

OpenESB Newsletter

Edition1, volume1 - April 2013



The best integration environment

Editorial

Today is an important day for us, we're publishing our first newsletter dedicated to OpenESB. In this newsletter, we will provide you with information and news on the OpenESB community, end users and partners. Why did we decide to create this newsletter? All OpenESB users know that it is a powerful and efficient tool for integration, services definition and orchestration. However, it is not giving away a secret to say that OpenESB lacks visibility. However, every day, through our support and forums, we discover new large and complex projects made with OpenESB. This month we will talk about projects made by one of the biggest European Aircraft Manufacturers with

OpenESB. Since we did not get a written confirmation on time we refer to them by name. We hope this newsletter will encourage companies and people to talk about their works and project on OpenESB.

For technical people, we list the new links to our new development infrastructure set up to prepare new OpenESB versions. We have moved our source repository from SVN to Git Bitbucket. The last topic of the newsletter is dedicated to technical tips and tricks with OpenESB. We hope you will enjoy this initiative and look forward to your feedback and your collaboration.

Thank for our English reviewer Kevan Moran



OpenESB use case: Acando

Our partner Acando (www.acando.com) has in-depth skills and an extensive background on integration project with OpenESB. Acando integration teams designed, developed and deployed OpenESB projects for Telco, financial services, logistic companies and recently made a wonderful OpenESB project to manage spares warehouses and logistic for one of the biggest European Aircraft Manufacturers. In this project, OpenESB is used as middleware to facilitate message communication and transformation. We read with a great pleasure the short feedback sent by the company below.

Message from XXXXXX Spare

XXXXXX Spares, the Material, Logistics and Suppliers centre of XXXXXX, is using Glassfish ESB as a message broker component for communication with its partners. A lot of different message types, most of them standardized by the ATA, are sent to and from external partners. Beside of well-established communication channels like SITA, Fax (or even Telex), also SOAP based web services are now implemented. Using of Glassfish as an intermediate component between internal interfaces and a variety of partners gives additional flexibility, reliability, and security. While process design is made quick and easy by the graphical interface, reusing code fragments is possible by working directly on XML level. Being far more than a simple proxy, the Glassfish process validates the data, authenticates and authorizes the users and ensures that the data is delivered, if the receiver is temporarily unavailable.



OpenESB resources

A new organisation for OpenESB sources management and build has been required to support future OpenESB versions, new distributions and new platforms.

Git

We stopped using SVN for GIT since it is safer and easier for large development projects like OpenESB. Code is divided in three large parts: Netbeans-SOA, Core-Bus, and JBI-Component. Source code can be downloaded here:

OpenESB part	Source code URL
Netbeans-SOA	https://bitbucket.org/opensb/netbeans-soa
JBI-CORE	https://bitbucket.org/opensb/opensb-core/
JBI-Component	https://bitbucket.org/opensb/opensb-components/

JIRA

Please report bugs and errors here: <https://opensb.atlassian.net/>

Documentation and forums

You can find many OpenESB documentations on the web. We made a compilation of more than 70 papers, video and tutorial. This compilation can be found on <http://open-esb.net>. Registration is required to access to the compilation. The best forum on OpenESB is <http://openesb-community-forum.794670.n2.nabble.com/>. For news on the community you can subscribe on LinkedIn OpenESB forum http://www.linkedin.com/groups?gid=126145&trk=hb_side_g.

Download Binary

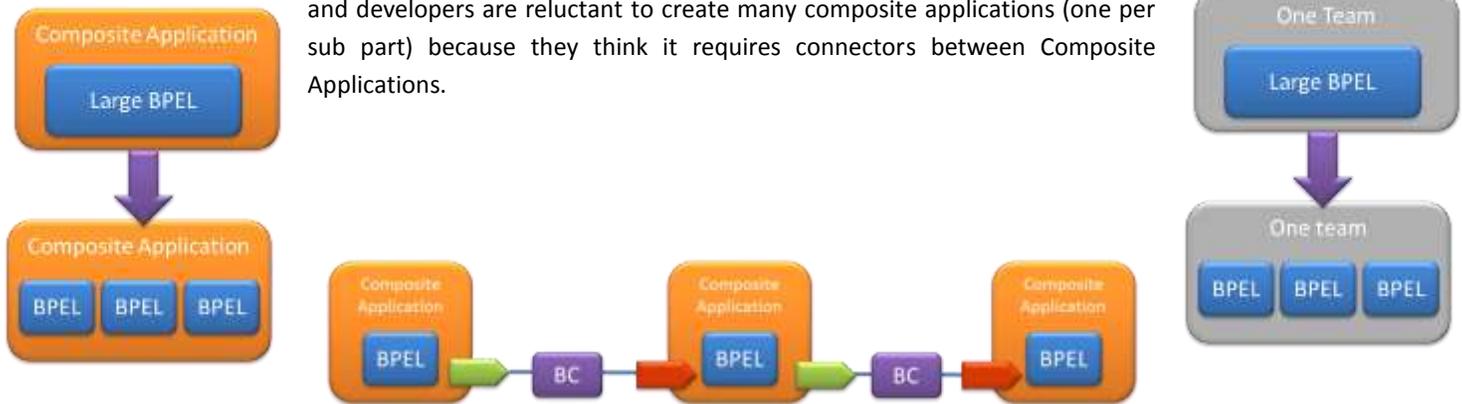
Today www.open-esb.net proposes you to download three OpenESB versions V2.1 (deprecated), V2.2 and V2.3 beta 01. If you are new in OpenESB, start with 2.2 and wait for V2.3 that will come very soon. V2.3 Beta can have some unexpected behaviour that can disturb new joiners.



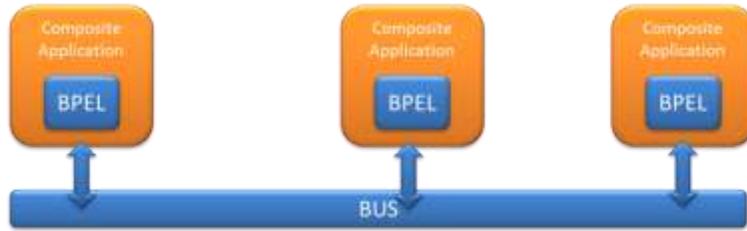
OpenESB tips and tricks: OpenESB BPEL as external module

When BPEL size becomes significant, complications can occur and impact development efficiency. The first and the most obvious reason for that is BPEL size. Very few developers have a 31" screen on their desk to watch the entire process at once meaning most spend a lot of time browsing up and down. Moreover, large or very large BPELs are not really reusable and they are difficult to debug and maintain. In order to avoid large BPEL development, we can split the entire process into smaller processes as we do in any other language. It is interesting to see that many team managers are unable to apply this simple rule and deploy huge BPEL. Others projects aware of maintenance issues split the BPEL in few parts and gather them in the same composite application. Within this configuration, BPEL subparts are linked together in the same composite application and technical dependencies can appear. In our example, in the left case one team works on the large BPEL and in the right one, one team works on BPEL subpart because they belong to the same composite application. Some architects

and developers are reluctant to create many composite applications (one per sub part) because they think it requires connectors between Composite Applications.

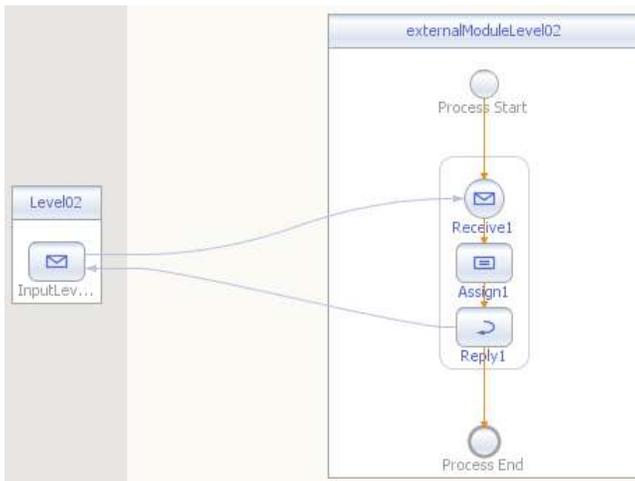


To avoid many Binding Components in a composite application OpenESB recommends that you define a BPEL project as External module. Messages between BPEL modules will be sent directly through the bus and received directly from the bus.



How to do it

- 1- Create a BPEL module with XSD, WSDL and the subpart of the process



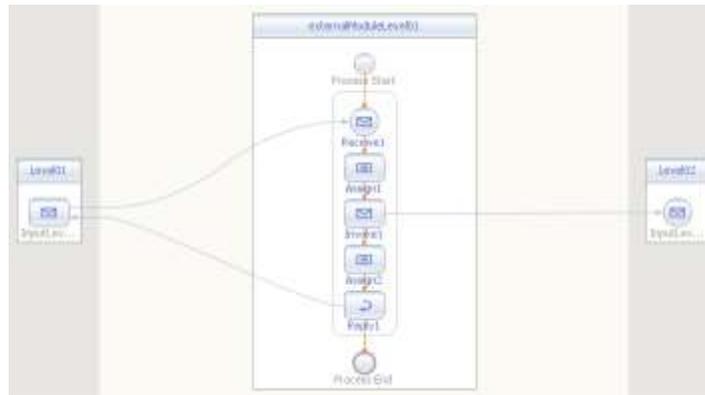
- 2- Create a composite application and put the BPEL module in the composite application. Don't add any binding component and deploy the composite application as is.



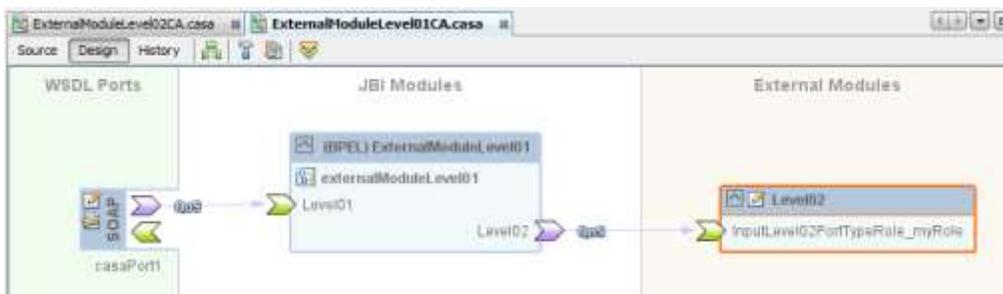
Then click right on the green arrow. Note Interface, Service and Endpoint names.



- 3- Create a new BPEL module where you invoke the previous BPEL.

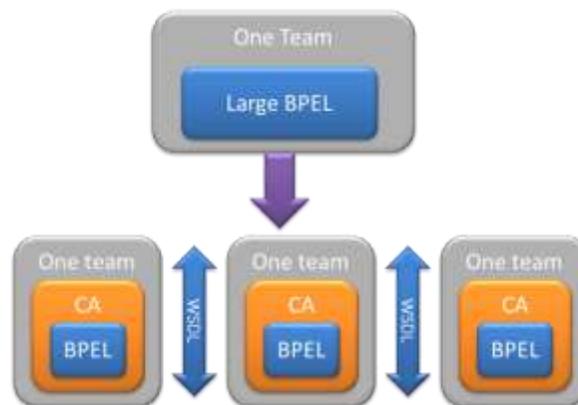


- 4- Create a Composite Application for this BPEL and add an external module. Setup the external module with Interface, Services and Endpoint names as below. You will be able to connect the second BPEL with the external module. Then add a SOAP BC to test your BPEL Chain.



Conclusion

By using External Module proposed by OpenESB, you minimise coupling between subparts of business process and are able to dedicate a development team to each sub part of the process



OpenESB partners

In system development, Acando (www.acando.com) has extensive experience working with customised IT support to develop customers' operational processes. Acando has the expertise to help its customers with solutions based on leading platforms and development environments, often in combination with components and frameworks based on open standards and open source code. Acando is also an internationally leading supplier in the area of business integration. This guarantees a strategically correct start, an efficient organisation and management of



integration issues, as well as quality-assured integration deliveries. Acando has many years of experience and top level expertise in the areas that are critical for efficient management and delivery of integration projects. Acando systematises its experience from previous integration projects to offer cost-efficient solutions that provide high quality and deliver concrete business value.



Our sponsor:

For many years, Pymma has been deeply involved in the OpenESB community and is one of its founders and main sponsors. We invest time and budget to create a new development infrastructure, get expertise on the code developed by SUN, issue new versions, and promote the product. Thanks to time and budget investments, we developed a unique skill and



an expert's network on the market. We propose OpenESB best services: Training, Consulting, Development and Support. By starting a partnership with Pymma, you benefit from this unique knowledge and make it available to your projects and your development teams. Kick off with OpenESB and take advantage to the best integration tools.

Contact US: contact@pymma.com

Wordwild: + 44 79 44 36 04 65

Germany: +33 6 51 01 73 82

USA-CANADA: +1 514 578 26 18

Pymma Consulting
is proud to support
OpenESB community
and its newsletter

Training
Consulting
Development
Support

if your company wants to sponsor this newsletter or present an interesting OpenESB project please contact us at contact@open-esb.net