Business Process as a Service (BPaaS): The BPaaS Design Environment

Robert Woitsch[1], Knut Hinkelmann[2], Ana Maria Juan Ferrer[3], Joaquin Iranzo Yuste[3]

[1] BOC Asset Management, Operngasse 10b, 1040 Vienna, Austria,
robert.woitsch@boc-eu.com
[2] FHNW Universities of Applied Sciences and Arts Northwestern Switzerland,
Riggenbachstrasse 16, 4600 Olten, Switzerland,
knut.hinkelmann@fhnw.ch
[3] ATOS, Calle de Albarracín 25, 28037, Madrid Spain
{ana.juanf|joaquin.iranzo}@atos.net}

Abstract. This paper introduces the project idea of Business Processes as a Services (BPaaS) that is worked out in the H2020 project CloudSocket. Concept models and semantics are used to align domain specific business processes with executable workflows that are deployed and in production in a multi-cloud environment. The Business Process Management System Paradigm (BPMS) is requesting the functional capabilities of the so-called BPaaS Environments (i) design, (ii) allocation, (iii) execution and (iv) evaluation, which technically compose the CloudSocket Broker platform. This paper introduces first findings of aligning customers' business needs with BPaaS cloud offerings using a model-based approach.

Keywords. Business Process as a Service, Cloud Workflows, Business and IT Alignment

1 Introduction

There is currently a rapid growth of services published by companies on the internet, and a need to efficiently combine them, since it is a key to economic success. For a time, companies have been implementing Service-Oriented Architectures (SOA) in their businesses through commercial products, for companies like, for instance, IBM, Microsoft, SAP - obtaining more flexibility and agility in the business processes. Hence, more and more companies, suppliers and markets need to create, adapt and/or restructure their business processes and workflows in order to create added value. The project idea of the EU-project CloudSocket is to abstract parameters from pure technical distinctions up to business and domain-specific characteristics, in order to describe and distinguish cloud offerings on domain-specific business processes.
The challenge of Business Process as a Service (BPaaS) is to address business and IT-alignment. Modelling of business processes involves different profiles having different points of views and diverse scopes of knowledge. Regardless of the chosen standard to design business processes, the smooth interaction between the different actors -

such as business analysts and service integrators - is essential to define and exploit efficient business processes. Typical for such a semantic distance there is a dispersion of required knowledge: a) on the business level, the domain-specific business processes, involved actors, requirements, business constraints and indicators are identified, whereas b) onthe technology level, the business inputs are analyzed and the executable workflow is created.
Hybrid business process and workflow models are developed to realize the different abstraction layers in one modelling environment. In the following the design of BPaaS as one of the core parts of the EU project CloudSocket [1] is introduced.

2 BPaaS Introductory Sample

We use the business process for sending Christmas cards to illustrate the idea of BPaaS. Although it is obvious that such a simple business process is embedded in a Customer Relationship Management software that may be provided as a SaaS solution, we use this well-known context to introduce the vision of BPaaS.

First, the business process flow describes the activities of Christmas card distribution, such as using pre-selected images or creating own images, entering the text, uploading the recipient email-list or when to send the email. Those actions may cause issues in the cloud with respect to image copyrights, legal compliances of the text, legal compliances of storingprivate data – such as email addresses – and, finally, IT resources' allocation, in case all emails are sent at the same time.

Such aspects are considered Cloud-specific extensions of the business processes that are necessary to configure the technical behavior of (a) the workflow – e.g. introducing services that check the copyright of an image vs. manual acceptance of terms - and (b) the IT-infrastructure – e.g. data processing of private data within Europe.

The domain specific business processes are transferred into executable workflows by interacting between business and IT-cloud offerings. For example, the integrator can gather more information in order to identify potential cloud services to cover the defined features and the business analyst can support the integrators in finding the best option together, in order to avoid entering unnecessary loops, or worse, having misunderstandings that might lead to an incorrect definition of the business process.

After the allocation, the BPaaS is defined as (a) the domain specific business process information, (b) the executable workflow, (c) the deployment rules and relevant deployment packages, as well as (d) Service Level Agreements (SLA), Service Level Objectives (SLO) and indicators that specify how the BPaaS is monitored.

The BPaaS is offered in a marketplace that is similar to a SaaS marketplace. After buying the BPaaS in that marketplace, it is automatically deployed in a multi-cloud infrastructure and the operation of the BPaaS is monitored with several sensors. Semantic abstraction and human interaction enable the abstraction back from BPaaS logs to business dashboards, indicating that all data has been stored in Europe, no data violation took place, and cloud-bursting had been performed within the limits of additional IT costs. This dashboard enables a learning cycle.

3 BPaaS As A Concept

Business Processes in the Cloud is mentioned in the literature [2], but typically as executable business processes used as a synonym for workflows [3, 4, 5, 6]. Workflows are seen as orchestrations of IT services in the cloud.

CloudSocket uses BPaaS as business and IT alignment and therefore focuses on the composition of cloud based domain-specific business processes, having low-coupling, modular and encapsulated functionalities, as well as decoupling components that can react more quickly to changes in the business process.

In order to distinguish the different terms, a Taxonomy Wiki is published at [7]. We define the following four layers as expressed in [8]:

Layer I – Domain Specific Business Processes describe the activities of a worker, which are, in the way they are presented, not executable by a workflow engine.
Layer II – Executable Workflows represent workflows that orchestrate the interaction between human tasks and tasks executed by software services. One domain specific business process typically maps too many executable workflows.
Layer III - Cloud-deployable Workflow-Bundles are packaged for cloud deployment consisting of all relevant configurations, so that it can be deployed in the cloud automatically on demand. One workflow typically maps to many bundles.
Layer IV – Deployed Workflow in Production reflects a bundle that has already been purchased and deployed in the cloud. The corresponding workflow is ready to be instantiated and executed by the user by starting the workflow.

In order to align the aforementioned four layers, each layer has to be described in an appropriate form. Hence, there are two challenges to be met: Firstly, to find appropriate representation formats for each individual layer, secondly to find appropriate "weaving" mechanisms to link the different layers.

3.1 Related Work

Meta-modeling is a well-known approach to provide concepts and instruments for (a) the appropriate representation of the layers with concept models as well as (b) the realization of weaving mechanisms. Besides the technical functionality, the provision of a model repository, as well as the flexible adaptation approach, the ADOxx platform is collaboratively developed via the ADOxx.org [9] community with more than 1000 developers and more than 3000 stakeholders world-wide. Hence, in order to guarantee sustainability after the project period, the whole conceptual implementation is performed on the open, collaborative CloudSocket development space of ADOxx.org [9].

There are different standards to cover, not only the business approach but also the technical domain. Business Process Execution Language (BPEL) [10] defines an interoperable integration model that should facilitate the expansion of automated process integration between partnership through the exposed Web Service Definition Language (WSDL) and focusing on the technical point of view. It has appeared as a convergence of language features from IBM's Web Service Flow Language (WSFL)

and Microsoft's XLANG. This standard has been adopted by the main companies (IBM, Microsoft, BEA, ORACLE and SAP) in different execution environments like Bitztalk (Microsoft), BPEL Process Manager (Oracle), WebSphere Business Modeler (IBM) and WebLogic Integration (BEA). The reference for Open Source environment is Apache ODE, and there are others like Orchestra and ActiveBPEL. All of them implement the standard BPEL and others like WS-HumanTask [11] and BPEL4People [12], which intend to support a broad range of scenarios that involve people within business processes, creating human tasks in a service-oriented manner.

Business Process Modeler (BPM) focuses on the business side using notations, such as BPMN 2.0 [13], IDEF [14], BPMS [15, 16], UML [17], BPAL [18] or CMMN [19] each preferable, depending on the aspects that have to be described. Business processes with the intention to be executed preferably use BPMN2.0 that describes the shapes and connections for drawing business-process diagrams as well as their meanings and file formats. The execution environments are more recent, and the new standard BPMN2.0 is implemented mainly by Open Source solutions, such as Activity, Camuda, BonitaSoft, JBPMN, or Open Use solutions like ADONIS® CE.

In CloudSocket, the format (a) BPMN for business processes, (b) CMMN [19] for case-management and (d) DMN [20] for decision-management have been selected. The Enterprise-modelling framework from Zachmann [21] is used as the basic ontology for business and IT alignment. The CAMEL format [22, 23] is used for modelling Cloud-based applications.

The ADOxx meta-modelling platform is used to develop the BPaaS Design Environment in order to (a) realize a hybrid modelling tool that can model both aspects – the domain specific business processes, as well as the cloud- specific technical workflow, (b) enable semantic lifting of business processes and workflows to facilitate smart business and IT alignment within a powerful meta-modelling environment.

3.2 BPaaS Concept and Approach

We embed BPaaS into the BPMS [16] paradigm. The functional capabilities are specified in the CloudSocket architecture [8] that consists of loosely coupled, and hence exchangeable and partly optional environments. The BPaaS Environments that supports the BPaaS lifecyle are (a) BPaaS Design Environment describes business processes, business requirements and workflows, (b) BPaaS Allocation Environment linking deployable workflows with concrete services, (c) BPaaS Execution Environment that executes and monitors the workflow, and (d) BPaaS Evaluation Environment that lifts key performance indicators back to business level. Additionally (e) the BPaaS Marketplace is required to enable the customer to buy the BPaaS.

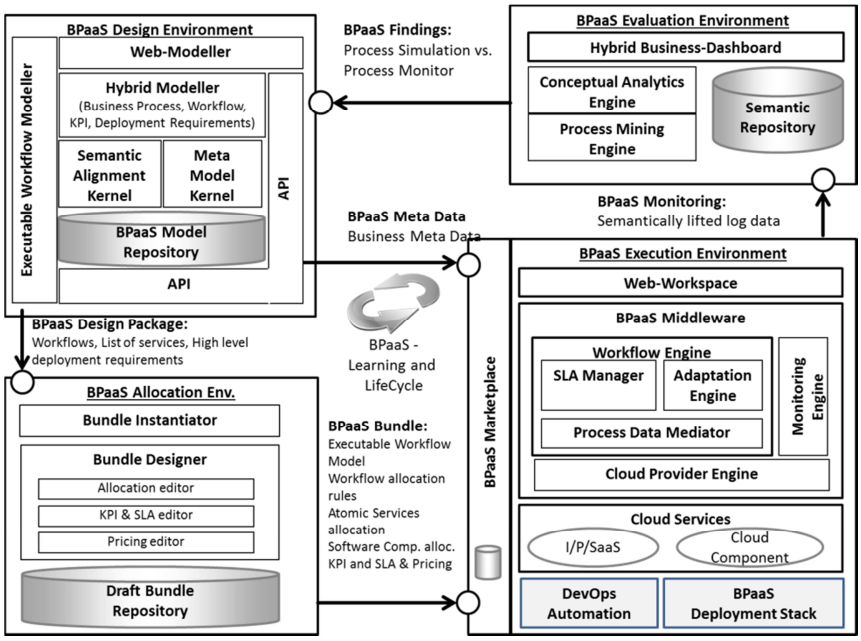

Fig. 1 High Level Architecture of BPaaS Environments [8]

Fig. 1 introduces the four major building blocks, whereas each of the four building blocks supports one phase of the BPMS paradigm when applied for business process management in the cloud. In the following the expected use cases and major components per BPaaS Environment is introduced, for a detailed description of the technical components, please refer to [8].

BPaaS Design Environment

The BPaaS Design Environment has the overall goal to design all parts of the BPaaS. Such a BPaaS design includes domain specific business process models, executable workflow-models, storage of/a collection of business process models and workflows, semantically annotated business process and workflow models - in order to map sdomain specific business process models to executable workflows in a semi-automatic or automatic manner Thus, the above discussed functionality maps a domain specific need of the business process to a formal specification of an executable workflow, supporting all types of modelling that can be involved in such a transition like (a) extended business process specification in form of organizational diagrams and document models, (b) additional case-management models to introduce adaptive business processes, (c) additional decision-management models to introduce deployment decisions as well as (d) Key Performance Indicator models to introduce measures on all levels.

BPaaS Allocation Environment

The goal of the BPaaS Allocation Environment is to configure allocation directives and rules for an executable workflow model,to be deployed and executed in the cloud. An executable workflow model, as produced by the BPaaS Design Environment, does not contain information in terms of which concrete services can be exploited for realizing the functionality of the business process tasks. BPaaS Allocation Environment enables to make informed selection of which services to choose for each business process task. The same set of requirements are also driving the decision about which IaaS offerings to select in order to deploy software components of the BPaaS.

BPaaS Marketplace and Execution Environment

The BPaaS Execution Environment deploys and executes a BPaaS bundle, once this has been purchased by a customer at the BPaaS Marketplace. Thus, this environment actually takes care of deploying the BPaaS according to the deployment plan included in the bundle, as well as importing the respective executable workflow model into a workflow-engine. A messaging platform is provided in the Cloud to enable (i) synchronous operation of component elements, (ii) multiple data formats, (iii) context awareness applications, (iv) non-standard application interface protocols. Another goal of this environment is to support the monitoring and evaluation of the BPaaS according to the KPIs and SLOs that have been defined for it.

BPaaS Evaluation Environment

The BPaaS Evaluation Environment has the overall goal to evaluate a BPaaS in order to provide optimization suggestions to its designer. This evaluation comes in various forms: (a) the assessment of KPIs, (b) the derivation of best deployments for the BPaaS, (c) the production of adaptation event patterns and rules and (d) the discovery of bottlenecks and problematic business model parts. Thus, the externally seen functionality of the BPaaS Evaluation Environment maps to initiating the performance of evaluations and the retrieval of the various evaluation results produced.

4 BPaaS Design Environment: Conceptual Architecture

The conceptual architecture is concerned with different concept-model layers as well as with the semantic interaction to enable a weaving between the layers. Due to complexity reasons, only the relevant parts are introduced from the full specification [29].

4.1 The BPaaS Meta Model

The meta-model overview is provided in FDMM form [25, 26].

A meta-model is a tuple $\mathbf{MM} = \langle \mathbf{MT}, \preccurlyeq, \text{domain}, \text{range}, \text{card} \rangle$ where MT is the set of the defined model types, i.e. for i=1,…,m we have $\mathbf{MT}=\{MT_1, MT_2, \dots, MT_m\}$.

The $\mathbf{MT_i}$'s (i=1,..,m) are themselves tuples $\mathbf{MT_i}=\langle O_i^T, D_i^T, A_i \rangle$, where:

O_i^T is the set of object types or classes,
D_i^T is the set of data types, and
A_i is the set of the attributes.

In CloudSocket we use the following model types:
MT_1 Business Process Diagram (BPD defined in BPMN),
MT_2 Choreography Diagram (CD defined in BPMN)
MT_3 Company Map (CM supporting BPMN)
MT_4 Data and Document Model (DDM supporting BPMN)
MT_5 Working Environment Model (WE supporting BPMN)
MT_6 Decision Requirement Diagram (DRD defined in DMN),
MT_7 Semantic Transit Model (STM using RDF)

To introduce the relevant parts of FDMM used in our CloudSocket definition, we introduce:

MT={BPD, CD, CM, DDM, WE, DRD, STM}.

$\preccurlyeq$ defines an ordering on O^T. Let $o_1^t, o_2^t \in O^T$ we say o_1^t is subclass of o_2^t, if $o_1^t \leq o_2^t$

The **domain** is a function with domain: $A \rightarrow P(O^T)$

The **range** maps an attribute to the power set of all pairs of classes and model types, all data types, and all model types.

$$\textbf{range}: A \rightarrow P(\bigcup_j (O_j^T \times \{MT_j\}) \cup D^T \cup MT)$$

The card function

$$\textbf{card}: O^T \times A \rightarrow P(\mathbb{N}_0^+ \times (\mathbb{N}_0^+ \cup \{\infty\}))$$

For details on the modelling language, please refer to the CloudSocket development space on ADOxx.org [9].

Weaving is a technique where different models are linked with each other.

4.2 Horizontal BPMN and DMN Weaving

In addition to the MTs defined or supporting BPMN [13], we have to add cloud-specific extensions to define deployment rules. The deployment rules are defined on domain-specific business process level using the standard DMN [20]. Hence a horizontal weaving mechanism from BPD to DRD is implemented in the following way:

$$MT_{BPD_{spec}} = \{MT_{BPD_{spec}}, \preccurlyeq, \text{domain}, \text{range}, \text{card}\},$$

$$O_{BPD}^T := O_{BPD\_spec}^T$$

$$A_{BPD} := \{A_{BP_{spec}}, \text{Referenced Decisions-from}, \text{Referenced Decisions-to}\}$$

Currently, the deployment rules are expressed as “decisions” following the DMN specification. It is expected, that support and simplifications will be needed to ease the description of deployment-relevant rules on such a high level.

4.3 Vertical BPMN cloud-specific Enrichment

In order to support the vertical alignment between a business processes and the workflows, the model type BPD is extended with a cloud-specific description concept.

This new concept is introduced as Service Description (SD). It is based on the FODA approach [27], where each business process activity is analyzed according IT requirements. The class ‘Service Description’ contains attributes which describe the requirements derived from the business process for cloud services considering (a) technical, (b) legal and domain, and (c) business dimensions.

Those attributes are used to semantically lift each business process task in order to enable a pragmatic but still precise semantics in text-format - to allow free description of requirements. The expectation is that the free-text format can partially be transformed into a semi-formal representation after experience and user feedback. A semantic lifting of those requirements is foreseen to partially automate the business and IT alignment.

Fist the Attribute of the BPD is extended with:

$$A_{BPD} \coloneqq \{A_{BP_{spec}}, \text{Referenced Service Description-from}, \text{Referenced Service Description-to}\}$$

We have to extend the MT_{BPD} with the class “Service Description” as following:

$$\mathbf{O}^{T}_{BPD} \coloneqq \{O^{T}_{BPD}, \text{Service Description}\}$$

$$\mathbf{TechnicalAttrs} \coloneqq \{\text{Input}, \text{Output}, \text{Functional Requirements}, \text{Time}\}$$

$$\mathbf{BusinessAttrs} \coloneqq \{\text{Vendor Issues}, \text{Payment}, \text{Trust}, \text{Security}, \text{Costs}\}$$

$$\mathbf{DomainAttrs} \coloneqq \{\text{Privacy}, \text{Data Compliance}, \text{Location}, \text{Restriction}\},$$

$$\mathbf{LegalAttrs} \coloneqq \{\text{Relevant Regulation List}, \text{Required data checks}, \text{Required data processing}\}$$

$$\text{Then } \mathbf{A}_{BPD} = \{\text{Name}, \text{TechnicalAttrs}, \text{BusinessAttrs}, \text{DomainAttrs}, \text{LegalAttrs}\}$$

Attribute attachments:

$\forall$ attr $\in \{A_{BPD}\}$: domain(attr)= {Service Description},

$\forall$ attr $\in \{A_{BPD}\}$: range(attr)= {String},

$\forall$ attr $\in \{A_{BPD}\}$: $\text{card}(\text{Service Description}, \text{attr}) = \langle 1,1 \rangle$,

4.4 Vertical Alignment with Semantic Lifting

Weaving between business process models and workflow models can also be performed through a so-called loose coupling, where business process models and workflow models are semantically annotated with the same ontology. The so-called semantic lifting of both, the business process model and the workflow model enables the use of semantic matching mechanisms to find the best match, hence link the business process to a workflow.

There are different ways of implementing semantic lifting [28, 29], which can be used for weaving between the different modelling layers. Independent of the semantic lifting being performed manually or semi-automatically, the concept models need extensions to introduce the semantics.

Here we refer to the so-called Semantic Transit Model Type (STM) as MT_7. This implementation enables to semantically lift any object of the business process or workflow models with a set of concepts in the Semantic Transit Model, which have been imported from the ontology.

Hence, the Semantic Transit Model Type is defined as:

$$MT_{ST} \coloneqq \{O_{ST}, D_{ST}, A_{ST}\}, \text{ where}$$
$$O_{ST} \coloneqq \{Concept\}$$
$$D_{ST} \coloneqq \{String\}$$
$$A_{ST} \coloneqq \{ Name, URI, Referenced\ concept\text{-}from, referenced\ concept\text{-}to\}$$

In order to enable the semantic lifting with a reference from any object in any model type, we use the super class of all objects to define that link in form of:

$$\forall\ class\ x\ \in \{O^T_{BPD}, O^T_{CD}, O^T_{CM}, O^T_{DDM}, O^T_{WE}, O^T_{DRD}, O^T_{STM}\}$$
$$\exists\ class\ \boldsymbol{super}\text{: super} \geq x$$

domain($Referenced\ concept\text{-}from$)={ $Referenced\ concept$}

range($Referenced\ concept\text{-}from$)= {super}

card($Referenced\ concept\text{-}from = \langle m, \text{n}\rangle$) for m,n$\in N$

This semantic lifting via a MT STM provides a tool support via the references – implemented as so-called INTERREFS in ADOxx – but does not necessarily need a fully established Ontology Management System (OMS) interaction.

Hence, it is expected that the interaction with the OMS may be introduced at a later stage of the project, to raise the maturity level of semantic lifting, through a tight interaction with an OMS. The aforementioned results can be downloaded from ADOxx.org [9].

5 BPaaS Design Environment: Technical Architecture

This section introduces the parts of the technical CloudSocket architecture [8] that consists of loosely coupled, and hence exchangeable and partly optional BPaaS environments. According to the BPMS lifecycle we distinguish between (a) BPaaS Design Environment, providing modelling tools for business processes, business requirements and workflows, (b) BPaaS Allocation Environment, providing configuration tools to link deployable workflows with concrete services, (c) BPaaS Execution Environment, providing automatic deployment engines, monitoring engines and a workflow engine, and (d) BPaaS Evaluation Environment, providing semantically lifted log-data, conceptual analytic engine and a business dashboard.

Additionally, the BPaaS Marketplace is required to enable the customer to buy the BPaaS.

In order to focus on the BPaaS Design Environment, the component diagram for the BPaaS Design Environment is explained.

Fig. 2 depicts the different components of the BPaaS Design Environment that enables the browsing, selection, creation or modification of domain specific business process models. The following components are provided:

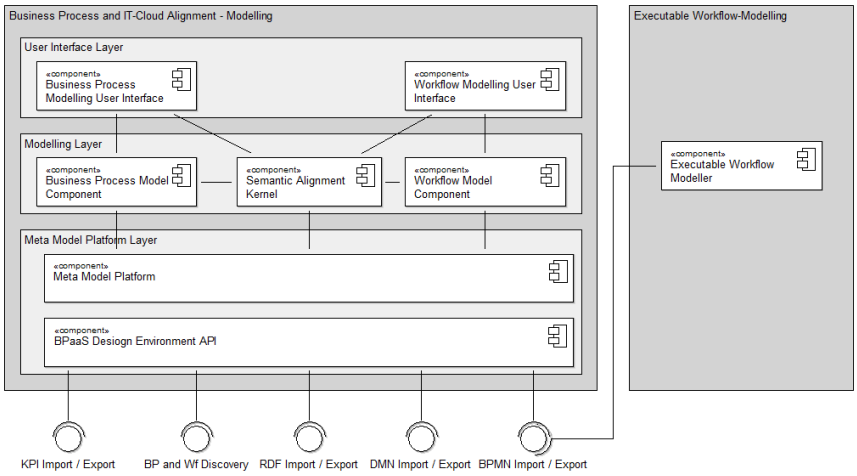

Fig. 2 BPaaS Design Environment Component Diagram

Business Process and Workflow Modelling: The user interface enables the management and manipulation of business process models and that of workflow models. It is explicitly not foreseen to model a workflow in this user interface, due to its delegation to the technical design component.
Semantic Alignment Kernel: This component realizes the semantic lifting of concept models. Depending on which lifting approach is applied, this component consists either of a set of ADOScripts that are executed within the meta-modelling platform, or it consists of third party ontology-management systems like ArchiMEO.
Meta-Model Platform: The meta-model platform ADOxx enables the storage of all relevant models within the BPaaS Model Repository. A hybrid modelling approach can be realized by configuring all aforementioned modelling languages using the features of the meta-model platform.
BPaaS Design Environment API enables different interaction protocols – using script languages and data formats – using transformation rules. REST services enable the access of models like the query and extraction of all design-relevant parts of the BPaaS bundle.

In order to ensure compatibility of the workflow designer and the workflow engine, there is an additional component added to the design environment.

Executable Workflow Modeler is compatible with the Workflow Engine that is operating the BPaaS in the BPaaS Execution Environment. Hence, the design of executable parts for the workflow is performed in the design environment of the used work-

flow engine, whereas the alignment relevant parts are modelled within the aforementioned workflow modeler. Although BPMN is used as a standard, based on current experiences it is advisable to define the executable workflows in the environment of the Workflow Engine and use the BPMN standard to import those models into the BPaaS model repository.

Data interfaces are provided in form of (a) BPMN import/export, (b) RDF import/export and (c) DMN import/export, (d) KPI import/export using a proprietary format and interfaces to discovery and retrieve (e) business process and workflow alignment information.

6 Initial BPaaS Demonstration

The business process to send Christmas cards to customers has been created as a proof of concept to introduce the idea of BPaaS. It is a simple process that allows to set forth concepts such as software components - pieces of software that can be deployed automatically on the cloud for the processes, acting as an own service - and atomic services - existing cloud services -, based on a simple process for all the roles and skills. Common terminology can be found at [7].

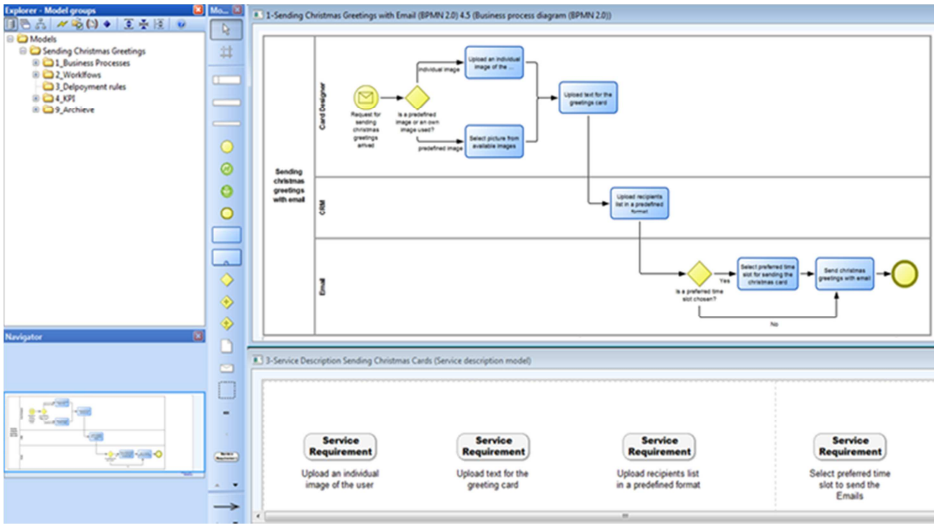

Fig. 3 Business model for the Christmas greetings card

Fig. 3 indicates the necessary activities and the relationships on the business level. This information is enhances with the introduction of i) textual explanation for the different activities, including technical requirements, restrictions, specifications and comments such as the use of predefined images and messages, preferred time slots for sendingemails ii) business goals and rules such as restrictions of data, legal compliance iii) key performance indicators (KPI) on business level such as time delivery, location of the cloud. The models on the lower window depict the service-requirement

description in form of own symbols. Each of those service requirement objects have properties to describe (1) the requirement (name, description and responsible author) (2) the function description (with fixed, flexible and free text annotation), (3) input and output data (with fixed, flexible and free text annotation), (4) Non-functional requirements are reliability, availability, a good sample-case, and planning-relevant information (with fixed, flexible and textualannotation)., (5) business description on vendor criteria, trust, security, payment, costs and business related requirements (using textual annotation), (6) regulatory description on data location, certificate, data protection and maturity model (using fixed, flexible and textual annotation). Details on the attributes and the different semantic annotations are provided in [29].

The executable process is built based on the aforementioned input. At this time, the two different points of view for the same process have to converge on the correct definition, for which it is necessary that the workflow designer contacts with the business analyst in order to start bidirectional iteration and transform the processes from the high-level business design to the executable process in **Fig. 4.**

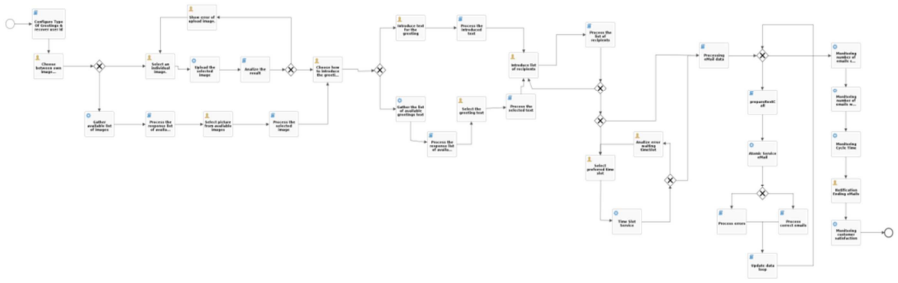

Fig. 4 Executable workflow for the Christmas greetings card

The workflow designer has to analyze the inputs provided in order to define manual tasks, internal tasks and service tasks, which could be filled by available clouds services. The following actions demonstrate the nature of design decisions:

Card Designer actions: Several technical actions are necessary such as the selection of the image and the Christmas greetings message. After analyzing the available cloud-services, the conclusion is that there are no available options to cover all the requirements. Therefore, it can be decided to use the CloudSocket software component, which is a simple service to provide the best predefined images and messages for the Christmas greeting cards.

Email actions: There are different inputs, such as a 3rd party web-application, or if it exists, own cloud-application. Hence, they have to define a clever strategy for using the cloud servers and a collaborative search to identify the best email cloud-services, since there are technical requirements (type of interfaces, protocols, security issues) and business requirements (price, legal issues, agreements, etc.). In addition, it is necessary to clarify if the service should be used only for a period during Christmas or it is worth to have a contract for the whole year in order to identify the best cloud service.

Finally, when the executable workflow is finalized, the business analyst creates the Business Process Package through the design environment; including a combination of a) Business Process (including Service Requirements Specification) as the business view, b) Workflow as the technical view, c) Rules for deployment and sensor definitions and d) KPIs and goals to measure the success of the package. The package is released in the repository and available via the API for further use in the allocation environment [8].

Additional information on the deployment infrastructure had been added to the workflow in order to build the CAMEL file [22, 23]. Pre-defined deployment plans from Virtual Engines of different sizes have been provided and combined with the workflow file, the images of the business process and workflows as well as explanatory text and payment information. As a conclusion, throughout the BPaaS composition there is a large spread of required knowledge.

7 Conclusion and Outlook

This section provides an overview of how BPaaS will be put into two demonstrations.

1. The Business Incubator form Baden Württemberg focuses on supporting the "Coaching and Finance" efforts of start-ups facilitating designing, analyzing and simulating individual business plans, and processes. These aspects also demand a high degree of adaptability of Cloud Services for Start-ups, e.g. Customer Relationship Management, Order Management, Human Resources Management. The current observation in this Business Incubator Use Case is that "supportive" business processes can be applied across several startups. BPaaS addressing e.g. Customer Relationship can be offered to a wide range of startups. "Management" or "Core" business processes are not so obvious in this first phase.
2. The Business Process Broker use case identifies typical business episodes of potential SMEs in different application domains such as eHealth, Manufacturing, Photonics, Government, Security, e-Commerce, Retails, etc. but share a common set of business processes.

   The observation in the second use case is that most of the potential end users of the CloudSocket have the potential need of generic business processes that may be individualized while the business process maturity growth. Hence, the flexible configuration of business processes, hiding the complexity of the cloud and providing easy to use solutions, is a promising market segment.

The initial phase in defining BPaaS as a concept, creating its technical architecture, describing its terminology and working out Business Process Cloud-readiness Levels has been concluded and can be found [33].

Initial tools and prototypes can be downloaded from [1, 31], and [32]. The conceptual implementation of the different model types can be collaboratively joined at the development space on ADOxx.org [9]. First research findings on improved business and IT alignment can be seen here [29].

8 Acknowledgment

We thank the other members of the core team of the CloudSocket project in setting up this project and achieving first results with us - in alphabetical order Jürgen Jähnert (BWCON), Dimitris Plexousakis (FORTH), Stefan Wesner (UULM).

9 References

1. CloudSocket Project, www.cloudsocket.eu, access: 2015.05.18
2. Stieninger M., Nedbal D., Characteristics of Cloud Computing in the Business Context: A Systematic Literature Review, Global Journal of Flexible Sytems Management (March 2014) 15(1):59-68 DOI 10.1007/s401771-013-0055-4
3. Euting S., Janiesch C., Fischer R., Tai S., Scalable Business Process Excecution in the Cloud, in IEEE Int. Conference on Cloud Engineering 2014, DOI 10.1109/IC2E.2014.13, 2014
4. Muthusamy V., Jacobsen H.A., BPM in Cloud Architectures: Business Process Management with SLA and Events, in (Eds.) Hull R., Mendeling J., Tai S. BPM 2010, LNCS 6336, pp5-10, Springer Verlag Berlin-Heidelberg, 2010
5. Lui X., Yand Y., Cao D., Yuan D., Chen K., Managing Large Numbers of Business Processes with Cloud Workflows Systems, Proceedings of the 10th Australasian Symposium on Parallel and Distributed Computing (AusPDC 2012), Melborne, Australia, 2012
6. Di Modica G., Tomarchio O., Matching the business perspectives of providers and cusomters in future cloud markets, in Cluster Comput (2015) 18:457-475, DOI 10.1007/s10586-014-0364-1, Springer New York 2014
7. CloudSocket Project, D2.2 BPaaS References, Definitions & Common Terms, https://www.cloudsocket.eu/common-understanding-wiki/-/wiki/Main/CloudSocket+Technology+and+Terminology+Wiki, access: 2016.02.20
8. CloudSocket Project, D4.1 First CloudSocket Architecture, www.cloudsocket.eu,
9. ADOxx.org, http://www.adoxx.org/live/web/cloudsocket-developer-space/space, access: 24.07.2015
10. Web Services Business Process Execution Language (WS-BPEL): http://docs.oasis-open.org/wsbpel/2.0/varprop, access: 20.01.2016
11. Web Services - Human Task (WS-HumanTask) Specification Version 1.1: http://docs.oasis-open.org/bpel4people/ws-humantask-1.1-spec-cd-06.pdf, access: 20.02.2016
12. WS-BPEL Extension for People (BPEL4People) Specification Version 1.1: http://docs.oasis-open.org/bpel4people/bpel4people-1.1.html, access: 20.02.2016
13. OMG, Business Process Model And Notation (BPMN) Version 2.0: http://www.omg.org/spec/BPMN/2.0/, access: 16.03.2014.
14. Integrated Definition Methods, www.idef.com/, access: 13.03.2014
15. BOC 2006, ADONIS®, Method Manual, BPMS Method, Version 3.9, Volume 3.
16. Karagiannis, D., Junginger, S. and Strobl, R.,1996, Introduction to Business Process Management System Concepts, in: B. Scholz-Reiter, E. Stickel (Eds.): Business Process Modelling, Lecture Notes in Computer Science, Springer.
17. OMG UML, 2009, OMG Unified Modelling Language™ (OMG UML), Infrastructure, Version 2.2, Available at: http://www.omg.org/docs/formal/09-02-04.pdf, Retrieved: 15.04.2009.

18. Nicola De A., Missikoff M., Proietti M., Smith F., A Logic-Based Method for Business Process Knowledge Base Management, http://www.iasi.cnr.it/~proietti/papers/DenMisProSmi_SEBD10.pdf, access: 13.03.2014
19. OMG, 2013. Case Management Model and Notation (CMMN), Available at: http://www.omg.org/spec/CMMN/1.0, access: 16.03.2014
20. OMG, 2015, Decision Model and Notation, http://www.omg.org/spec/DMN/1.0/Beta2/, access: 31.07.2015
21. Zachmann J.A. Zachman Institute for Framework Architecture, Enterprise Architecture: A Framework, http://www.zifa.com/framework.pdf, access: access: 16.03.2014
22. Ferry, N., Song, H., Rossini, A., Chauvel, F., Solberg, A.. CloudMF: Applying MDE to Tame the Complexity of Managing Multi-Cloud Applications. In: Bilof, R., editor. UCC 2014: 7th IEEE / ACM International Conference on Utility and Cloud Computing. IEEE Computer Society; 2014, p. 269–277. doi:10.1109/UCC.2014.36.
23. Brandtzæg E, Parastoo M., Mosser S. Towards a Domain-Specific Language to Deploy Applications in the Clouds. In Cloud computing 2012: 3rd international conference on cloud computing, grids, and virtualization, pages 213-218. IARIA, 2012
24. projects.eclipse.org/projects/modeling.tmf.xtext, access: 16.04.2014
25. OMiLAB, www.omilab.org, access: 2015.06.30
26. Fill H.G, Redmond T., Karagiannis D. (2013): Formalizing Meta Models with FDMM: The ADOxx Case, in: Cordeiro, J., Maciaszek, L., Filipe, J.: Enterprise Information Systems, LNBIP, Vol. 141, 429-451
27. FODA-Approach, http://www.sei.cmu.edu/domain-engineering/FODA.html, access: 02.12.05
28. Hrgovcic V., Karagiannis D., Woitsch R., Conceptual Modeling of the Organisational Aspects for Distributed Applications: The Semantic Lifting Approach, in the proceedings of IEE COMPSAC 2013, Kyoto, 2013
29. CloudSocket Project, D3.1 CloudSocket BPaaS Design Environment, www.cloudsocket.eu, access: 2016.02.20
30. CloudSocket Project, D2.1 Use Case Analysis, www.cloudsocket.eu, access: 2015.05.18
31. ADONIS Community Edition, www.adonis-community.com/, access: 2015.07.24
32. yourBPM, download [https://github.com/CBPM-WG/Fitman-CBPM/tree/master/Atos], based on Activiti [activiti.org/download.html], access: 2015.07.24
33. CloudSocket Project, D2.3 Cloud Transformation Framework Demonstrator, https://cloudsocket.eu/transformation/, access: 31.07.2015