

Intersection of Business Process and Software Development Process: the Middle Tools

Thanadee Ujjubandh,
MagicDraw RConverter Product Manager
No Magic Asia Ltd
Bangkok, Thailand
Email: thanadee_u@nomagicasia.com

Abstract

Only one third of the software development project can complete within budget, on time and meet the customer requirement. One of the top ten recipes to make project success is a clear business objective.

Business people use business model to describe their business and use various tools to build the business model. For developers, they do not understand business process; but UML diagrams and programming languages. This two group of people need to meet halfway. This research proposes the middle tools to deliver information from business process tool to UML modeling tool.

1. Background

Only 28% of software development can finish within budget and meet the customer requirement, 23% fail and cancel the project before completion and 49% finish but over budget or spend more time or not meet all customer requirements.

Executive Support	18
User Involvement	16
Experienced Project Manager	14
Clear Business Objectives	12
Minimized Scope	10
Standard Software Infrastructure	8

Firm Basic Requirements	6
Formal Methodology	6
Reliable Estimates	5
Other	5

The table above shows the recipe for the project success [1]. For business people, they use a lot of business modeling and modeling tools to clarify the business objective. Most technical people do not understand the business process, nor the business-modeling notation. They have not used the business-modeling tool before; all they know is how to build applications. UML is the standard model for system analysis and design [7]. Strange but true, this group of people having totally different background collaborates to develop software.

Some solve this problem by combining business models and UML together and they call it Business Modeling with UML [3]. In this method, the business people and the technical people need to agree to learn the new model and need to change their modeling tool to support this new model. So conversion cost and learning curve are the dominant drawbacks in this method.

This paper proposes another method to reduce the gap between business and technical in a different way by setting users standard formats to exchange data, integrate the export module to business process modeling tools and integrate the import module to the UML modeling tool. In this

Outputs. By linking them to Inputs and Outputs Business Objects provide a detailed view of how information is transformed through Activities. Business Objects are the vehicles by which data flows through an organization and exist in a variety of formats - paper forms, electronic forms, or database tables.)	
“Data Object” (A Data Object represents a group of functionally related Business Object Fields that are used in one or more Business Objects. For example, a Customer Information Data Object is comprised of six Business Object Fields: Customer Name, Address 1, Address 2, City, State, and Zip Code.)	“Class” with stereotype <<dataObject>>
“Business Object Field” (A Business Object Field represents a single field of information that may be associated with one or more Data Objects. In turn each Data Object is associated with one of more Business Objects. The entire list of Business Object Fields may be thought of as representing a business data dictionary of the selected business scope.)	“Attribute” with stereotype <<businessObjectField>>
“User Defined Resource” (All Activities are completed with the assistance of at least one Resource. Resources are defined as any person, system, or department that	“Class” with Stereotype <<userDefinedResource>>

assist in transforming Inputs into Outputs.)	
“User Defined Resource Type” (All Resources belong to a Resource Type. Resource Types (and related Resources) should be created only when default Resources types (Business Units and Roles), are insufficient.	“Class” with Stereotype <<userDefinedResourceType>>
“Business Object” that contain “Data Object”	“Aggregation” link that shows “Data Object” is the part of “Business Object”

2.2.3 For Activity diagram

ProActivity	MagicDraw
“Process”	Create a separate activity for every separate process. With Stereotype <<process>>. Activity should be placed in Process Family package.
“Sub-Process”	Activity in process activity. Nested sub-processes will be mapped to inner activities with Stereotype <<sub-process>>.
Activity	Call Behavior Action Control flow should be drawn from initial node to the first

	activity. Control flow should be drawn from the last activity to final node with Stereotype <<paActivity>>.
Activity with property "ActivityCondition" - Yes	Decision Node with Stereotype <<paActivity>>.
Text on outgoing relationships from Activity with Condition.	Guard property of Control Flow relationships
"Input/Output" ((Inputs represent information necessary for one or more Activities to begin. Inputs come in many forms: e-mail, fax, phone call, or electronic file. Outputs represent information created as a natural byproduct of a completed Activity.)	Input/Output pins connected with object flow relationship with Stereotype <<input/output>> .
Activity contains 2 outgoing relationships to another activities.	Vertical synchronization
"XOR"	Merge Node
"Role"	Swimlane

After we get the overall picture of mapping rule then we will create the ProActivity Profile - A UML Profile is used to define a specific set of extensions to the base UML in order to represent a particular domain of interest. Then create the export module in ProActivity Client to export xml data from ProActivity to XML format. Next step is to create a new import module inside MagicDraw to read the data from the ProActivity Exported file in this case the Java API for XML Processing API (JAXP-DOM) is used for accessing the information in XML Exported file and finally convert it

to Class, Use Case and Activity diagram in MagicDraw.

3. Conclusion

At present, business users and software developers talk with different languages. This problem can be solved in the level of modeling tool by creating additional import and export module on each side. If the data that use for import/export become standards for each vendor, the user will change to use any tools for exchange data between business models and UML. For future study we should think about the bidirectional method to exchange information between business modeling and UML or the most effective way to combine business model, UML and MDA [6] together.

References

- [1] "Extreme Chaos", The Standish Group International, 2001.
http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf.
- [2] Vytis Ciemolonskis and Paule Zaksauskiene, "MagicDraw-ProActivity Integration Requirement", No Magic, Inc. European Software Development Center
- [3] Hans-Erik Eriksson and Magnus Penker (2000), *Business Modeling with UML Business Patterns at work*, OMG Press.
- [4] MagicDraw is the product of No Magic Inc., "MagicDraw UML",
<http://www.magicdraw.com/>.
- [5] ProActivity is the product of ProActivity Inc., "ProActivity",
<http://www.proactivityinc.com/>.
- [6] "OMG Model Driven Architecture",
<http://www.omg.org/mda/>
- [7] "UML", <http://www.uml.org/>