

Enterprise Distributed Object Computing

EDOC

A new standard from the object management group (OMG) promises to address enterprise needs with UML

The need to collaborate and integrate has become of strategic importance to the modern enterprise. Supply chain integration, customer management, B2B and enterprise application integration are all manifestations of this need to improve and automate collaboration using enterprise information systems. The EDOC specification from the OMG provides a standard way to model business collaborations with UML and “drill down” to executable systems using middleware such as web services, ebXML, J2EE, MQ-Series, NET and, of course, Corba. This is the first of a multi-part series on EDOC, this month we will see what it is and how it fits into the enterprise applications landscape.

Part of the Model Driven Architecture Vision

EDOC is part of the OMG’s vision of a “Model Driven Architecture”, or MDA for short. In the MDA vision the application lifecycle is driven from high-level, Business focused UML models that can be “mapped” to various technologies for implementation and execution. Those who have used UML know that generation out of UML models has been difficult and usually produces code that is “to generic” for practical use. For this reason most UML models are merely guides to a subjective and manual process of implementing the application. MDA addresses these problems in two key areas;

UML Models intended for MDA are produced according to some “Profile”, a very specific way to use UML for a particular purpose. There could be profiles for real-time, for data modeling or for enterprise computing. EDOC is the standard UML profile for distributed enterprise computing, the way people, departments and companies integrate and collaborate. As such, EDOC becomes the modeling framework used at the enterprise level. Profiles make UML sufficiently structured and precise to use for MDA and to drive later stages in the development life cycle. These precise high-level models are known as “Platform Independent Models” (or PIMs) in the parlance of MDA.

The other half of the MDA vision is “mapping”, mappings specify how to go from a platform independent model to a specific technology, such as J2EE or web services (These are called “Platform Specific Models, or PSMs, in MDA). This mapping takes the platform independent model, combines some detail required for that mapping, and produces the artifacts required to implement that model in the target technology. Since MDA models usually don’t have all the information necessary to generate a complete system, programmers still need to “fill in the blanks” by implementing specific methods and supplying specific detail in the generated system. Various approaches to MDA generate more or less of the system.

We all know that technologies are transient and our business systems invariably outlive the infrastructure of the day, MDA gives us a way to design and build our systems in a way that is less tied to a particular infrastructure or vendor – making the systems more flexible and longer lived. It also provides the opportunity to automate much of the development process, generating code and other artifacts from the model in a reliable and consistent way. Since both the profile and the mapping are tuned to the problem at hand, the “overly generic” problems of UML code generation are largely mitigated. So we have the combined advantages of a longer-lived and more flexible system leveraging fast, automated development.

Automated Business Collaboration

Collaboration is the essence of business today. Streamlining your business processes means that the parts of your company, your suppliers and your customers are all working together, improving service and reducing costs. Improving collaboration means that both your people and your systems must work together effectively.

The Internet has been great at letting people access information and services through web browsers, but computer systems can't use a "web browser" to interact with other systems – they need something more structured. The capability for systems to interact on the internet is still leading edge and has inspired a whole new set of technologies for business-to-business and application to application integration. Some of these new technologies are; web services, XML, J2EE and Corba. But, the way to understand and design these automated business collaborations has been missing, and each technology vendor has provided different kinds of tools to get the job done – making it very difficult to deal with a wide range of technologies or products. This is where EDOC comes in.

A Top-down view of EDOC

The highest-level concept in EDOC is the "Community Process", a set of "roles" that interact in a particular way for some business purpose. For example a community process could be defined for ordering parts or filling a prescription. When fully complete, the community process specifies all the information that flows between the parties, when and under what conditions. A "Process Component" that plays that role in the information system then represents each of these roles. A role is just a way to talk about some business function, like an order taker, expense approver or broker. Everything in EDOC centers on this central concept of roles interacting for a business purpose.

Roles interact in well-defined ways as defined by a *protocol*, a specification of the data that flows between parties. Part of this protocol specification is the business documents that flow between each role and a choreography of when the documents may be sent. Protocols may be reused in multiple collaborations. In the community process you don't see the protocol detail, this detail is a "drill down" from the high-level model. This concept of drill-down to detail is prevalent in EDOC.

Protocols may be as simple as a single data flow or event or as complex as a long-running and diverse conversation between business partners.

Once we have defined our roles, protocols interactions we can "drill down" into that role. Inside of a role you may find more roles – more "parts" of your enterprise interacting, you may find a connection to your DBMS system, workflow or messaging infrastructure. This ability to drill-down into your enterprise architecture provides a way to understand your system at successive levels of detail, maintaining a high-level view while linking to implementation detail.

EDOC defines sub-profiles for how each role interacts with and can become a part of your enterprise system using events, DBMS systems and business processes detail. The EDOC patterns profile allows generic patterns to be developed and used to build enterprise models. A sister RFP from the OMG also shown how your legacy systems integrate using Enterprise Application Integration (EAI).

The final part of EDOC is the mappings to particular technologies. While some technology models are included for J2EE and Messaging, the mappings have not yet been standardized; this is work still to be done. There is also collaborative work going on in the Java Community Process, JSR-159 will specify how process components can implement process roles using Java and J2EE. With this in place, one EDOC model will be able to support business collaborations using any of these technologies and will produce the technology specific form of the collaboration, such as a WSDL (Web Services Definition Language) specification or Corba-IDL file. [Note that this can be done today, but the mappings are not yet standard.]

An example EDOC community process

A typical example of an edoc process involves a “Buyer” and “Seller” role. The buyer and seller have “ports” (“buy” and “sell” are ports) , each port is connected to another role with a complementary interface. The information that flows across the ports is defined by a protocol, which specifies both the structure of the information that flows and “Choreography”,. Or ordering of that information flow. Once of these roles could be implemented by a party inside or outside the enterprise and could use a variety of technologies for interaction.

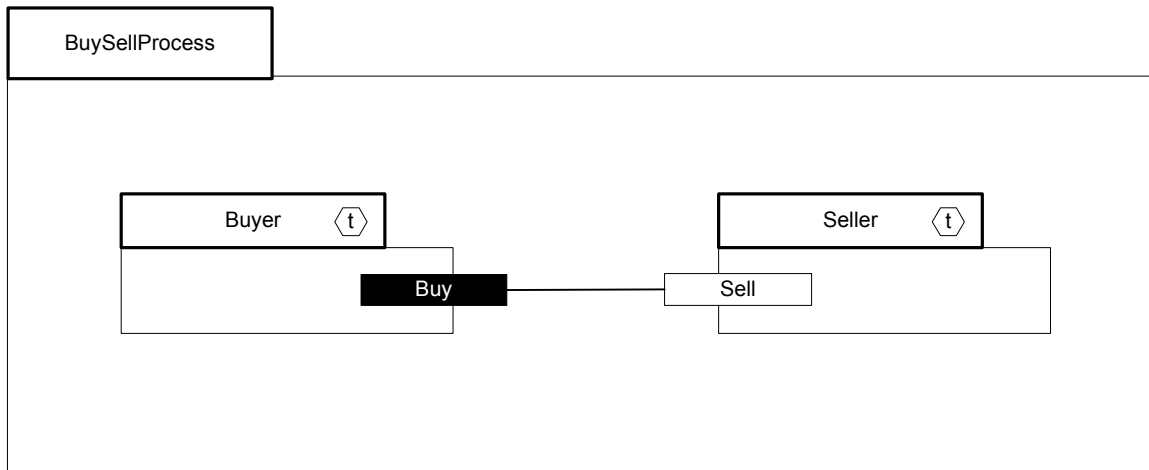


Figure One

Community Process

The choreography of a protocol is shown using a UML activity diagram with each information flow being one activity.

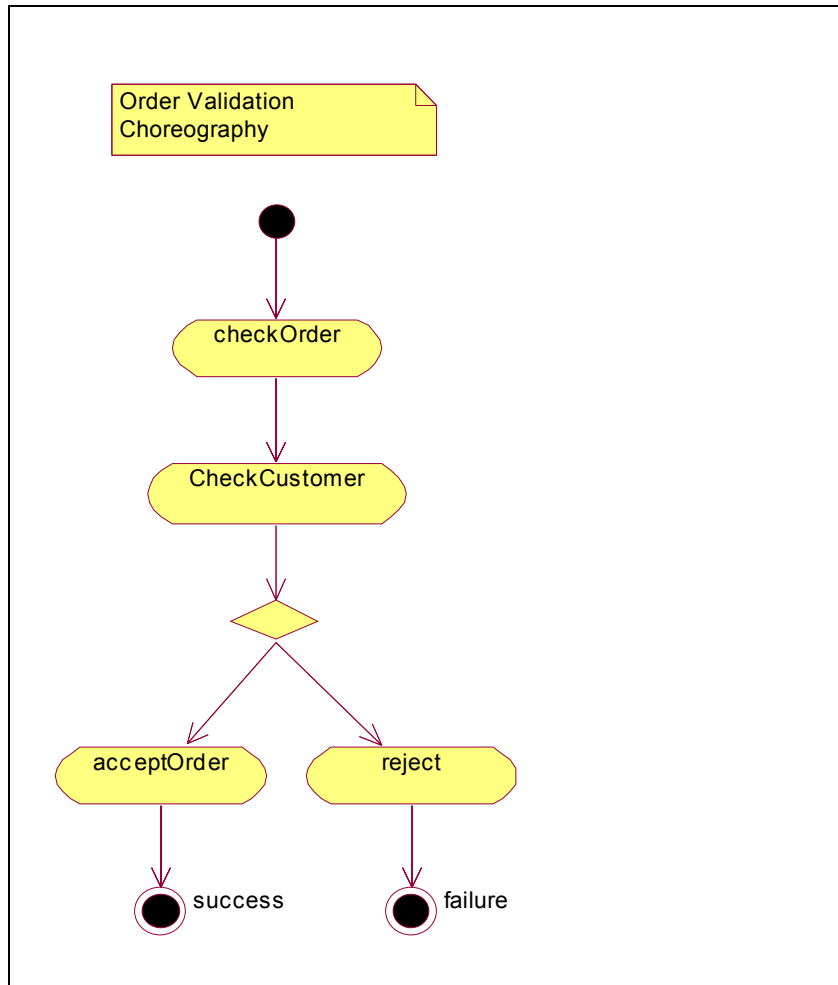


Figure Two
Choreography of protocol

In summary

EDOC has the capacity to address fundamental issues facing developers today – how to understand distributed business processes, how to implement them effectively on a wide range of technologies and how to improve the development process. It is part of a broad vision that could have a profound impact on the way software is developed, managed and deployed.

Read more about it

EDOC – www.omg.org/edoc

MDA – www.omg.org/mda

WSDL - <http://www.w3.org/TR/wsd1>

JSR-159 - <http://www.jcp.org/jsr/detail/159.jsp>

In the next installment

In the next installment we will look at EDOC processes in more detail, focusing on the core of the specification.

About the author

Cory Casanave is president of Data Access Technologies (www.enterprise-component.com), a company specializing in tools, training and services for enterprise collaboration and automated development. Mr. Casanave has 25 Years experience in the architecture, tooling and development of business software and was one of the contributing authors of the EDOC specification. He can be reached at cory-c@enterprise-component.com.