

## Eclipse BPMN Modeler – Introducing Intalio|Designer

---

<b>Arnaud Blandin</b>	<b>Ismael Ghalimi</b>	<b>Hugues Malphettes</b>
Intalio Inc, EMEA Manager	Intalio Inc, CEO	Intalio Inc, Lead Developer
6 rue du conseil general 1205 Geneva Switzerland	1000 Bridge Parkway, Suite 210 Redwood City, 94065, CA USA	1000 Bridge Parkway, Suite 210 Redwood City, 94065, CA USA
<a href="mailto:blandin@intalio.com">blandin@intalio.com</a>	<a href="mailto:ghalimi@intalio.com">ghalimi@intalio.com</a>	<a href="mailto:hmalphettes@intalio.com">hmalphettes@intalio.com</a>

### **Abstract:**

Eclipse is becoming the standard platform for developing applications in a non-Microsoft based environment. Applications are now to be seen beyond the simple scope of software application. Indeed business processes can now be designed and implemented in a graphical tool. Intalio has leveraged the various Eclipse projects and the Eclipse 'plugin' architecture to build a complete BPMS Designer. One core component of its BPMS Designer that allows business users to design business processes has recently been donated by Intalio to the Eclipse project: the BPMN Modeler.

The aim of this paper is to present the benefits of using Eclipse to design business processes as well as providing an introduction the BPMN Modeler project.

## Eclipse: the BPMS IDE

---

### One Single Tool for Designing Processes

Not long ago, you needed up to seven tools for going from a process map to deployed code: one for modeling the process at a business level, an other to describe technical details, a third to build connectors to external systems, a fourth to map data in and out, a fifth to specify business rules, a sixth to design workflow user interfaces and a seventh to deploy all the code on a collection of proprietary runtimes components. They all required different kinds of expertise, ran in different environments, used different languages, lost information when going from one to the other, and made the whole thing so complex than no end-user could actually use them without the outrageously expensive consulting services of a software vendor that got all the pieces through multiple acquisitions rather than building them from scratch with a very clear architecture in mind.

Modern BPMS [BPMS] enforcing the BPM 2.0 [BPM20] principles puts everything you need within one tool, and that tool sits on top of Eclipse. With Oracle and SAP now supporting Eclipse, no other integrated development environment — beside Microsoft's Visual Studio — will matter anymore.

### Eclipse and BPMS

Integrated Development Environments (IDEs) are very much like Operating System, in the sense that they provide the foundation upon which software vendors can build their tools, without having to re-invent the wheel all the time. And much like Operating Systems, they tend to be the victim of commoditization and consolidation, rather sooner than later. Oracle became one of Eclipse's strongest supporters, Borland has decided to sell its JBuilder group, and NetBeans is not exactly driving developers to Sun's platform in droves. This leaves two contenders on the ring: Eclipse and Microsoft Visual Studio. But because Microsoft does not like to leave much room to its competitors, the vast majority of BPMS vendors have adopted Java as a development platform, and this is where Eclipse comes into play. In other words, if you're Microsoft, your BPMS will be built around Visual Studio, but if you're not, Eclipse is the way to go.

The main benefit of using a standard IDE for developing the tools that are part of a BPMS offering is that you can leverage existing components that have been developed by third-party vendors. For example, Intalio is using the rule editor from JBoss Rules, a KPI designer from Celequest, a WYSIWYG XForms editor and a set of tools to generate web services developed by the Axis2 team; all of which are available as Eclipse plugins.

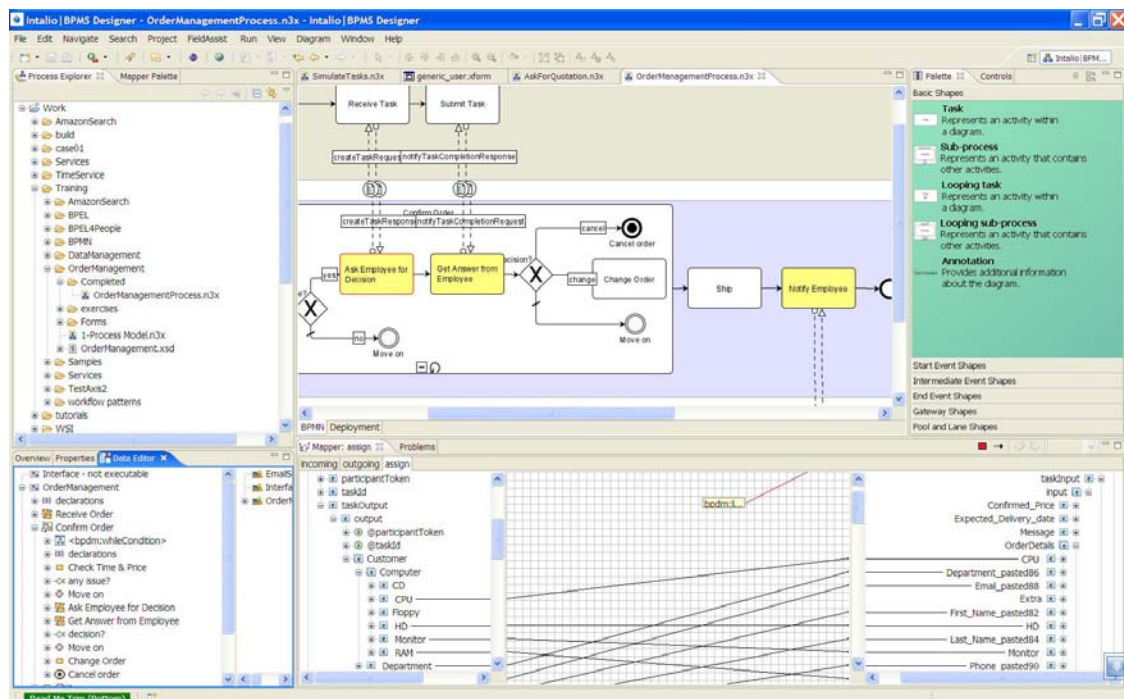
## ECLIPSE BPMN MODELER

Also, because Eclipse provides support for the most popular source control systems, integration with the customer's application lifecycle management infrastructure is greatly simplified, and the BPMS vendor gets all this essentially for free.

However, using Eclipse instead of a proprietary framework creates a couple of challenges. First, Eclipse is a very complete platform that makes extensive use of advanced technologies such as EMF and GEF. The learning curve for using these new frameworks has to be taken into account. Second, because Eclipse was originally developed for software engineers and architects, it makes everything more complex than it should be for your average process designer. A process designer is someone that is able to understand the issues exposed by the business analysts while being perfectly able to explain those issues to the IT team responsible for maintaining the different business applications.

As a result, a BPMS vendor adopting Eclipse as underlying IDE must make sure to hide this complexity under simpler user interfaces. This is not an easy task, but the benefits you can gain by running on top of Eclipse largely compensate for the migration effort.

For all the reasons outlined above as well as for the maturity of the GMF project; Intalio has adopted Eclipse as the platform to develop its Designer [DESIGNER] tool that allows both the business analyst and the IT engineer to collaborate in order to design and implement processes.



## Modeling in BPMN

Since the merge between OMG and BPMI.org, BPMN [BPMN] – Business Process Modeling Notation – is recognized as the de-facto standard notation for representing processes. Until now, there has been no standard notation for designing business processes. Event Process Chains (EPC) [EPC] is a great notation, but it is a proprietary one that is not only supported by IDS Scheer ARIS. UML Activity Diagrams [UML] are nice, but business analysts somehow cannot use them. Flowcharts are what they'd rather go for, but these tend to be limited to the modeling of single processes, which does not accommodate the requirements of a Service Oriented Architecture (SOA). Something better was needed, and this is why BPMI.org developed BPMN a couple of years ago.

BPMN is one of the most important standards that any good BPMS should support. However there is no such thing as a BPMN Modeler in Eclipse and there is a growing community that is looking for using such a tool.

Thus the Intalio|Designer team has decided to donate its BPMN Modeler code base to the Eclipse Foundation under the STP project [STP] – the SOA Tool Platform. The BPMN Modeler is obviously packaged as an Eclipse Plugin.

## BPMN Modeler

### LEVERAGING GMF

The BPMN Modeler fully leverages the Eclipse Graphical Modeling Framework [GMF] that provides components in order to develop editors based on EMF (Eclipse Modeling Framework) and GEF (Graphical Editing Framework).

Following GMF best practices, a flexible and extensible domain model for BPMN has been first created using EMF followed by a corresponding Graphical Model. The GMF generator model was later used to map those 2 models to generate the corresponding BPMN diagram Editor.

The generated code was customized to provide the look and feel defined by the BPMN 2.0 specification. Editing policies and validation layer were used to enforce the constraints defined by the OMG BPMN Committee.

The main approach for designing the BPMN domain model was to choose a modular implementation: the BPMN domain model is light-weight and only reflects shapes and containments. It does not enforce all rules and constraints. It does not define all the properties attached to each BPMN object.

It provides the advantage of totally decoupling the constraints from the model. Each time a constraint is revised by the BPMN Committee; the graphical definition won't be

re-generated. It is very close to a Model Driven Architecture [MDA] approach where the model and its constraints are linked but decoupled.

Different applications can leverage this design: for instance a client application could define all the properties in the BPMN 2.0 spec and implement the proposed mapping to produce BPEL. Another application that purely targets non executable diagrams does not need that part of the BPMN spec. It could focus on adding documentation, linking to external documents and other diagrams.

The BPMN Modeler also provides nice enhancements to GMF. We can mention for instance the ability to write vertical text or an extension to the connection routers to produce edges with rounded corners.

## Conclusion

The Eclipse Platform provides the needed foundation to build an integrated Business Process Management development environment. Given the current traction of the Eclipse project, it is likely that Eclipse will be the only alternative to Microsoft Visual Studio and thus it is very important to start leveraging the frameworks offered in the platform.

As a member of the Eclipse STP project, Intalio has recently donated the code base of a BPMN Modeler. Such a tool fully highlights the fact that Eclipse can be more than a technical development environment and business oriented features can be built on foundation such as GMF.

If you add the plugin mechanism on top of that; it means that Eclipse can become an on Demand platform where you will select the plugins that are relevant to your daily work, be it a business analyst job or a developer job.

## References

---

- [BPM20] BPM 2.0 – Ismael Ghalimi – ITRedux  
<http://itredux.com/blog/bpm-20/>
- [BPMN] Business Process Modeling Notation – OMG  
<http://www.bpmn.org/>
- [BPMS] Business Process Management Suite – Wikipedia  
<http://en.wikipedia.org/wiki/BPMS>
- [EPC] Event-driven Process Chains – Wikipedia  
[http://en.wikipedia.org/wiki/Event-driven\\_Process\\_Chains](http://en.wikipedia.org/wiki/Event-driven_Process_Chains)
- [GMF] Eclipse Graphical Modeling Framework – Eclipse Project  
<http://www.eclipse.org/gmf/>
- [MDA] Model Driven Architecture Specification – OMG  
<http://www.omg.org/mda>
- [UML] Unified Modeling Language – OMG  
<http://www.uml.org/>
- [STP] SOA Tools Platform Project – Eclipse  
<http://www.eclipse.org/stp/>