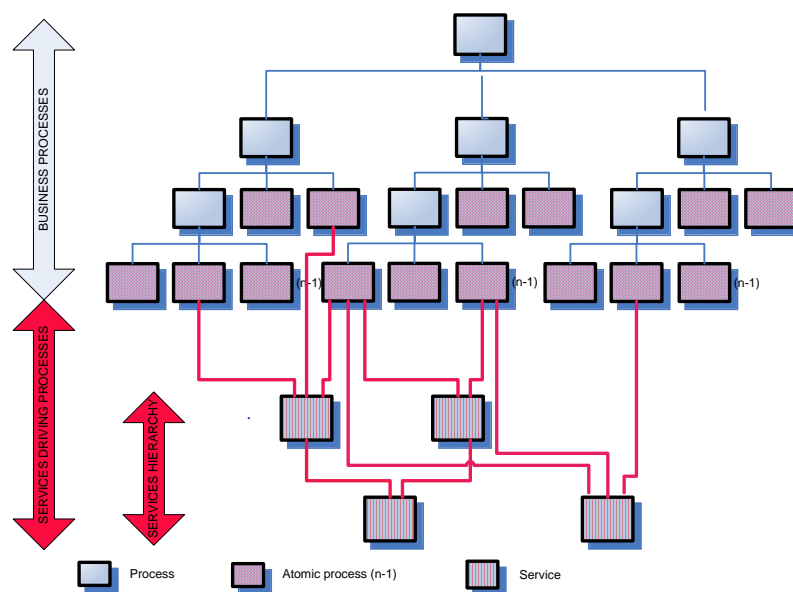
We certainly want to avoid a spaghetti Services Oriented Architecture resulting from an ad hoc process of bringing together many interconnected and interwoven application systems. Recognizing that while Business Process Modelling is essentially a top-down process, the Services Oriented Architecture is a bottom-up process consisting of an assemblage of constituent atomic processes and/or services. It is important to recognize that while a service is a commitment of the business to achieving an outcome, the process is a mechanism to deliver or achieve that outcome.

Fig.6 shows the Services that may be members of a Service Group, positioned at the bottom half, which may execute one or more re-usable Atomic Processes as per the Event Services defined in Fig.4 and positioned in Fig.3.

## 7 Summary

Process and Service modelling initiatives generally develop their models without much emphasis on data. Moreover, process modelling paradigms generally bury process sequence of operations as a thread within a composite process. More often than not, these buried operations are elementary atomic reusable components. The

resulting models are generally not flexible or sufficiently reusable, suffering from update anomalies and redundancies. Addressing “service” as a major deliverable component, an ORM metamodel was developed in line with ISO 19763-5 Metamodel for Interoperability: Metamodel for Process Model Registration, to harmonize atomic component processes using control sequence and event models to enable the delivery of a totally flexible model set facilitating metamodel interoperability and cooperation between systems via their respective models. The paper also provides a limited ORM based review of ISO 19763-5, and uses underlying component processes to develop a metamodel for a deliverable Services Oriented Architecture containing control sequence models, event models, and bridges to associated data models or web services.



**Fig. 6.** Overlay Positioning Services and Processes.

## Acknowledgement

The authors would like to acknowledge productive discussions with Dr. Robert Meersman, Dr. Sjr Nijssen, Paul Thompson, Dr. Yangfan He and Dr. Jian Wang.

This research project was supported by the National Basic Research Program of China (973) under Grant No.2006CB708302 and 2007CB310801, the National High Technology Research and Development Program of China (863) under Grant No.2006AA04Z156, the National Natural Science Foundation of China under Grant No.90604005, 60703018 and 60703009, and the Provincial Natural Science Foundation of Hubei Province under Grant No.2005ABA123, 2005ABA240 and 2006ABA228.

## References

1. International Organization for Standardization(ISO): ISO Technical Report TR9007: Information processing systems—Concepts and Terminology for Conceptual Schema and the Information Base(1987)
2. Object Management Group (OMG): Business Process Modeling Notation (BPMN) Specification, Final Adopted Specification(2006)
3. Chris Metcalf and Grace A. Lewis: Model Problems in Technologies for Interoperability: OWL Web Ontology Language for Services (OWL-S), CMU/SEI Technical Notes (CMU/SEI-2006-TN-018)(2006)
4. International Organization for Standardization (ISO): ISO/IEC 18629: Industrial automation systems and integration - Process Specification Languages(2004)
5. Tony Morgan:Business process modeling and ORM. In Proceedings of On the Move to Meaningful Internet Systems: OTM 2007 Workshops, Vilamoura, Algarve, Portugal. pp. 581-590, Springer-Verlag Berlin Heidelberg(2007)
6. International Organization for Standardization (ISO): ISO/IEC 19763-5: Information technology – Framework for metamodel interoperability –Part 5: Metamodel for Process Model Registration, Working Draft(2008).
7. Chong Wang, Keqing He: Extending Metamodel Framework for Interoperability (MFI) to Register Networked Process Models. Dynamics of Continuous Discrete and Impulsive Systems- Series B- Applications & Algorithms, vol. 14(S6), pp. 72--78. Watam Press, Waterloo(2007)
8. International Organization for Standardization (ISO): 12207: Information Technology-Software Lifecycle Processes(1995).
9. Terry Halpin: UML data models from an ORM perspective: Parts 1-10. Journal of Conceptual Modeling, Inconcept(1998-2001), <http://www.orm.net>.