ERP Integration in S-BPM Processes

15

Max Dirndorfer

Abstract

Introducing a standard ERP system gives best-practice processes to companies; but what if a company has developed a better practice and wants to implement it? ERP projects are often very time and resource consuming. One alternative opportunity is the use of S-BPM together with ERP systems. In this article, several ways are shown how this interaction can be realized. Practical examples are presented based on SAP ERP. The intention is to empower readers to apply the shown concepts to their projects.

15.1 Introduction

Over the last decades companies have invested large amounts of money in their ERP (Enterprise Resource Planning) systems (Monk and Wagner 2013). Along with the license fees for the systems, the money was mainly spent either to adapt the ERP systems to the companies, needs or to adjust the companies to fit the ERP systems processes. Often a mixture of both can be found. Even though ERP systems are established in many companies now, this does not mean the story ends here (Leon 2014).

There are still several reasons why processes within ERP systems have to be re-vised or updated, e.g., to keep up with the market and future requirements. Furthermore there are often processes or process parts that are not supported by the ERP system itself. These are often performed within further systems, which can

M. Dirndorfer (✉)
Technische Hochschule Deggendorf, Edlmairstraße 6 und 8, 94469 Deggendorf, Germany
e-mail: max.dirndorfer@th-deg.de

A. Fleischmann et al. (eds.), *S-BPM in the Wild*,
DOI 10.1007/978-3-319-17542-3_15

lead to isolated data islands (Müller and Loeblich 2013). This contradicts the paradigm of central data storage as it is proposed by ERP systems.

These are some reasons showing that there is still much space for ERP projects. In praxis, traditionally ERP projects are realized in one of the following two ways:

1. The ERP is complemented with the needed functionality. In SAP ERP environments this means individual ABAP program code. The strategy can lead to several disadvantages like the risk of making the system incompatible with vendor updates.
2. Additional systems for particular tasks with interfaces to the ERP system are introduced (e.g., special CRM systems). Possible disadvantages of this solution can be that those additional systems also may not fully support the requirements, which leads to similar problems as those with the actual ERP system.

Due to the disadvantages of these two methods the idea came up to use the features of S-BPM and the Metasonic Suite to describe and run additional functions extending the ERP systems standards while being able to use the agile and holistic features of S-BPM (Fleischmann et al. 2013; Obermeier et al. 2014).

15.2 Project SUGGEST

To accomplish the idea of using S-BPM along with ERP systems, Project SUGGEST[1] was initiated. It was a research project located at TH Deggendorf conducted in cooperation with Metasonic AG with a term of about one year starting in March 2013. The project was under the scientific direction of Prof. Dr. Herbert Fischer on the university's side and Nils Meyer (CTO) on the side of Metasonic. The purpose of project SUGGESTS was to elicit the interaction between S-BPM and ERP systems, especially SAP ERP. With the goal of implementing real working examples a team of one scientific associate as well as two student apprentices were settled at TH Deggendorf. The team worked closely together with the software engineers and consultants of Metasonic. At every stage of the project the expertise of various experts was involved, including experts from consulting as well as IT and process staff from corporate users.

The first investigations were aimed to find out which basic kinds of interaction between S-BPM and ERP systems can be useful. In small workshops with ERP and S-BPM experts it was discussed what forms of interaction can add a reasonable value for practice. Initially, five relevant categories were identified; see Fig. 15.1. Those simply describe how the S-BPM processes and the ERP systems are related:

1. *Upstream processes*, meaning an overall process starts outside of the ERP system in an S-BPM process and at a certain point the process execution and the

[1]SUGGEST is a German abbreviation. In English it means something like subject-oriented design of ERP systems.

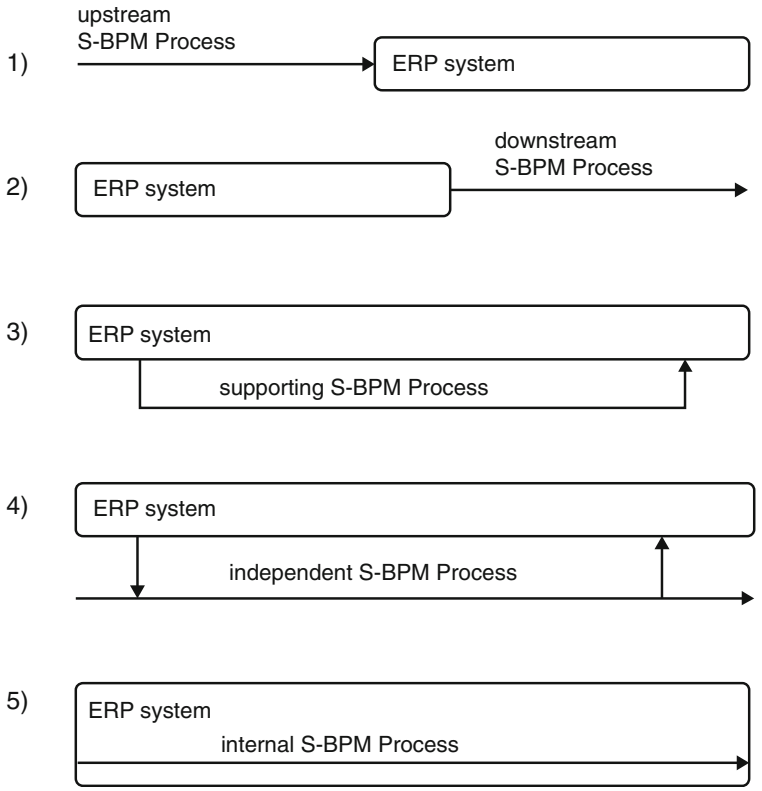

Fig. 15.1 Ways of interaction between S-BPM processes and ERP systems

collected data are passed to the ERP system. These can be preparing processes, e.g., to collect data outside of the ERP system by users with no ERP access.

2. *Downstream processes* are started within the ERP system. In the next step an S-BPM process is triggered and relevant data is passed over. The area of application is similar to the upstream processes, except that they are for follow-up tasks.
3. *Supporting processes* are processes running basically in the ERP system. At predetermined points S-BPM processes are started to support the main process.
4. *Independent processes* run from start to end outside of the ERP system but data is read from or written to the ERP. For example, this can be a process that replaces a standard ERP process without compromising the integrity of the data of the ERP system.
5. *Internal processes* stand for the use of S-BPM to describe and run processes inside the ERP system. This requires a strong interweaving between the ERP system and S-BPM.

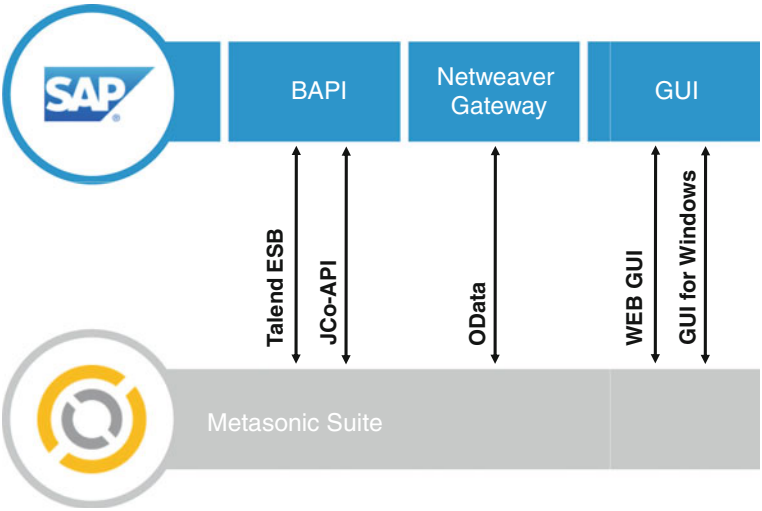

Fig. 15.2 Considered interfaces between the Metasonic's S-BPM Suite and SAP ERP

Obviously it is also possible to make all kinds of combinations of the named integration types; for example, a process started in the ERP system is continued in an S-BPM process and goes back into the ERP system.

15.3 Finding a Way of Communication

After knowing in what ways the integration between S-BPM processes and ERP systems can be useful in praxis the next step was to find out how the communication between the different systems can be realized. To keep the project manageable the context was reduced from ERP systems in general to SAP ERP, but always by cross-checking with other solutions so as not to lose the overall picture.

Relevant interfaces were identified by obtaining expert knowledge and literature reviews. Interfaces on both sides, the SAP system and the Metasonic Suite, were identified. Figure 15.2 gives an overview of the finally considered interfaces. The interfaces (displayed as black arrows in Fig. 15.2) are described as follows from left to right.

15.3.1 Talend ESB

Talend ESB[2] is an Enterprise Service Bus which is available in different versions. Some are under open source license and for some a subscription is needed (those

[2]http://www.talend.com/products/esb.

include extra features). The ESB uses a graphical mapping to design and configure services for the interaction between different systems. Talend ESB offers Connectors to address SAP systems. Via graphical connectors the data from SAP BAPIs (Business Application Programming Interfaces) can be mapped to S-BPM business objects and vice versa. The advantage of communicating with BAPIs is that the integration takes place on business level (inherent part: e.g., data checks and transactional access) and BAPIs are very well documented.

The Metasonic Suite offers the option to directly integrate talend ESB services. Without any line of code it is possible to call talend services. Therefore an interface description is required in the talend ESB which can be generated automatically while defining an S-BPM process with the Metasonic Suite. The generation is done based on the description of business objects that are defined in the respective S-BPM process.

15.3.2 JCo-API

The second method considered is the use of the API (Application Programming Interface) JCo (Java Connector). It is a Java library offered by SAP to its customers with appropriate licenses. It allows any Java program to access SAP BA-PIs and other remote enabled SAP function modules. Talend ESB described in Sect. 15.3.1 also uses JCo, but abstracts the Java programming part.

With the Metasonic Suite every state in the internal behavior of a subject can call a refinement. A refinement is custom Java code that is executed when the state is entered during the execution of an S-BPM process. To realize the communication between SAP and the Metasonic Suite, again the data objects are mapped to the interfaces of SAP BAPIs. This is realized within the Java code by the use of the library. With this method data can be exchanged between SAP and S-BPM processes not noticed by the user.

15.3.3 OData

The third way of integration is via the OData web service protocol. OData services for accessing data from SAP systems can be generated with SAP NetWeaver Gateway. OData services can be integrated in S-BPM processes via refinements by the use of OData4j (Java library). OData is XML-based and uses the REST (Representational State Transfer) concept.

15.3.4 Web GUI

The next way considered is the use of the SAP Web GUI. SAP offers a browser-based user interface. It is a one-to-one implementation of the GUI for Windows. This user interface can be directly embedded as an external application via its URL

in a state in an S-BPM process. This can be used for processes running mostly outside of SAP. In a particular process step an SAP transaction can be called directly and viewed by the user within the Metasonic GUI. It is also possible to pre-allocate fields in the SAP GUI with data collected in the S-BPM process.

15.3.5 GUI for Windows

Similarly to the Web GUI integration, it is also possible to open the SAP GUI for Windows from an S-BPM process. This is realized by the generation of an SAP shortcut which is generated and opened in a process step. Again, SAP transactions can be accessed directly and fields can be pre-allocated.

15.4 Practical Application

For each of the integration options described in Sect. 15.3, a prototype implementation was created based on a practical use case. The results were shown to corporate users. Thereby important clues could be identified about which integration forms are particularly suited for what purpose and what further forms of integration are required. Some improvements have been transferred directly back into the prototypes.

All prototypes are based on the same use case. An employee can make a purchase requisition (BANF). For this purpose he or she must specify the relevant data such as material number, quantity, or delivery date. After that the requisition has to be released by his line manager if it is higher than a certain amount (e.g., 500 EUR); otherwise it is released automatically. All necessary information has to be stored in the SAP system.

15.4.1 Prototype Talend ESB

In this prototype the talend ESB has to technically interact with two systems, the SAP ERP and the Metasonic Suite. The talend ESB has the main advantage that there is no need to write program code.

First of all an S-BPM process model was designed; see Fig. 15.3. It uses three subjects, two for human participants, “Purchaser” and “Approver”, and one fully automated for the SAP system. The SAP system does not necessarily have to be a separate subject, but the communication with the SAP system could also be handled via the internal behavior of the human subjects. By using a separate subject all technical data exchange with the SAP system is encapsulated.

The second step is to establish the connection between the S-BPM process and the ESB; see Fig. 15.4. This is handled via the internal behavior of the SAP subject. In the state calling an ESB service, a business object has to be chosen as the source, for the service input data, and one as the target of the service output data. The necessary schema for talend can be generated and imported in talend ESB.

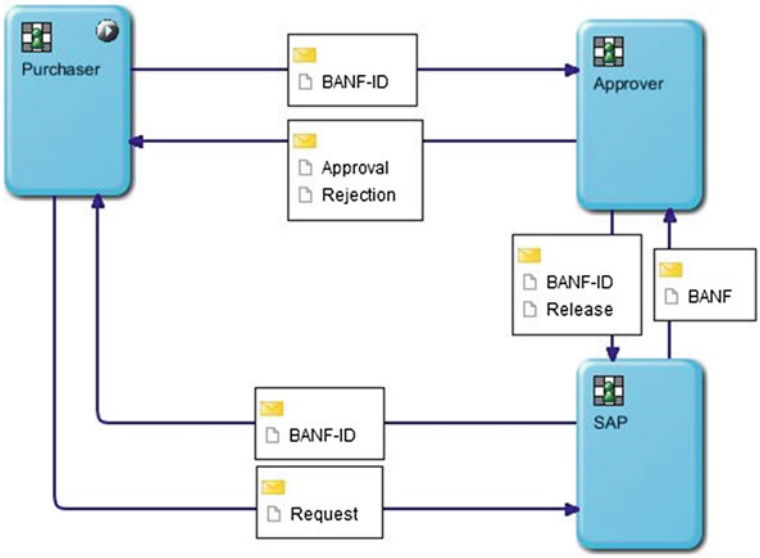

Fig. 15.3 S-BPM subject interaction diagram for Talend Prototype

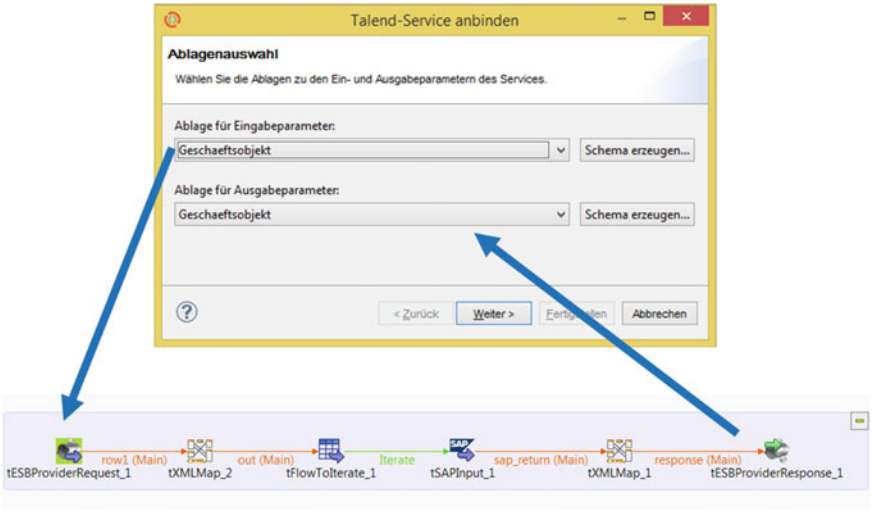

Fig. 15.4 Talend ESB service calling an SAP BAPI mapped to a business object from an S-BPM process

The third step is the realization of the communication between talend ESB and SAP ERP. The lower part of Fig. 15.4 shows the model of the talend ESB service. The service gets the data from a business object of the S-BPM process. This data is

mapped and sent via a "tSAPInput" connector to an appropriate SAP BAPI. The answer of the BAPI is mapped back to a business object of the S-BPM process and the data can be used in the next S-BPM process. The users of this prototype only use the GUI of the Metasonic Suite with no need to open the SAP GUI.

This approach has one big advantage: The graphical mapping needs no programming skills to integrate S-BPM with SAP ERP. However it must not be assumed that it can be realized by any employee of any department. At least some affinity towards IT and familiarization with the talend ESB is required. Thus the integration is realized on the BAPI layer of SAP ERP; this approach makes use of their inherent business logic and control mechanisms.

15.4.2 Prototype JCo-API

The second prototype uses the JCo-API. This means some source code has to be written. The Metasonic Suite allows running custom Java code as refinements from any state of an internal behavior. The advantage is that there is no need for third-party software. The communication is realized directly between the Metasonic Suite and SAP ERP.

The basis for this prototype is the same S-BPM model as that used in Sect. 15.4.1 (see Fig. 15.3). The internal behavior of the SAP subject now doesn't call a talend ESB service, but starts a refinement. By the use of JCo it is possible to use the functionality of an SAP BAPI. Again the input and output data from the BAPI is mapped to a business object, but this time within the Java code. The following sample code exemplarily shows a simple call of an SAP BAPI and is not the detailed implementation used in the real prototype.

```
...
JCoDestination destination = JCoDestinationManager
  .getDestination(DESTINATION_NAME);

// selection of SAP function
JCoFunction function = destination.getRepository()
  .getFunction("STFC_CONNECTION");

// creation of import parameters
JCoParameterList importparam = function
  .getImportParameterList();
importparam.setValue("REQUTEXT", "foobar");

// function call
function.execute(destination);

// reading export parameter (response)
JCoParameterList exportparam = function
  .getExportParameterList();
String resptext = (String) exportparam.getValue("RESPTEXT");
String echotext = (String) exportparam.getValue("ECHOTEXT");
...
```

In the example above a very simple BAPI "STFC_CONNECTION" is called. The BAPI has one plain import parameter, REQUTEXT, and two plain export parameters, ECHOTEXT and RESPTEXT.

From within the refinement, access to business objects is granted via the Metasonic Suite APIs. So the mapping from import and export parameters can be handled. For the user the process runs exactly the same way as the one using the talend ESB. The users only see the GUI of Metasonic process runtime.

As with the integration via talend ESB, JCo also uses the BAPI layer; this means things like data checks and transactional access are supported. The advantages of this approach are that there is no need for additional software products and existing Java knowhow can be used directly.

15.4.3 Prototype OData

The third prototype is realized with the OData protocol. SAP NetWeaver Gateway enables SAP ERP to use OData. The internal SAP processes need not be known. The integration is done via a web service. The web services can be generated from BAPIs or can be programmed based on ABAP Objects.

Again the same process model is used (see Fig. 15.3), but the internal behavior of the SAP subject is adjusted. Once more a refinement is coded. The information objects from the SAP system have to be mapped to the business objects of the S-BPM process. The communication with the OData service is realized with the help of the OData4J library. It is possible to read data from the SAP system and use it in the S-BPM process as well as manipulate data in the SAP system (create, update, or delete).

When the prototype was realized, unfortunately no SAP system with OData support was available for testing. Therefore, it was only demonstrated that basic access to OData is possible from S-BPM processes.

This approach again needs Java programming knowledge, but has one advantage: OData is a standardized procedure for data exchange. If the OData interface is well defined and provided by SAP NetWeaver Gateway it is easy to use and integrate the functionality in the S-BPM processes.

15.4.4 Prototype Web GUI

The integration of the Web GUI into an S-BPM process is quite simple. The Web GUI as well as the GUI of the Metasonic Flow are browser-based. The Web GUI can be integrated directly in the process surface of the Metasonic Flow. The browser compatibility of the SAP Web GUI must be considered as it is not compatible with all browsers in versions.

The integration is different from the one before. The SAP interaction cannot be grouped in one SAP subject. The SAP transactions are called as an external application in the internal behavior of the subjects. It is as simple as copying the

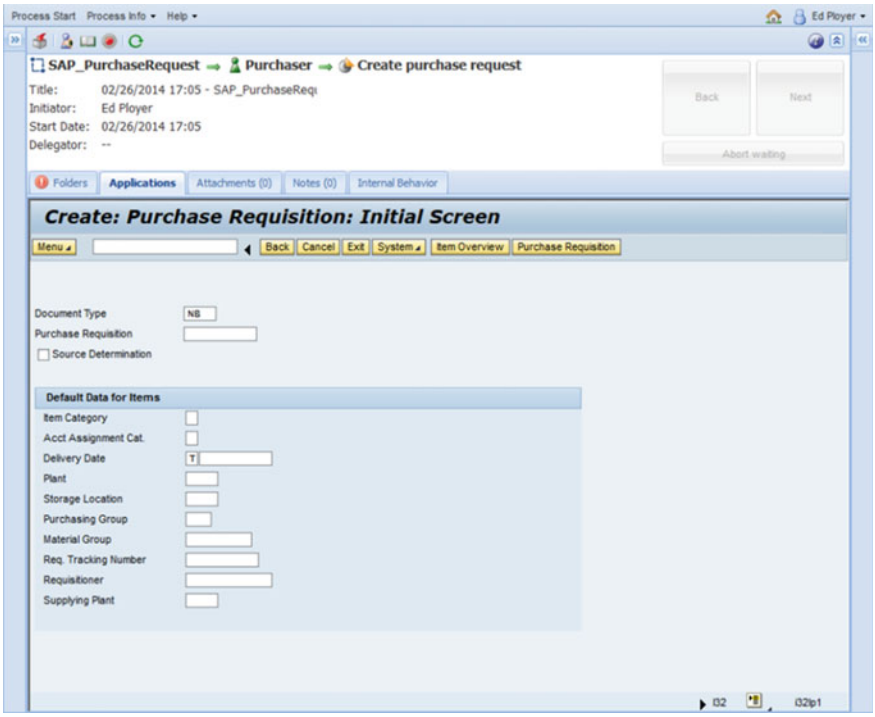

Fig. 15.5 Integration of the SAP Web GUI into an S-BPM process

URL of the Web GUI and adding some parameters for the call of a certain SAP transaction. Fields can be pre-filled with data collected from the previous process steps. The user simply sees the SAP masks integrated in the GUI of the Metasonic Flow; see Fig. 15.5.

The advantage of this approach is that it can be realized very simply. If a company has the Web GUI already in use, it is as simple as copying the URL of the SAP transaction to the relevant S-BPM process step and setting certain parameters. The disadvantage is that this approach is not as flexible as the previous ones; only the standard SAP masks are used. Adjustments at this point are only possible via the SAP system itself.

15.4.5 Prototype GUI for Windows

The integration of the GUI for Windows is quite similar to the integration of the Web GUI. The GUI for Windows cannot be integrated directly in the browser, but has to be opened in its own window. In order to achieve this a SAP shortcut file has to be generated, downloaded to the client PC, and executed. It is possible to jump into certain transactions and to pre-allocate fields.

Advantages and disadvantages of this approach are identical to those in the integration via Web GUI. Additionally, the user has to work two environments which can lead to confusions.

15.5 Results and Outlook

ERP systems can be integrated with S-BPM processes in various ways. To be able to integrate an ERP system it has to offer open interfaces as SAP ERP does. The so-far implemented prototypes do only cover 1–4 of the initially described ways of interaction (see Fig. 15.1). There is no way so far to describe and modify the internal processes of an ERP system directly; this must be part of further investigations.

Figure 15.6 summarizes the required process steps of the different integration approaches. The length of the shown processes is no indicator of the real effort. It depends on many factors such as precognition or process complexity.

The topic is of big interest to industry, as a follow-up project shows. Together with a large company from the financial sector (does not want to be named), a real application is planned. The use case there is a process that is normally performed with the company's SAP system. There are branch offices that don't have access to the SAP system. To enable them to start this process and to collect the necessary data an S-BPM process is used and the data is transferred to the SAP system. This helps to overcome the initially described problems with ERP systems. It empowers the organization to realize additional functionality while staying adaptive and agile for future requirements.

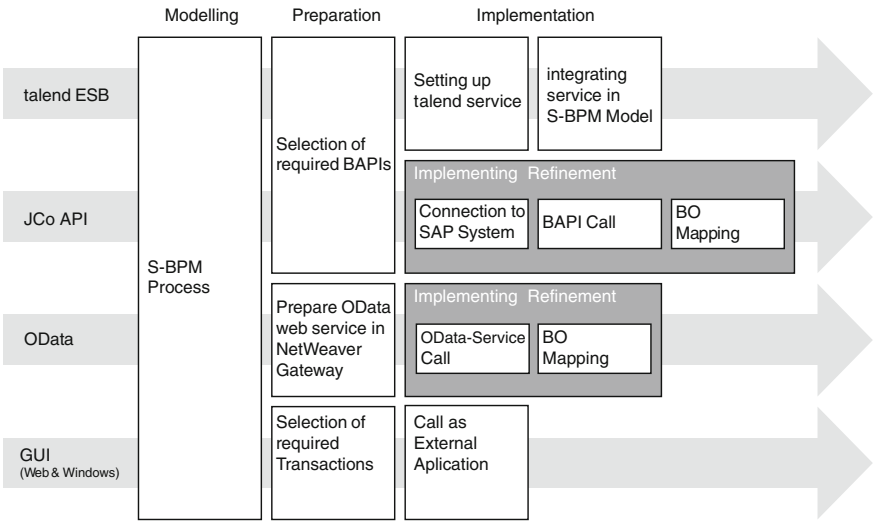

Fig. 15.6 Process steps for integration of S-BPM processes and SAP ERP

Acknowledgments The authors acknowledge the financial assistance of Metasonic in the development and evaluation of this project.

References

Fleischmann A et al (2013) Agiles Prozessmanagement mittels Subjektorientierung. In: Reinheimer S, Gluchowski P (eds) Agilität in der IT. dpunkt, Heidelberg, pp 64–76 (in German)

Leon A (2014) Enterprise resource planning, 3rd edn. McGraw Hill, New Delhi

Monk EF, Wagner BJ (2013) Concepts in enterprise resource planing, 4th edn. Course Technology, Boston

Müller S, Loeblich M (2013) Innovatives Business Process Management. ERP Manage 4/2013: 40–42 (in German)

Obermeier S et al (2014) Geschäftsprozesse realisieren: Ein praxisorientierter Leitfaden von der Strategie bis zur Implementierung. 2nd edn. Springer Vieweg, Wiesbaden (in German)