

EAML: A MOF-Based, Common Enterprise Architecture Meta-model

John C. Butler, Ravi Hubbly, Manfred R. Koethe,
and Dr. Walcelio L. Melo

Executive Summary

The General Services Administration (GSA), Office of the Chief Information Officer (OCIO), pursues new ways of applying computing and communications technologies to the practical problems of information management. The goal is to reduce the cost and improve the quality of government services, reduce technology risk, and share the results of projects throughout the Federal sector.

In order to assist GSA OCIO to attain its goals, we have created the core of a MOF-based common enterprise architecture modeling language (EAML) and the kernel of an environment to store, retrieve, and handle EA models described in this language. To do so, the most common EA models used in this organization have been studied. Then, a common language able to express these EA models and the relationships among them have been created. The resulting language is heavily based on OMG EDOC standard. However, the current version of OMG EDOC required enhancement in two fundamental ways. First, the EDOC metamodel issues have been corrected. Second, new modeling elements have been added in order to make OMG EDOC rich enough to support the EA models currently used at GSA. This paper aims at describing the main characteristics of EAML meta-modeling concepts.

1. An Overview of EDOC

As commented in the previous section, the current version of EAML is heavily based on EDOC. This section presents a short overview of EDOC. Readers familiar with EDOC can skip this section. Those interested in having a more comprehensible view of EDOC should refer to the OMG EDOC specification[2].

As stated in the OMG EDOC specification, EDOC aims at simplifying the development of component based systems by means of a modeling framework that conforms to the OMG Model Driven Architecture. Regarding the use of EDOC as a core for our MOF-based common meta-model for enterprise architecture, EDOC provides the following modeling capabilities:

- ❑ A platform independent, recursive modeling approach that can be used to describe EA models at different levels of granularity and different degrees of coupling.
- ❑ Out-of-the-box modeling concepts for describing the business, system, and technological aspects of EA models.
- ❑ Traceability among different kinds of models.
- ❑ Extensibility allowing EDOC to be enhanced to support different kinds of EA models

EDOC is defined as a structured set of MOF metamodels. Only selected parts out of this set have been used for the work described in this document; these are:

- ❑ The Component Collaboration Architecture (CCA), which provides the core component and collaboration modeling capabilities at varying and mixed levels of granularity.
- ❑ The Entities model, which allows the modeling of EA concepts in the form of entity relationships, where the entities are represented by CCA components. The original EDOC Entity model was considerably changed in order to provide equal support for EA data models on conceptual and instance level.
- ❑ The Events model, which allows the modeling of elements representing business events. They may be used on their own, or in combination with other EDOC elements to model event driven systems.
- ❑ The Business Process model, which extends the CCA with modeling elements for the expression of intra- and inter-component semantic in form of workflow-like processes.

2. Changes to the EDOC Metamodel

A collection of changes were applied in the context of this project to the EDOC specification, as finalized by the OMG. These changes were necessary to correct syntactical errors found in the specification; to make the metamodel less sensitive to imperfections of existing metamodeling tools; and finally some modest functional enhancements were introduced. All changes will be reported back to the OMG.

The majority of applied syntactical corrections fell into categories like:

- ❑ Missing or misplaced Association Role names
- ❑ Missing or incorrect Multiplicities
- ❑ Missing or incorrect Association Navigability
- ❑ Misspelled Names, missing Type Declarations

Figure 1 show a sample of the syntactical changes done in EDOC. Figure 1.a shows EDOC before the changes and Figure 1.b after the changes.

a) Syntactical Changes (before)	b) Syntactical Changes (after)
---------------------------------	--------------------------------

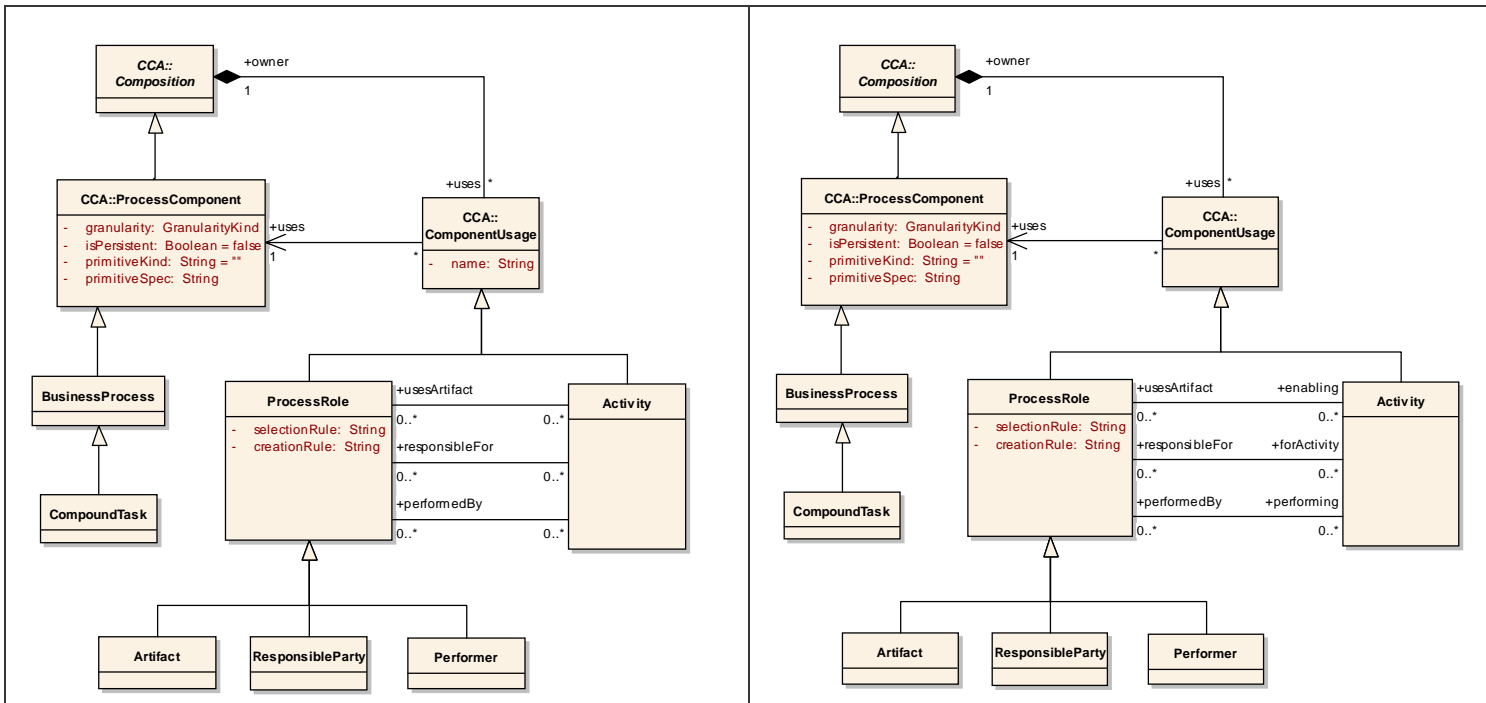


Figure 1: Syntactical changes made in EDOC

Also, several changes were required to make it possible to handle EDOC with the meta-tools used in this project, e.g., Eclipse/EMF. Basically, the following changes have been made for this purpose:

- ❑ Changes to Association Role names to avoid keywords
- ❑ Changes to Association Role names to avoid double definitions
- ❑ Additional Stereotypes to “teach” the tools how to handle Multiple Inheritance

Finally, semantic changes have been made in order to enhance EDOC functionality. In the following it is described what changes have been incorporated in EDOC:

- ❑ Model Structure: The whole model is now rooted.
 - ModelElement: is the root of all EDOC model elements
 - NamedElement: is the root of all EDOC model elements that have a name.
- ❑ The new structure supports Multiple Inheritance
- ❑ Redesigned Data Type Model
- ❑ Redesigned Entity-Relationship Model
- ❑ Addition of an Organizational Structure Model

Figure 2 shows the new EDOC model root. The common root is ModelElement. A new class AspectableElement specializes ModelElement by providing identity and a selector mechanism to provide aspect-oriented instantiation mechanisms. All model elements, except for a few aspect control classes, are now derived from AspectableElement. Model elements which previously carried individual “name” attributes are now derived from NamedElement, which specializes AspectableElement by introducing a unified naming concept.

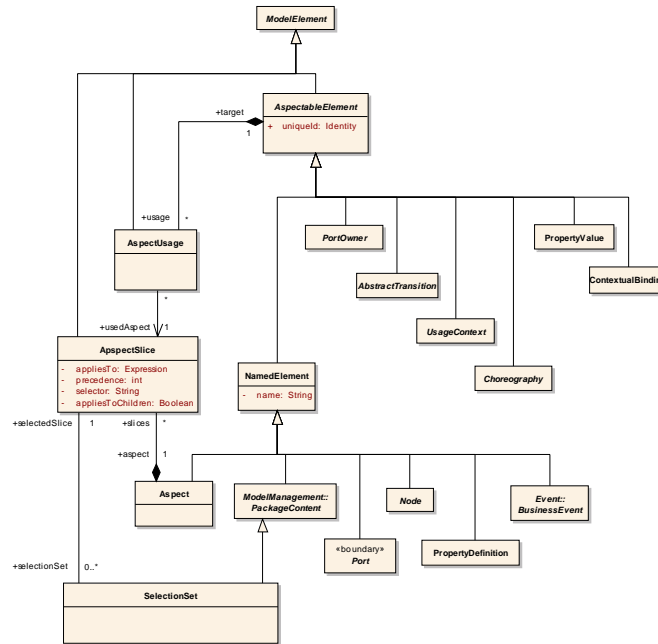


Figure 2: EDOC root element

Figure 3 shows a change within EDOC CCA to support multiple inheritance for CCA components. Figure 3.a reflects the original, which supported only simple inheritance.

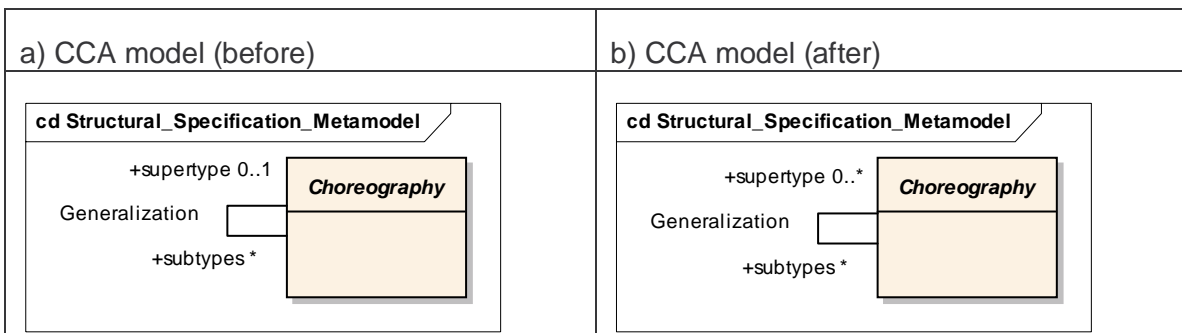


Figure 3: Multiple inheritance in EDOC.

The Document Model provides data type definitions for the use in CCA component collaborations. It turned out that the originally available data type definitions were too restrictive. Also, the reworked Entity Model imposed new requirements on the Document Model.

Even though the two diagrams, “before” and “after” look different, the changes are actually minimal:

- ❑ All data types support now multiple inheritance
- ❑ Attribute and Reference (see Entity Model) are now derived from StructuralFeature, which replaced the previous Attribute
- ❑ StructuralFeature is now named and allows redefinition

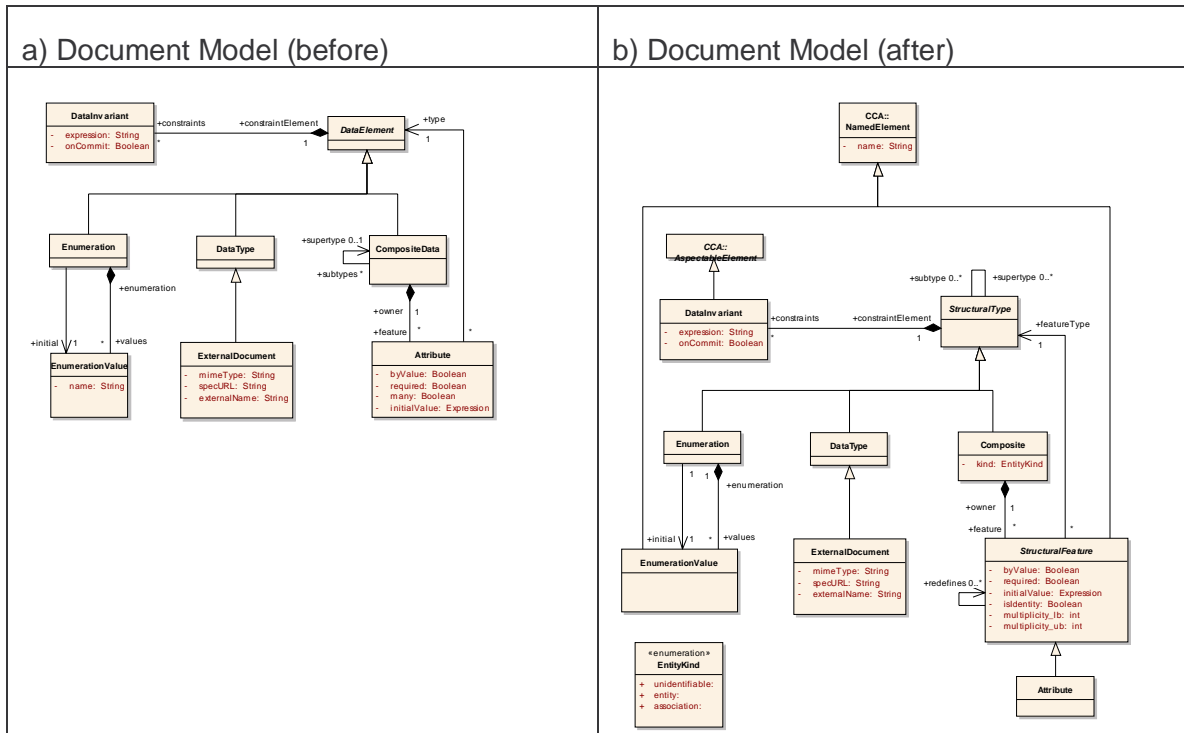


Figure 4: Data model sub-system

2.1. EDOC Entity Model

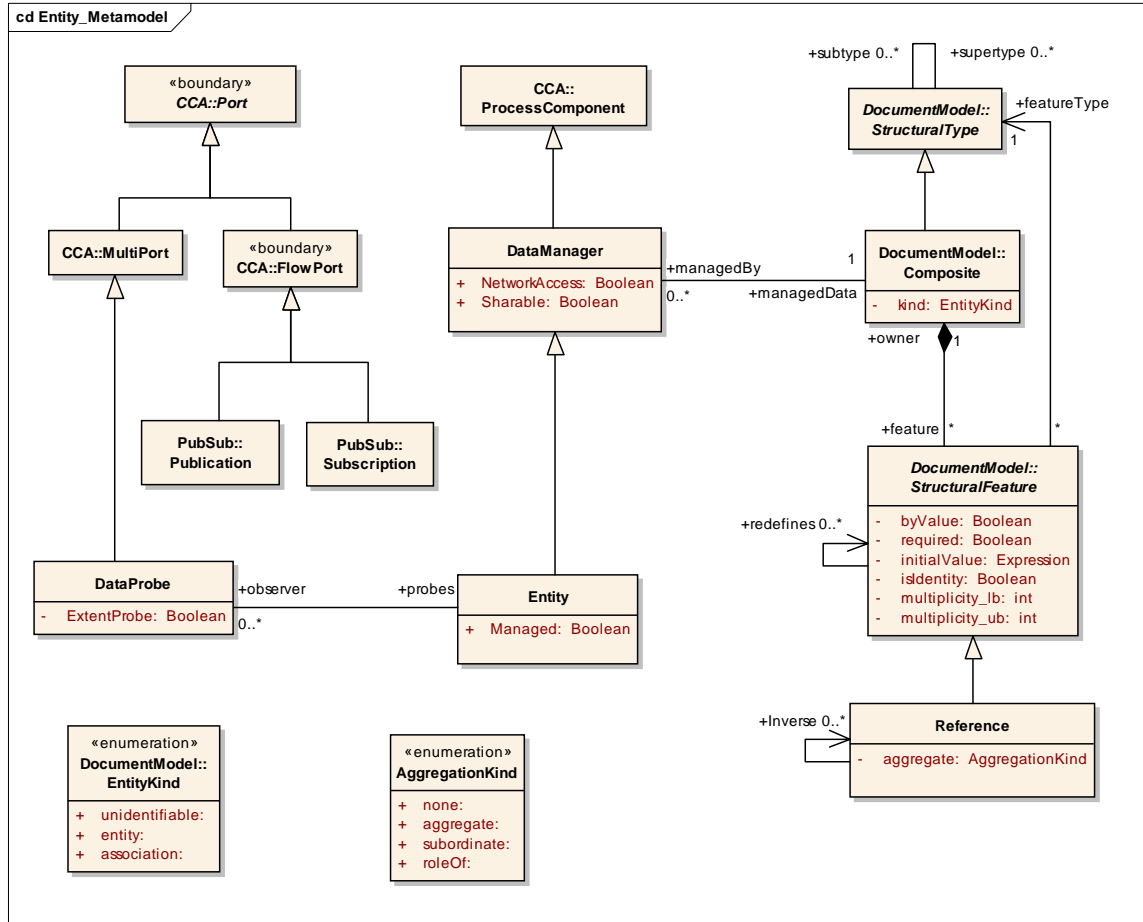


Figure 5: Enhanced version of EDOC Entity Model

In order to support Business Data Structures and Business Information Needs EA models¹, EAML required that EDOC Entity Model be significantly enhanced. Figure 5 presents the enhanced version of the EDOC Entity Model. This enhanced version was successfully used to capture the Business Data Structure EA models.. The changes applied to the EDOC Entity Model are:

- ❑ CompositeData was renamed to Composite and carries now an attribute of type EntityKind. This acts as a type modifier to characterize the Composite instance as entity, association, or as an unidentifiable data element. This way Composite can represent a relationship of an entity-relationship model as a first class modeling concept. Composite is a concrete specialization of StructuralType defined in the Document Model of ECA.
- ❑ A Composite can have many features, which are either Attributes or References. StructureFeature, the abstract base class for Attribute and Reference carries

¹ A Business Information Need Model is an EA model Enterprise Artifact that defines the specific set of knowledge (business intelligence) that a particular role or Organizational Unit requires for making decisions and/or completing a unit of work.

constraining flags and the multiplicity. It supports redefinition through the redefines association.

- Reference, a concrete sub-class of StructuralFeature, captures the type of reference that binds two or more entities. Via the aggregate attribute the relationship can be further detailed into an aggregate, subordinate, or simply a role.

2.2. EDOC Process Model

The EDOC Process Model was used *as-is* by EAML. Figure 6 shows the EDOC modeling components we have used in this work to describe EA models. The description of these model components can be found in the OMG EDOC specification[2].

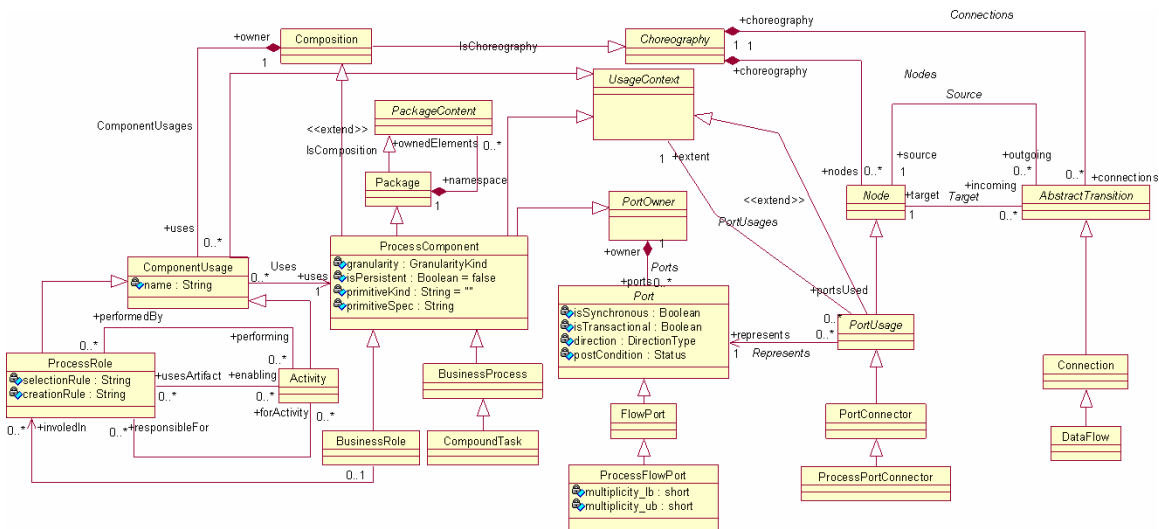


Figure 6: EDOC modeling elements for business process model

2.3. EDOC Organizational Structure Model

In order to support Organization Structure EA models, EAML required that a new sub-model be added to EDOC. Figure 7 shows how EDOC was enhanced to model organizational structures. Each **OrganizationalUnit** is represented as a CCA component and reified by one or multiple **BusinessEntity** components. **BusinessEntity** is a specialization of **Entity** (from the Entity Model). As specializations of CCA components, **OrganizationalUnit** and **BusinessEntity** benefit from the component collaboration and recursive refinement mechanisms. Any organizational topology can be modeled.

In most organizations are the externally visible interfaces of an organizational unit defined by the roles this unit plays within the organization, and in particular within the business processes executed by the organization. These facts are expressed by the **BusinessRole** element, a CCA component owned by an **OrganizationalUnit**. Each **BusinessRole** can optionally express the participation in one or many business processes through a link to the corresponding **ProcessRole**. The business processes are not directly affected by this connection.

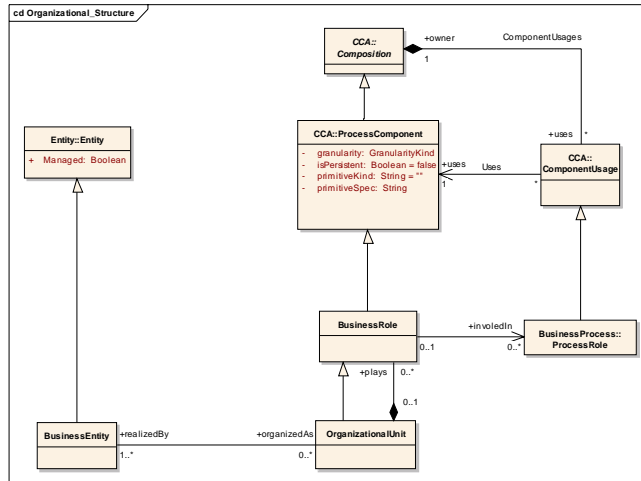


Figure 7: Organizational Structure Model

3. A Case Study

In order to validate EAML modeling capabilities, we have migrated EA models created by well-known EA tools to EAML modeling concepts. Given the fact that at GSA Popkins System Architect is one the most used EA tool, it was natural to map Popkins SA EA models to EAML.

Figure 8 shows the mapping between SA Entity-Relationship (ER), which is used to capture Business Data Structures, and EAML. Entities and relationships are first classes citizen and they are represented by the CompositeData model element. This allows n-tiers relationship to be easily captured in the model. For each CompositeData an Entity model element is also created in order to capture conceptual entity-relationship models. Further information about EDOC model elements can be found in the OMG EDOC specification.

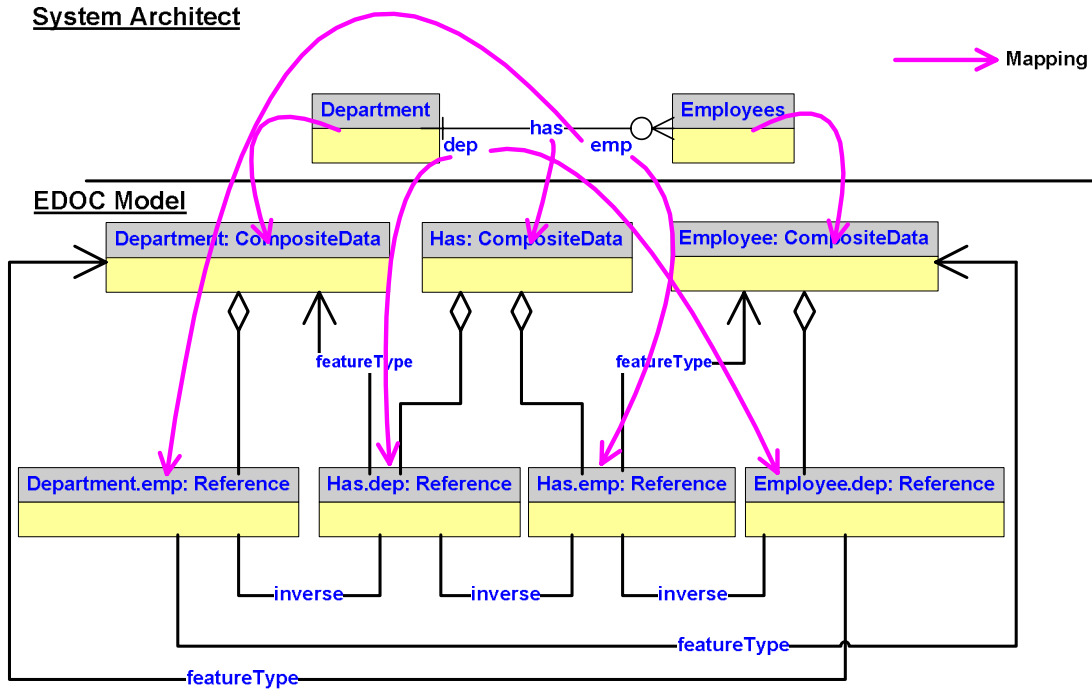


Figure 8: EAML mapping for Business Data Structures EA Models (SA Entity-Relationship Models)

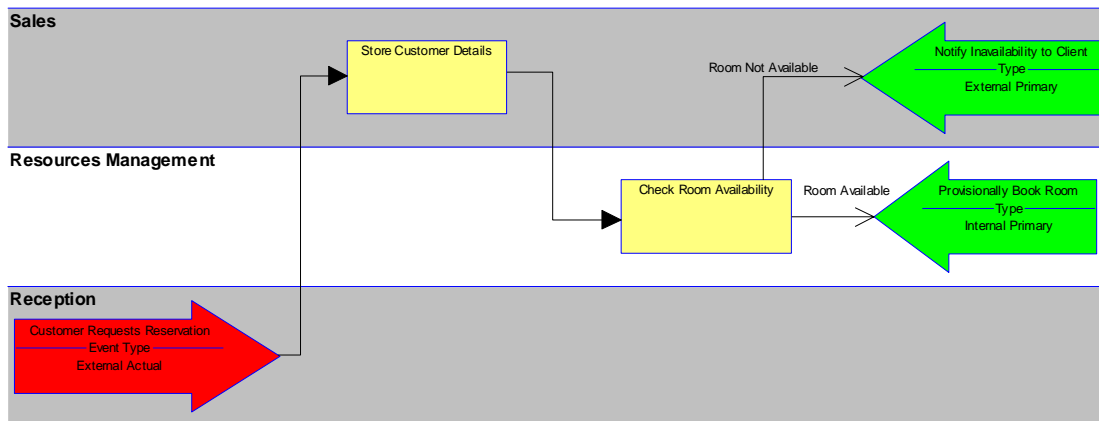


Figure 9: A SA Business Process Model sample

Figure 9 shows a sample of a Business Process Model built in SA. In this diagram, the arrow-pointing-right symbol represents an event, **Customer Request Reservation**. There is a mandatory flow (bold line) from this event to the process **Store Customer Details**. There is a mandatory flow (bold line) from process **Store Customer Details** to process **Check Room Availability**. There are two optional flow lines leaving the process **Check Room Availability** – thin lines with an not filled arrow on the line. These lines

denote a flow out of the process dependent on a condition. If Room Available then Result called **Provisionally Book Room** is done else Result Notify **unavailability to Client** is sent.

In addition, the event and processes in this diagram are plotted against the part of the organization where the event happens or the process takes place – these divisions, shown in the diagram as **Reception, Customer, and Accounts** – are known as "swim lanes," and represent units in the organization – or *Organizational Units*.

Hence a SA Business Process Model is made up of the following elements:

- ❑ Swimlanes (Organizational Unit)
- ❑ Events
- ❑ Results
- ❑ Elementary Business Process
- ❑ Mandatory Flow
- ❑ Optional Flow

The mappings of these model elements to EAML model elements were identified as shown in Figure 10

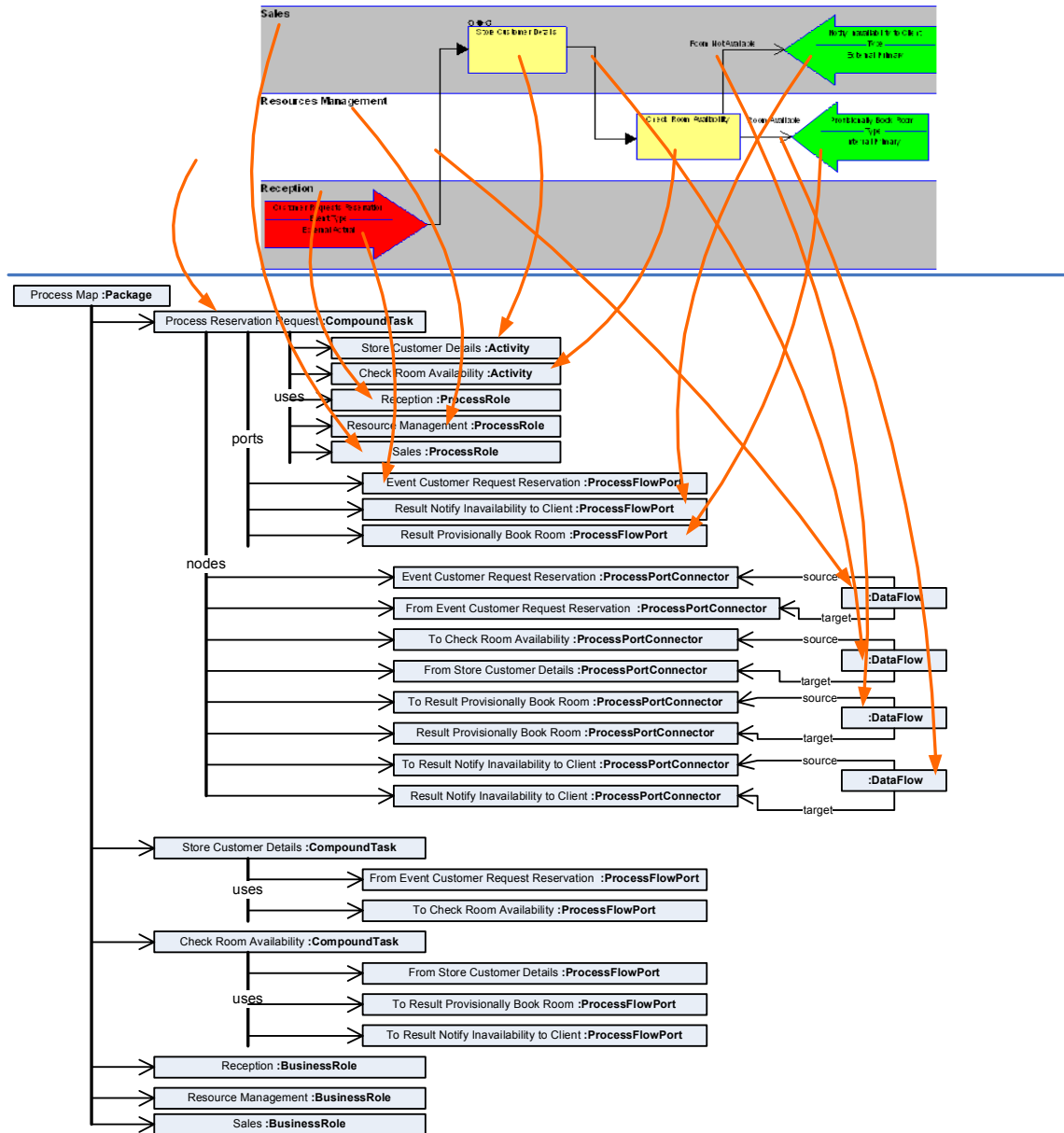


Figure 10: A SA Business Process Model mapped to EAML

4. Conclusion

One of the key aspects of capturing any architecture is the creation of models that graphically represent information about the system under study, whether that system is an information system, an organization, or an entire enterprise. While the system development world has coalesced around the use of UML as a modeling language, the EA industry currently has no standard modeling language akin to UML. Some use swimlane diagrams in conjunction business process decomposition diagrams. Others use strict process decomposition diagrams without swimlanes and still others use UML Activity diagrams. The result is a fragmented community wherein it is difficult to share models and thereby promote a common understanding and reuse.

In this paper, we have presented the core of an enterprise architecture meta-modeling language that addresses this issue. Instead of re-inventing the wheel, the model integration was accomplished by using a refined version of OMG EDOC specification. In order to validate the capabilities of EAML, we have migrated a few EA models built in Popkins System Architect and DAT ComponentX to EAML.

Based on our experience, we have verified that EDOC lacks concepts required for fully capturing enterprise architecture such as Vision, Strategy, and Plan. Without these concepts the EA cannot be tied back to the strategic or capital planning, organizational goals, and/or federal enterprise architecture reference models.

We will continue this work by enhancing Enterprise Architecture Modeling Language (EAML). The new versions of EAML will take advantage of other OMG languages as they emerge, e.g., BSBR and BPDM.

References

- [1] David S. Frankel, Paul Harmon, Jishnu Mukerji, James Odell, Martin Owen, Pete Rivitt, Mike Rosen, Richard Mark Soley. The Zachman Framework and the OMG's Model Driven Architecture. Business and Process Trends, White Paper, September 2003.
- [2] Object Management Group (OMG). Enterprise Collaboration Architecture (ECA) Specification. February 2004, Version 1.0, formal/04-02-01.
- [3] J.A.Zachman. "A Framework for Information Systems Architecture," IBM Systems Journal, Vol. 26, No. 3, 1987. (The same article was reprinted in 1999 in a special double issue of the IBM Systems Journal that is easier to locate: Vol. 38, Nos 2&3, 1999.)
- [4] Information on Zachman's current work can be obtained from The Zachman Institute for Framework Advancement (ZIFA) www.zifa.com
- [5] Zachman has recently prepared an electronic book, The Zachman Framework: A Primer for Enterprise Engineering