UvA-DARE (Digital Academic Repository)

D4.4 Prototype of the system for enhanced services recommendation

Sargolzaei, M.; Shafahi, M.; Afsarmanesh, H.

Link to publication

Citation for published version (APA):
Sargolzaei, M., Shafahi, M., & Afsarmanesh, H. (2014). *D4.4 Prototype of the system for enhanced services recommendation*. Glonet - Glocal enterprise network focusing on custumer-centric collaboration. https://doi.org/10.13140/RG.2.1.3697.4242

**Glocal enterprise network
focusing on customer-centric collaboration**

D4.4

Prototype of the system for enhanced services recommendation

Edited by

UvA
September 2014

GloNet WP4

.2

CUSTOMIZED SERVICE-ENHANCED PRODUCT SPECIFICATION

Deliverable data

Deliverable no & name	**D4.4 – Prototype of the system for enhanced services recommendation**
Main **Contributors**	*UvA:* *Mahdi Sargolzaei, Mohammad Shafahi, Hamideh Afsarmanesh*
Other **Contributors**	
Internal **Reviews**	*Luis Camarinha-Matos (Uninova),* *Victor Thamburaj (iPLON)*
Dissemination **level**	*Public*
Date	*September 2014*
Status	*Final*

Deliverable summary

This deliverable addresses the prototypes of two tools that are implemented as a part of the PSS sub-system in the GloNet system, namely the Service Specification Tool (SST) and Product/Service Discovery and Recommendation (PSDR) tool. The designs of these tools are represented in the deliverable D4.3. These two tools, as well as the Product Specification Tool (PST) represented in D4.1 and D4.2, are well integrated as PSS (Product and Service Specification), supporting proper specification of different aspects of the complex products. The PSS sub-system provides the set of needed mechanisms for specification, registration, discovery, and recommended ranking of sub-products and business services, as well as the composition of business services and assisting with the enhancement for products with business services. As such, this deliverable addresses the implementation aspects related to the service-enhanced product specification and recommendation, while the complete design of the functionalities for this sub-system is provided in deliverable D4.3. Moreover, a set of examples of using the SST and PSDR tools, with some screenshots from the PSS are represented.
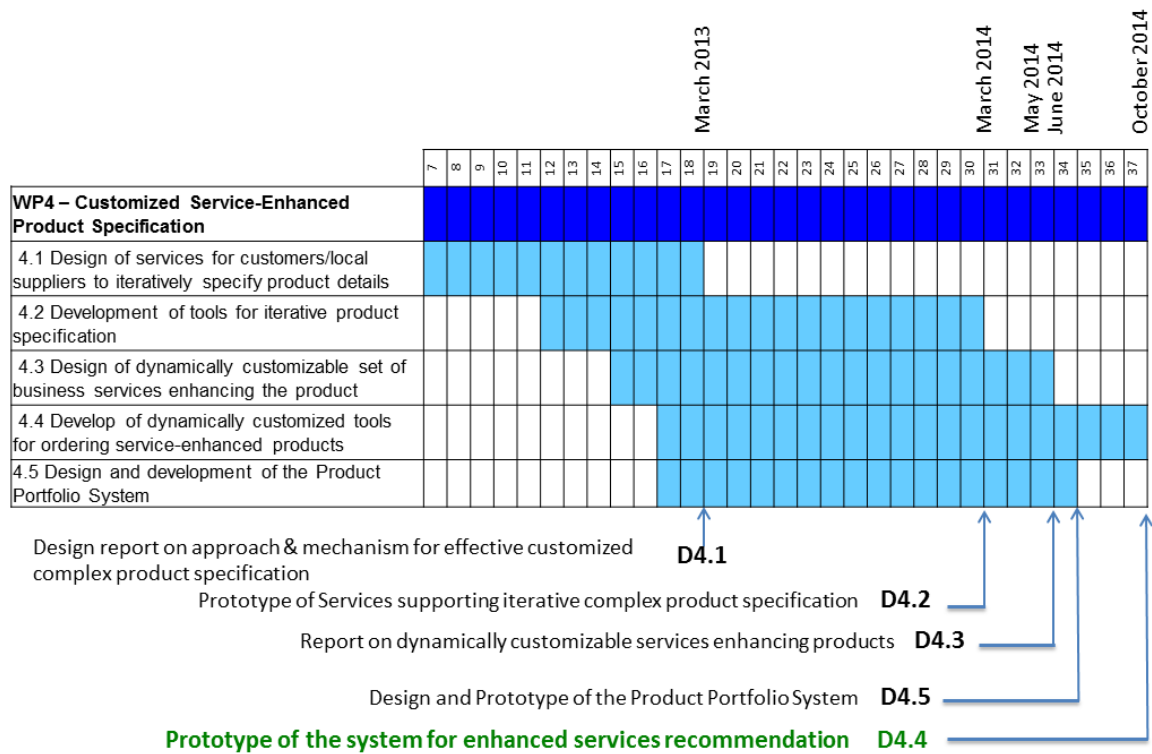
TABLE OF CONTENTS

PROJECT-RELATED SUMMARY

This deliverable (D4.4) is the fourth deliverable of WP4 that designs and develops the PSS (product/service specification) sub-system of the GloNet, and plays a main role for interconnecting the PST (product specification tool), SST (service specification tool) development of WP4 to other workpackages, and mainly WP5 and WP3. The D4.4 is the outcome of Task 4.4, and represents a detailed implementation report of several base functionalities that support the specification of business services within the complex product, and also potentially as enhancement for the specified products by their designers. The product specification functionalities have already been designed and developed, as described in the previous deliverables D4.2 and D4.3.

The deliverable D4.4 addresses the last major step in complex product specification, namely the customized service-enhanced product specification. Findings reported in deliverables: D1.1 ("Detailed requirements for GloNet use case and domain glossary") [1], D1.2 ("Specification of business scenarios") [2], D2.1 ("Required information/knowledge provision services specification") [3], D2.4 ("Mechanisms for defining composed services to support collaboration") [4], D4.1 ("Design report on approach and mechanism for effective customized complex product specification") [5], and D4.2 ("Prototype of Services supporting iterative complex product specification") [6] constitute the background for complex product specification addressed in this deliverable. However, the designed mechanisms and approaches addressed in D4.3 ("Report on dynamically customizable services enhancing products") [7] which were designed in Task 4.3, constitute the main input and the base for this deliverable. In fact, this deliverable aims at the Prototype developed for the tools designed and presented in D4.3, namely the Service Specification Tool (SST) and the Product/Service Discovery and Recommendation tool (PSDR). Furthermore, as addressed in the deliverable, SST and PSDR are also integrated with the Product Specification Tool (PST, addressed in D4.1 and D4.2). The three tools PST, SST, and PSDR together form the Product/Service Specification (PSS) sub-system of the GloNet, which supports iterative complex service-enhanced product specification.

The specifications generated for complex products in PSS then provides an input to product portfolio sub-system addressed in D4.5 "Design and prototype of product portfolio system". The following figure summarizes different tasks and deliverables of WP4.

Timeline headers: March 2013, March 2014, May 2014, June 2014, October 2014

Month numbers: 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37

WP4 – Customized Service-Enhanced Product Specification

4.1 Design of services for customers/local suppliers to iteratively specify product details

4.2 Development of tools for iterative product specification

4.3 Design of dynamically customizable set of business services enhancing the product

4.4 Develop of dynamically customized tools for ordering service-enhanced products

4.5 Design and development of the Product Portfolio System

Design report on approach & mechanism for effective customized complex product specification **D4.1**

Prototype of Services supporting iterative complex product specification **D4.2**

Report on dynamically customizable services enhancing products **D4.3**

Design and Prototype of the Product Portfolio System **D4.5**

Prototype of the system for enhanced services recommendation D4.4

.5

It is important to notice that similar to PST, the developments of SST and PSDR addressed in this deliverable play an important role in the success of functionalities addressed especially in WP5. This is due to the fact that the specifications generated by stakeholders using the developed product/service specification sub-system (PSS) of WP4, constitute a main input for identification of needed competencies from organizations, and thus selection of most-fit organizations in the VO creation process, as addressed in D5.22.

This deliverable provides a number of screenshots from the product/service specification (PSS) sub-system and is more focused on SST and PSDR.

1   INTRODUCTION

Complex products addressed in the GloNet, namely the solar power plants and intelligent buildings, are examples of products that are one of a kind in their design specifications, while they benefit from the reuse of already existing sub-specifications, and tailoring some previously specified designs to what exactly fits each specific complex product case.

Our requirement analysis of the GLONET in D1.1 [1] revealed that such complex products require supporting tools and systems for detailed specifications. Furthermore, complex products are dynamic and therefore the supporting tools for specification of products and services are needed to be used at different stages of the Product Life Cycle (PLC) of these complex products [7], [8]. Besides heavily being used during the design and engineering stage of the PLC, they are also needed infrequently during the operation/evaluation stage of the PLC, as well as during the pre-PLC phase of preparing the bid for the complex product. Please also note that the specification of a complex product including its sub-product specifications and business service specifications, is typically performed iteratively in a number of sessions, and potentially involving a number of different stakeholders at each time, who may collaborate to specify different components (sub-product and service) of the complex product. In fact, complex products specification is mostly done by some consulting or EPC (Engineering, Procurement, and Construction) companies in interaction with the customer. Several other stakeholders however (e.g. product and equipment providers, component developers, and business service providers) might also be involved in the VBE community around the complex product and will use the specification tool to provide the details of their products and services and to create awareness about them. Therefore, around each complex product, a variety of stakeholders (e.g. designers, manufactures, EPC staff, customers, etc.) can join together and form a ***Design Group*** to specify this complex product, and therefore should be supported in the GloNet.

As addressed in [7], the Product/Service Specification (PSS) sub-system consists of three main sets of tools, as also indicated in Figure 1 under the service-enhanced product support header. These include: the **Product Specification & Registration Tool (PST)**, the **Service Specification & Registration Tool (SST)**, and finally the **Product/Service Discovery & Recommendation Engine (PSDR)**.

Details related to the first tool, i.e. the *Product Specification & Registration* (PST) on its design and development are respectively provided in [5] and [6]. In [7], We have discussed the design of the other two tools SST & PSDR and addressed their needed functional and non-functional requirements, as well as our proposed approaches to realize them. Finally, in this deliverable, the implementation and development aspects of the last two tools are reported.
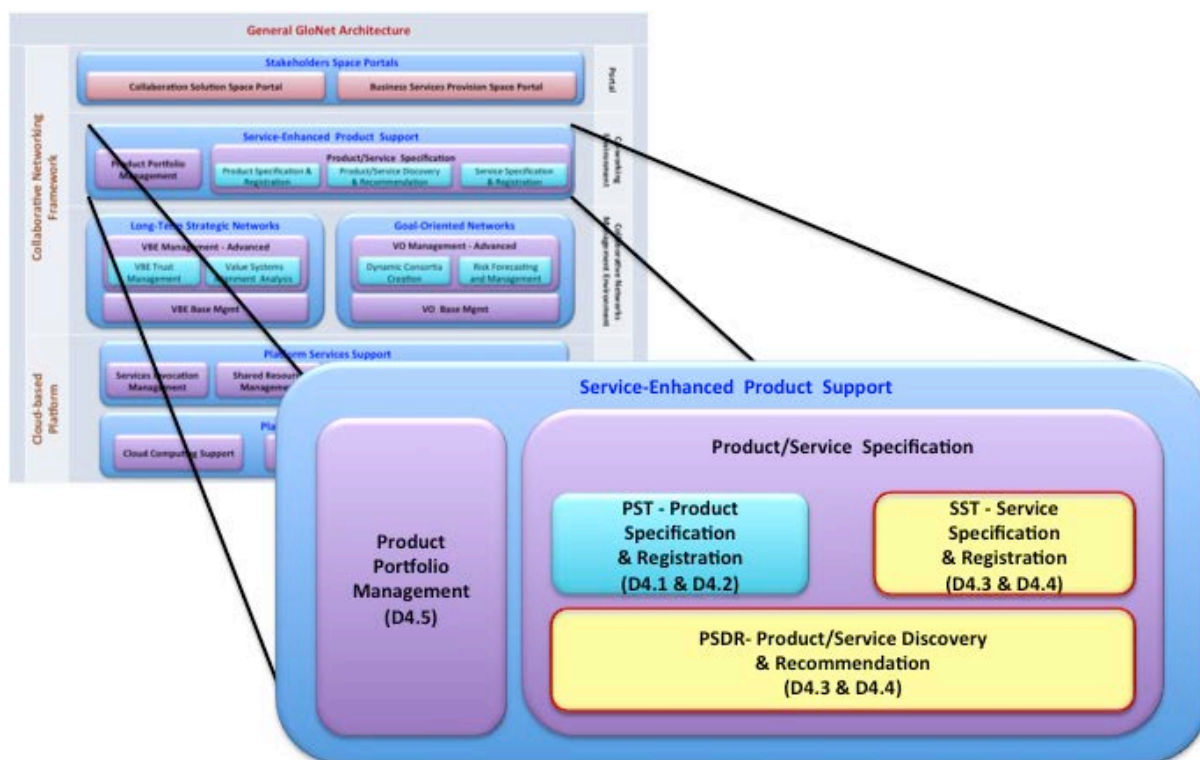
Figure 1. Service-Enhanced Product Support

Since the design of SST and PSDR prototypes have been detailed out previously (in [7]), in this document we only revisit some design aspects briefly when and if necessary to extend/augment the original design. As described in [7], each business service is materialized either manually through human activities (which we call Manual tasks), or automatically through some software (which we call Software services). The SST tool covers the specification of both of these kinds of business services, as well as the combination of some manual tasks and software services, called Composite services, as further addressed in this document.

Also it is important to note that the realizations of the non-functional requirements of the SST and PSDR (addressed in [7]), including security, scalability, availability, etc. are also supported similar to the non-functional requirements realization, of the PST, which is addressed in [6]. Therefore, we have not addressed these aspects here to avoid repetition.

The following sections of this document provide more details on certain important implementation aspects related to the service specification system, and product/service discovery and recommendation, structured in the next two sections. Section 2 addresses the implementation approach, and section 3 and 4 describe and provide examples for the use of the product/service specification sub-system. Some concluding remarks are provided in Section 5. Also in the Annex of this document, we have provided an example scenario for the use of the entire PSS sub-system of GloNet.

2   IMPLEMENTATION APPROACHES AND DETAILS

The product/service specification (PSS) sub-system is implemented in Java programming language, using the Spring [9] and Hibernate [10] framework. We have used Eclipse as the IDE (Integrated Development Environment) for our programming. Its database is built using the GloNet platform, and the MySQL [11] database management system.

In the remaining of this section, we will describe the technical details of the approach, as well as the technologies that are used to implement this sub-system of the GloNet.

2.1 General architecture

The general architecture for the implementation of the PSS is based on the MVC (Model–View–Controller) software design pattern [8], which enables modular development of this software's sub-systems. Please note that this architecture is a slightly extended version of the PSS architecture presented in [6]. While the original architecture of PSS addressed the PST (product specification tool), the extension here also covers the development of SST (service specification tool) and PSDR (product/service discovery and recommendation). Therefore, figure 2 also represents the new elements added to the entities and DAOs layers. These extensions are illustrated with boxes having text in color black. Nevertheless, the following text addresses all elements of the generated architecture that are reflected in the development of the SST and PSDR tools of each layer.
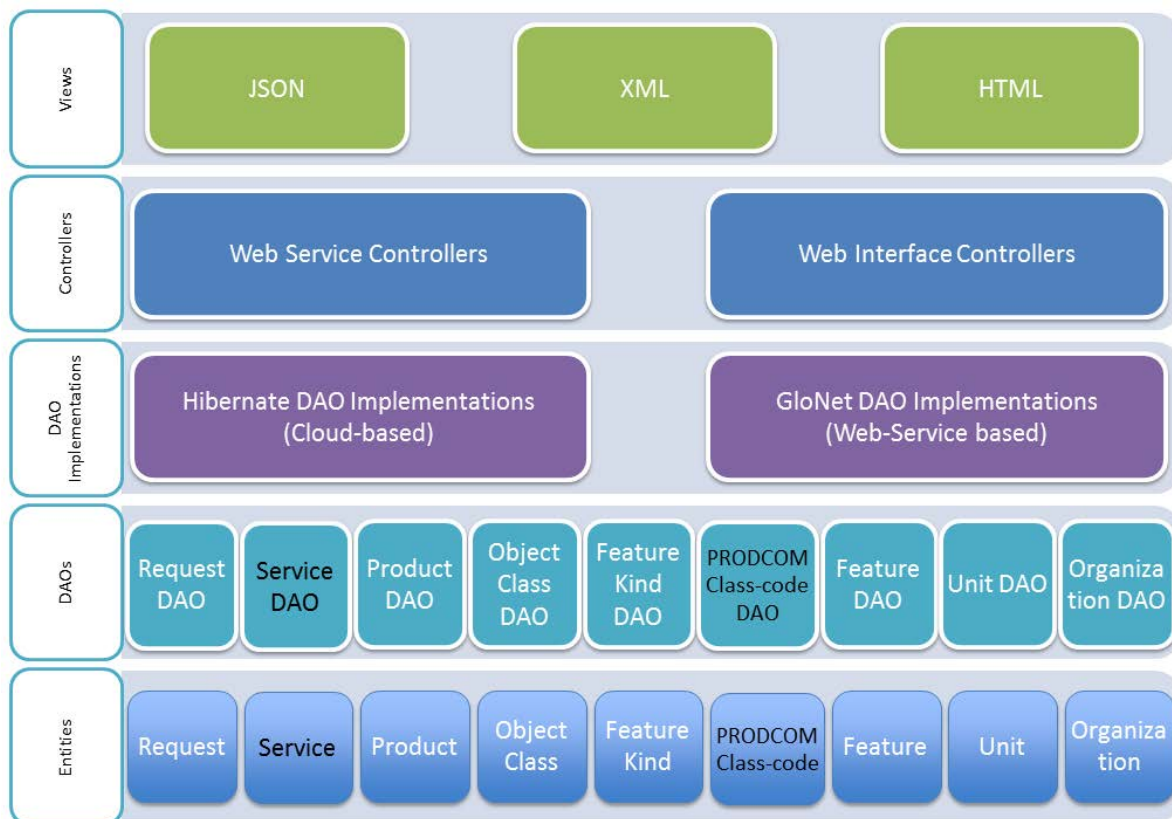
Figure 2 - Implementation architecture of the product/service specification Sub-system

Figure 2 represents the MVC-based architecture of the PSS sub-system, where it consists of five layers, as briefly described below.

1. **Entities**:
   This layer consists of the main entities introduced in the PSS sub-system, these entities are: Service, Request, Product, Object Class, PRODCOM Class-code, Organization, Class, FeatureKind, Feature, and unit. Also figure 3 shows the extended Class diagram cardinality of inter-relationships among the entity sets for the PSS sub-system indicated through the addition *Product Portfolio* and *PRODCOM Class-code* in the diagram.

   The Service entity represents all business Services (being manual or a software service), which can be defined in the PSS. The Organization entity briefly characterizes the organization/User who participates in the service specification process as a member of the *Design Group*. The Feature-kind entity represents all different kinds of features that can be defined in this sub-system within the Context of SST and PSDR. The Feature entity, characterizes different aspects of the business service, and extends every Feature-kind with its value, while the Unit entity specifies the scale for that value. The Request entity is the class representing the request for launching of a service (e.g. the composite site maintenance service in a Solar Plant, as addressed in details in [7]). The Class entity represents the generalization of the Services.
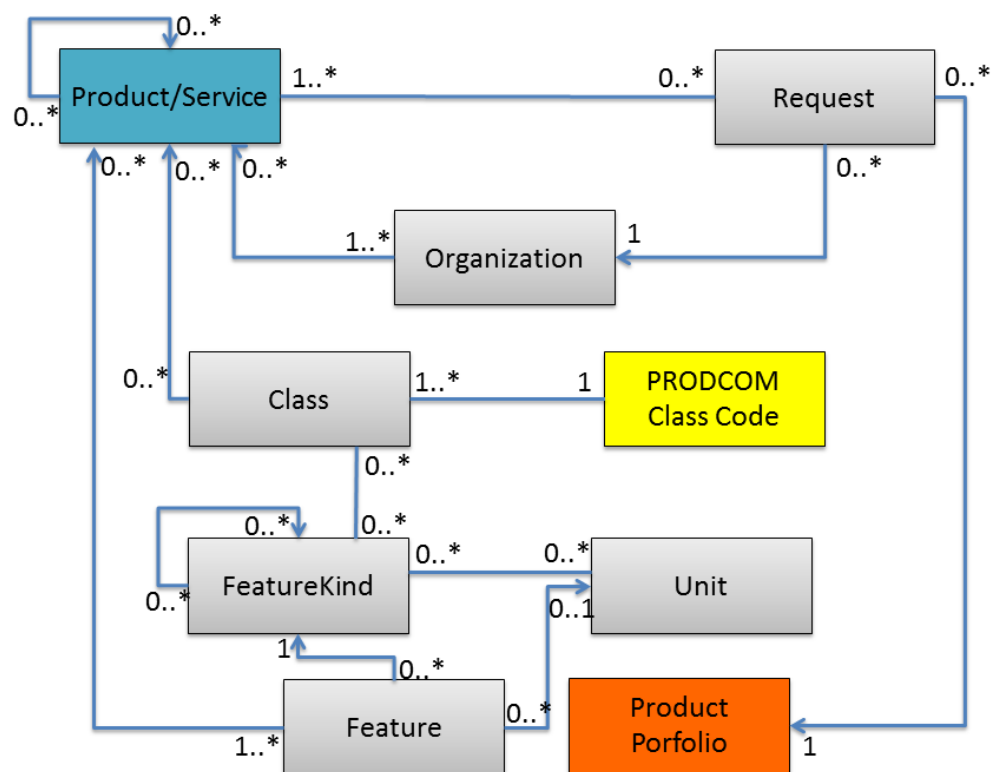
Figure 3- Class diagram and cardinality of inter-relationships among the entity sets

2. **DAOs (Data Access Object)**:
   DAO layer consists of the set of interfaces that represent the base operations developed for accessing every one of the entities defined in the lower layer, and enforces that the higher layer implementations in the PSS access all of these entities only through these provided minimum interfaces. In fact, DAO layer allows the using of entities, independent from both the technologies and the data sources.

3. **DAOs' implementation**:
   This layer implements the DAOs addressed in the previous layer, through providing access to each of the defined entities at the entity layer, e.g. access to the organization entity. We have implemented this layer using some web services developed on top of the GloNet

platform. Moreover, The DAO implementation provides Hibernate-based access to the MySql database through the cloud, e.g. to access the product/service and feature specifications.

4. **Controllers:**
   This layer handles the interactions between the users (human or software) and the sub-system, through two separate sets of controller components: (i) the web service interactions (i.e. by software) and (ii) the web interface interactions (i.e. by human).

5. **Views**:
   The Views layer represents a set of different interfaces on the exchanged data/information between the PSS and its users, which are either software system or human. This layer benefits from some technologies and standards for the user interface of the PSS including the HTML, JSON, AJAX, JQuery and XML. While HTML is used for the human interactions, XML and JSON are used for interaction with software systems. The AJAX technology and the JQuery framework are deployed to produce a smooth and interactive experience for the user.

Please note that [7] provides detailed description of the role played by PRODCOM codes in relation to product/service classes in PSS.

Also the integration of PSS with product portfolio as addressed in the next section, has required the representation of the relation between the Request entity and the name associated with the request in product portfolio.

More details about the implementation architecture of this sub-system and its main components, which are presented in figure 7, are described already in the deliverable [6].

2.2 Data Model

There are two main types of data entities in the PSS sub-system of GloNet, including the Products and Services of complex products. The data model for the products is already described in details in [5] and [6]. Figure 4 represents an entity–relationship model (an ER diagram) [12] as the data model for describing the different aspects of the business services in the PSS.

This ER diagram shows the relationships among meta-data elements for service specification tool (SST), and their cardinalities. It represents the relation between the enhancing services of sub-products and the other entities defined in the PSS according to the Entities layer of its general architecture (figure 3).
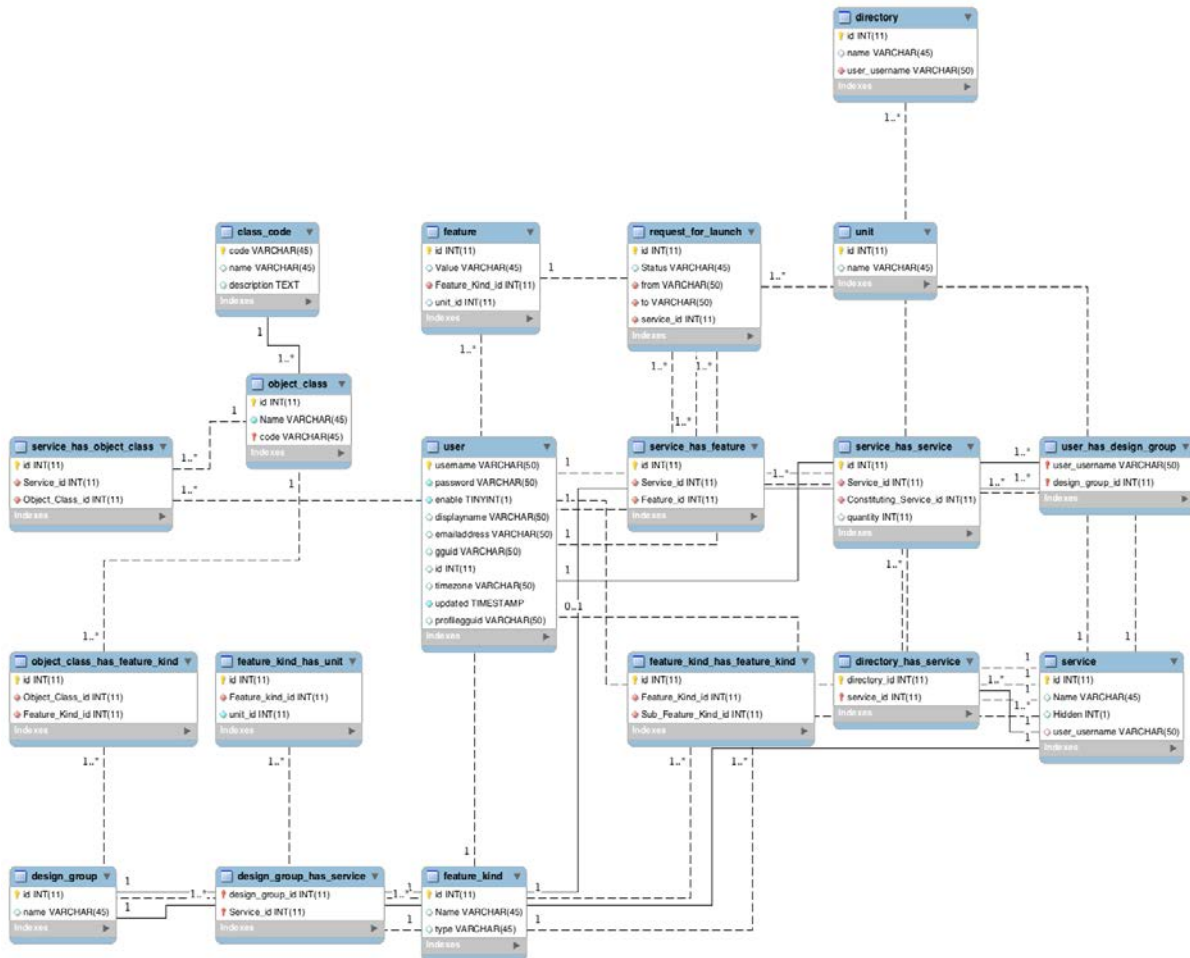
Figure 4- The ER diagram of the services in the PSS sub-system

2.3 Integration of PSS with the other sub-systems of GloNet

The PSS sub-system is integrated with other sub-systems of the GloNet system. Below is a brief description of this integration process for the SST and PST parts of the PSS with other GloNet sub-systems. Figure 5 shows the integration links between the PSS and other sub-systems of the GloNet.

i.   Integration with GloNet platform: The PSS sub-systems connect to the GloNet platform for accessing to the list of VBE members and to authenticate the users of PSS, through its single sign on feature. Furthermore, during the service specification process, the SST tool gains access to **Process Descriptions** (Workflows) defined by the user related to specification of a business service, which are captured in the platform. These workflows are designed using the BPMN Modeler software, and uploaded in the platform.

ii. Integration with Product Portfolio: The PSS creates a Product Portfolio instance in the *Product Portfolio system*, and assign a customer to it. Finally, the PSS stores an instance of the product or service specifications in the *Product Portfolio*.

iii. Integration with VO formation: The PSS can send a request for launching a VO, which starts the configuration/formation process of a VO corresponding to the specified service or product.
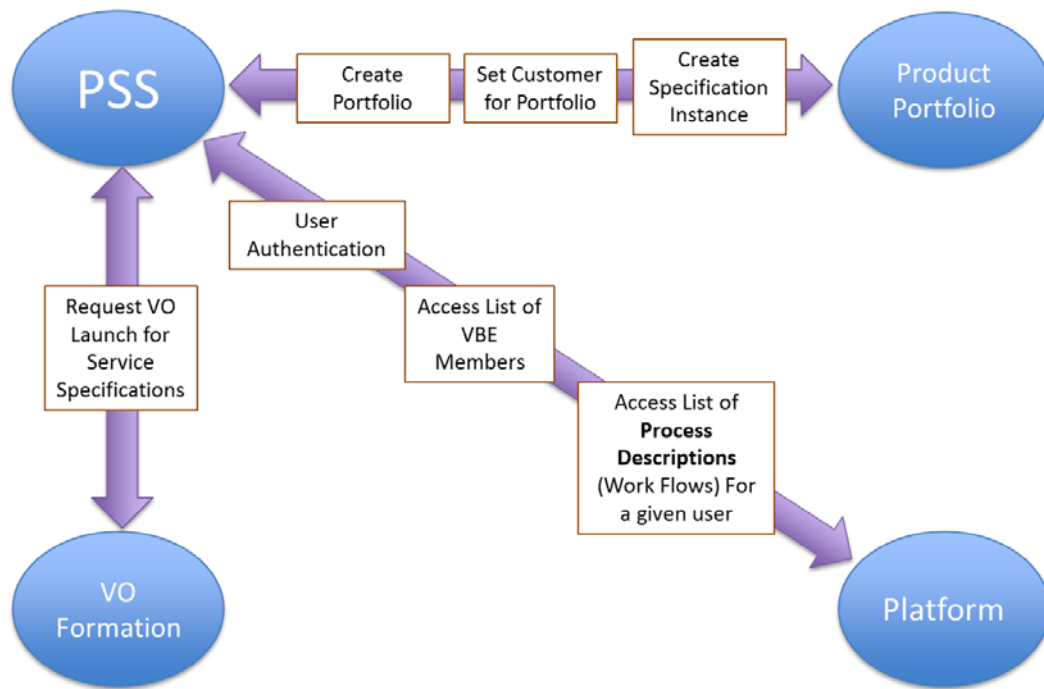
Figure 5- The integration of the PSS sub-system with the other components of the GloNet system

3   PSS – USING PRODUCT/SERVICE SPECIFICATION SUB-SYSTEM
– Focused on SST and PSDR tools

This section provides some *examples and snapshots* from running the service specification tool (SST), and the Product/Service Discovery and Recommendation (PSDR) while illustrating the final interfaces developed for use of these tools. The development has addressed specifically the functional/non-functional requirements addressed in details in [7]. As such, specific emphasis has been put on developing the needed data manipulation operations, i.e. Add, View, and Duplicate, on: services, classes, feature-kinds, and units.

Figure 6 shows the login form of SST (and other PSS tools) to authorize and authenticate users, which can be used only when the tool is used as a standalone tool, outside the GloNet. However, in case of login through the GloNet platform, the GloNet system authenticates the user, and thus the PSS sub-system can receive the single-sign on token provided through the GloNet platform. As such, in those cases, this login step will be bypassed.
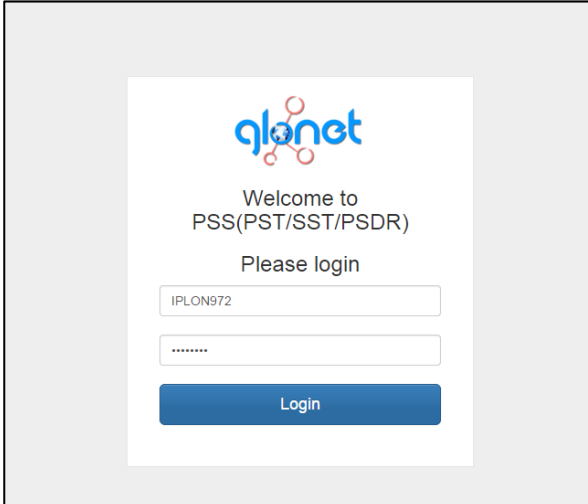
Figure 6 – Login form of the tools (only when used standalone)

Figures 7 show the first view of the PSS, which provides access to PST, SST, and PSDR tools, after user has logged in. Below is a screenshot of the general view (header and menus) of the PSS sub-system, as also presented before in [6].

Figure 7 – General view of PSS

- On the top right, the name of the user (who is logged in) is indicated. There is a sub-menu developed under the user name to view the *user profiles, system status* and finally a link to *log out* from the system. The user name is next to the name of the *Design Group* in which the user is working. Definition of the *Design Group* assists the user to better organize his/her design specifications, and provides the possibility of switching between different projects. Moreover, his/her partners in the design groups can contribute to the specification of the corresponding services and products.

  Please note that when the *Design Group* is set to Private (as in Fig. 7), the user works only on his/her private space, without any other partner. As figure 8 shows, the user can also indicate *another Design Group in the sub-menu, involving other partners in* product/service specifications.

  The *Design Group* name appears next to the name of *My Directory*. This option provides the possibility to assign different users the needed access right to the defined service specifications. This is mainly used to assist the user with organizing his/her own service specification folders/directories.
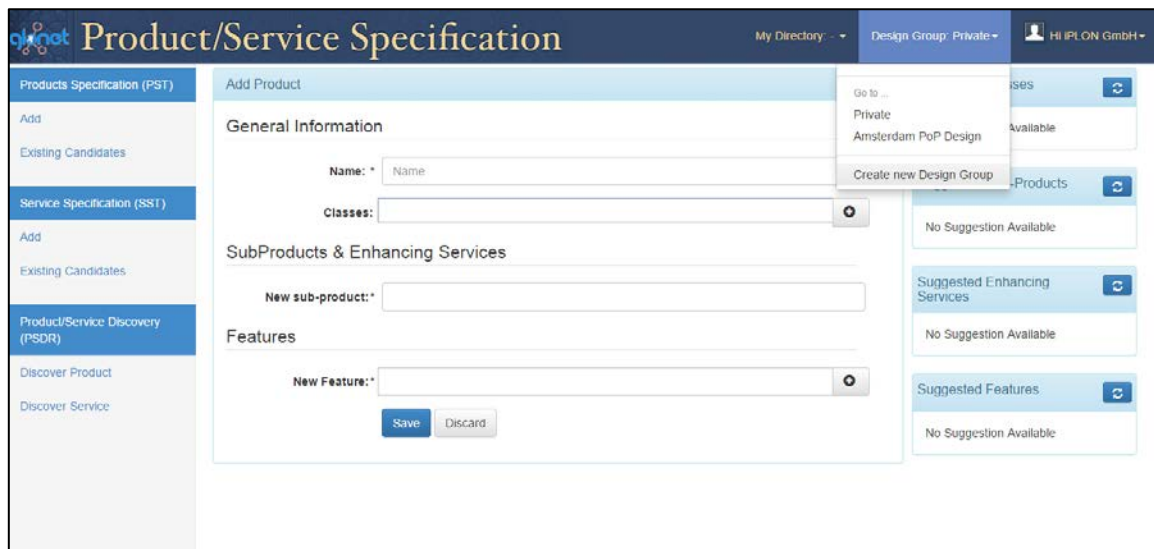
Figure 8 – Design Group menu

- On the left side, three tools (PST, SST and PSDR) developed in the PSS sub-system of GloNet, and their main functionalities are represented, each opening a set of menu items. These give access to different functionalities related to *specification of sub-products (PST), specification of services (SST)* and *Discovery of products and services (PSDR),* which are described further below and in the following sub-sections.

- In the center, a data entry space for the tools (SST, PST, or PSDR) is represented according to the called functionality, selected from the left side.

- On the right side, a number of suggestions may be represented, to help the user with the specification processes. These include suggestions of related classes, suitable sub-products, recommended enhancing services for sub-products, and suggested features, that can be used by the designer in his/her design. These suggestions are described further in the following sections.

4   SST AND PSDR TOOLS

Under the *service Specifications tool (SST)* menu, there are two sub-menus: *Add*, and *Existing candidates*. The user, through the *Existing candidates* link, is able to view the *Existing service Specifications*. The list of specified services is provided with their *names, classes*, and a drop-down list of icons (on their right) representing different *actions* that can be performed on them, as described later. Here we first describe different features supported by the SST and PSDR in the remaining of this section, and further illustrate their functionalities with some screenshots of these developed tools.

4.1 Adding a new service specification

Figure 9 shows the snapshot of the interface window for *SST,* where the *Add* sub-menu is selected from the menu under the *service Specifications (SST)*. This window is used to support specifying a new atomic and/or composite business service. As such, this interface can be used to specify/register/add a new business service at any level of abstraction and granularity.

To simplify the labeling of presentations in this section, this interface window is called "New Service" form. In the New Service window, the user can add an atomic/composite service specification by first providing a unique name for the service. The user can then optionally define one or more classes for the specified service.

Once the user, who is specifying a service, defines a class for it (e.g. the class *"Atomic Service"*) through the system all the features defined for that specific class will automatically pop up in this window. As such, the user is assisted with receiving the names of all features, which he/she should fill and specify as a part of the specification of that service. Furthermore, while providing input for some of those features might be optional, some other features of the class might be mandatory, which would then oblige the user to provide the needed input.

For example, in Figure 9 since the user has added the class "Atomic Service", for the service which he/she is specifying, a set of feature-kinds for this service have automatically popped up in the center of the *New Service specification* form (figure 9), for its further specification. These include the mandatory set of feature-kinds, such as "Context", "capabilities", and "Response time", which show up automatically on the screen. It is important to note here that providing input value (i.e. features) for these feature-kinds are obligatory (as marked with "*"). Therefore the user is obliged to define the value and unit for these features.

There are several other aspects involved in specifying a service that can be defined through this interface. For instance, if a service is composed of other services (a so-called Composite service [13][14]), this fact can be indicated in its specification, under the *New constituent Services*. Once introduced, the names of constituting services also automatically appear in the specification of the composite service. Consider the example shown in Figure 10. Here, the user has introduced the three needed constituting atomic services for the "site maintenance" service, namely the "Check and Report", "Wildlife Prevention" and the "Water Drainage". Therefore, these three services also automatically appear (as in this figure) as a part of this service's specification.

Figure 9 - New Service specification form

Please also note that for each of the constituting atomic services, the needed quantity of that service also needs to be specified. As the example shows, the above composite service needs to deploy the "Check and Report" service twice, imagining that it is needed once before the analysis of the damage and once afterward. Therefore, the user has indicated "Check and Report" with the quantity 2, "Wildlife Prevention" with the quantity 1, and the "`Water Drainage" with quantity 1. But clearly more than the quantity, the proper specification of a composite service must accompany the specification of inter-connections among its constituting atomic services. For this purpose, a workflow or a BPMN diagram can be defined for the composite business service, representing the orchestration among its constituent services [7]. This aspect is supported by the integration between the PSS sub-system and the GloNet platform, as addressed in Section 2.3, and identified through the feature called Process Description in the composite service specification (see Figure 10). This feature is used to provide a reference to the desired workflow for composite business services. For this purpose, the service designer (current user) creates his/her intended workflow through the GloNet platform that provides a BPMN Modeler, and uploads it to the platform. The BPMN diagram for the

above composite business service example is presented in section C of Annex I, screenshot I.16 (guideline scenario). Then through the PSS subsystem, when specifying composite services, the system provides a selection menu including the list of existing workflows owned by the user, to select from. As shown in figure 10, the designer can then choose his/her desired workflow, the "Site Cleaning-BP", for the composite business service. Please also note the remove button option (black minus circle ⊖) indicates that the feature is optional, i.e. it can be removed. For instance, constituting services for the composite service, can be added and/or removed, e.g. see figure 10, and its constituting services.

Figure 10- New composite service form

Finally, if the user wishes to add any other new features to a service specification, the *New Feature* can be specified at the bottom of the screen through the *New Features* section of the interface. The user will then need to indicate the feature-kind to which the feature corresponds. The example presented in Figure 11, indicates that the user has entered "Area" as the feature-kind for a new feature, which he/she has defined.

The user must then specify both the value and the unit for that feature. It is important to note that based on the feature-kind that the user selects/identifies for a new feature, the data-types for its value and unit will differ, according to those that are defined for the corresponding feature-kind. However, if the mentioned feature-kind is not already defined in the system; the user will then be immediately prompted with the window that asks him/her to first create that feature kind, before going further with the definition of the newly introduced feature. This is also explained with more details later on, in this section. At the last stage of this service specification process, the specification can be saved or discarded, through the two buttons that appear at the end bottom of this window.
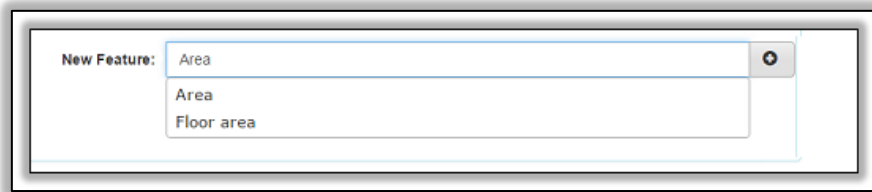
Figure 11- New Feature item pop op in the New service form of figure 10

4.2 Adding a new feature-kind

Every specified service through SST, is characterized by a set of features. Any identified feature in this sub-system requires that its feature-kind is defined priori to its use. To simplify users tasks, the user interface that supports definition/adding of new feature-kinds, as well as for enhancing the specification of already defined feature-kinds in this sub-system is supported through a pop-up window, as indicated in the "Create New Feature-Kind" form of Figure 12.

In order to add a new feature-kind, the user must first provide both a unique name for it as well as providing the domain or data type for the values of the features that instantiate this feature-kind. The user can further optionally indicate one or more possible units for the domain values specified for the feature-kind. Figure 12 shows the popup window of the c*reate new feature-kind form*, which can be triggered when defining the new service specification. At the last stage of the feature-kind specification process, the feature-kind can be either saved or discarded. When the definition of the feature-kind is saved, it gets automatically added to the missing field from the previous window that triggered it.

4.3 Adding a new unit

Within the *Create New Feature-kind* form, users can introduce a new unit (e.g. man-hour) for the features that instantiate a feature-kind, as indicated in Figure 12. The user is only required to enter a unique name for the new unit, and click on the corresponding plus sign ( ) for it, which appears in front of every unit input. These newly introduced units can then be associated for the features introduced through different feature-kinds. At the last stage of the unit specification process, the new introduced unit can be either saved or discarded.

Figure 12 – Create New Feature-kind form

4.4 Adding new classes

As discussed before, each service specification could be associated to one or more classes. As such, classes define the meta-data for services, and enable the Request for Launch process, which in turn indicate their set of feature-kinds that characterize them. To simplify users' tasks, the user interface that supports definition/adding of new classes is also through a pop-up window, as indicated in the "Create New Class" form in Figure 13. To add a new class, the user must provide a unique name for that class. The user can then also add a set of obligatory feature-kinds to be associated with this class. Since this window is a pop-up window, it is triggered through other interface windows, and every time after saving a new class, the new class is automatically added to the missing field from the previous window.

4.5 Viewing / managing existing product specifications

Once services are specified, they can be viewed by selecting the "Existing Specifications" item under the service specification menu. As such, depending on the selected Design Group (as indicated in the upper right corner of the screen), their associated existing specification window will appear, showing the list of all relevant existing service specifications (sorted by their names), which the user is authorized to view. In other words, the specifications that are included in this window are all those related to the specified Design Group.

In the example of Figure 14, the "iPLON" user has selected/indicated the "Amsterdam PoP" Design Group. Consequently in this example, only the restricted service specifications that belong to this Design Group are shown. Please note while "iPLON " might own only some of these restricted services, the other users in the design group might own some other specifications.
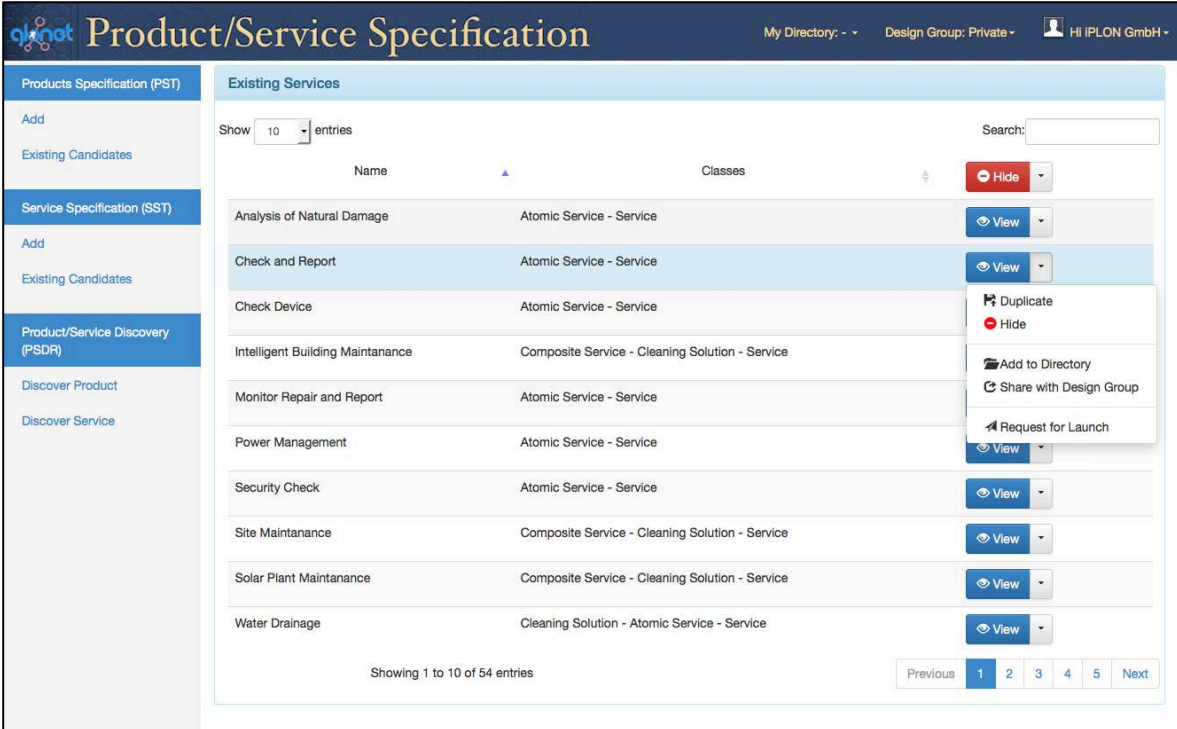
Figure 13- New Class form

Other than viewing the service specifications, authorized users can also manage these specifications by preforming the following set of actions:

- **View** (Shown with the icon 👁) action, which takes the user to the view details of the service specification. Figure 15 shows an example of this view for the specification of "Analysis Natural Damage" as an atomic service.

- **Duplication** (Shown with the icon 📑) action, which takes the user directly to a pre-filled "New Service" form. This simplifies the task of users, since in that form the specification information about the selected service is duplicated, which can then be edited by the user, thus defining a new similar service specification.

- **Hide** (Shown with the icon ⛔) action, which allows hiding the corresponding service specification from the specific existing services window, which is restricted for the *Design Group*. For instance the user finds a service useless for him/her use and so the user hides this specification from his/her view.

- **Add to Directory** (Shown with the icon 📁) action which provides the possibility to assign an already defined service specification to which user has access, to an existing directory of the user. This is mainly to assist the user with organizing his/her service specification folders. This means that by default when specifying a service, if the user has not indicated a directory on the top right corner of the screen, the specification does not get assigned to any specific directory. However, when a directory is indicated on top right corner while specifying a service, then that service will be allocated to that directory. Nevertheless, through this action which is provided in the Existing Specification form, services may be assigned and reassigned to different directories.

- **Share with Design Group** (Shown with the icon ⮌) action, which provides the user with the option to change the access rights/sharing status of a certain service specification that

he/she owns. The share options are available through existing services window, when the user clicks on its icon. Please note that when defining a new service specification, the access right to that specification is made private by default, that is if the user has not indicated a *Design Group* on the top right corner of the screen, otherwise the specification will become restricted to that *Design Group* by default. At any point in time, the owner of the service specification is allowed to broaden the access to that specification. This means that if a specification is private, then the owner can change it to restricted within a *Design Group*. But service specifications that are for instance restricted to a *Design Group*, their access cannot be reduced to private. In other words once the owner of a service specification grants certain access rights to others (to view the specification) he/she cannot withdraw that right later.

- **Request for Launch** (Shown with the icon ⚔) action, which enable the user to issue a request for launch of a service specification. This option is further explained in the next sub-section with more details.

Figure 14 – Existing Service Specifications Window (restricted for the *Design Group*)

4.6 Sending a Launch Request

When a designer completes the process of specifying an atomic/composite service (at any level of granularity), he/she may wish to initialize the process of realizing that service. For this purpose the SST supports the functionality of *Launch Request* (also referred to in this document in short as *request*) for starting this service realization process. As such, when and if a designer wishes that a certain service specification should be realized, he/she can announce this fact through building a request for it. For instance the designer of a service (check and report – see Figure 14), after specifying all its details, can build/make a launch request for this service, while providing its specification. In fact designers of innovative services are interested and curious to check if their

specifications can in fact be realized, meaning that they wish to know if their specifications are constructible and can materialize, and by which potential consortium of companies can it be done.

In order to build a launch request, the user can use the interface developed by the SST, as indicated in Figure 14, and labeled as the "Request for Launch" in the *Existing Services* form. Using this option, the user can send a new launch request for one of the already specified services. Please note that before a new Launch Request can be built for a service specification, the service must be first properly specified. Then through the interface presented in the *Launch Request* form, the user will identify a specified service (i.e. indicates its specification), to be included in the package for this request of launch.

Figure 15 – View form of a Service Specifications

Besides the identification of services, when defining and creating a launch request, also called packaging product/service specifications, it must be clear for which customer this design has been made. Moreover, the name of the complex product, for which this service is being launched, must be indicated so that the specification can be stored property in the product portfolio.

In Figure 16, a service specification (e.g. "Check and Report") is requested to be launched. The value of the "by" field of the package for this request would be indicated based on the *user-name* who is logged into the system (e.g. "iPLON"). The value of the "for" field of the request, is the customer name that is "City of Amsterdam" in this example. Finally, this specification later would be stored in the portfolio that belongs to the "Amsterdam Power Plant".

Figure 16- Launch Request form

At the last stage of the launch request process, the request definition can be saved through the button at the bottom of the form.

4.7 PSDR- Discovering of Services

The menu item on the left column for *Discover Service* is located under the Product/Service Discovery (PSDR) menu item. It opens a form to discover services and to rank the matched suggestions based on the user query. This constitutes a part of the product/service discovery and recommendation engine of the general GloNet architecture. In fact, this part of the tool addresses our mechanisms for discovering and matchmaking between the users' required criteria and the existing service specifications, in order to support the service designers with offering them the best-matched business services. As we described in [7], the ranking is done according to the similarity score that expresses approximate bi-simulation [15] between registered specification of each service and the users-submitted query. The discovery of most-fitting services can be done based on all the service specification features, including the service's syntax, semantics, behavior and quality criteria aspects.

Figure 17 shows a screenshot of the *Service Discovery* form, where the user can select some of the service features as the criteria of the search, and then set his/her desired values for them.

Figure 17 – Service Discovery form

Figure 18 shows the result window of the matched services for the example query, which is represented in figure 17. The results are ranked by the calculated similarity scores of the registered services. The mechanism of the search and ranking suggestions is described in [7].

Figure 18 – An example of Service Result window

4.8 PSDR- Discovering of products

The link to *Discover Product*, which is located under Product/Service Discovery (PSDR) menu, opens a form to Discover Products and Rank of matched suggestions based on a user query. It constitutes as a part of the product / service discovery and recommendation engine of the general architecture. As we have already described in [7], the ranking is done based on the

following three similarity methods: Standard-based recommendation, Specification-based recommendation, and User/Usage profile-based Recommendation.

Figure 19 shows a screenshot of the *Product Discovery* form, where the user can select some of the product features as the criteria of the search, and also indicate his/her desired values for them.

Figure 19 – Product Discovery form

Figure 20 shows the result window of the matched products for the example query, which is represented in figure 19. The mechanism of the search and ranking suggestions is described in [7].

Figure 20 – An example of Product Result window

5   CONCLUDING REMARKS

.27

This report describes the prototype developed for the service specification Tool (SST) and Product/Service Discovery and Recommendation Engine (PSDR). It takes as the main input the specified design requirements and the introduced mechanisms for the dynamically customizable services enhancing complex products, as addressed in [7].

The advantage of the currently developed prototype reported in this document is that it is both generic and reusable for different domains (other VBEs) and not limited to the solar power plants and intelligent buildings domains. Moreover, these two tools (SST and PSDR) as well as the product specification tool (PST) have been integrated within the Product/Service Specification (PSS) sub-system of GloNet. The PSS sub-system is further integrated with the GloNet platform, the product portfolio, the VBE management, and the VO formation sub-systems.

Finally the developed PSS sub-system also serves as the base for other developments in WP5, namely for providing and packaging the needed input for the functionality as required for consortium formation and operation support.

6   REFERENCES

1. GloNet D1.1 deliverable – Detailed Requirements for GloNet use case and Domain Glossary
2. GloNet D1.2 deliverable – Detailed Requirements for GloNet use case and Domain Glossary
3. GloNet D2.1 deliverable – Required Information/Knowledge Provision Services Specification
4. GloNet D2.4 deliverable - Mechanisms for defining composed services to support collaboration
5. GloNet D4.1 deliverable - Design report on approach and mechanism for effective customized complex product specification
6. GloNet D4.2 deliverable - Prototype of Services supporting iterative complex product specification
7. GloNet D4.3 deliverable - Report on dynamically customizable services enhancing products
8. Afsarmanesh, H., & Shafahi, M. (2013). Specification and Configuration of Customized Complex Products. In Collaborative Systems for Reindustrialization ,pp. 81-90. Springer Berlin Heidelberg.
9. The Spring Framework http://www.spring.io
10. Bauer, Ch., and Gavin K. (2005). Hibernate in action..
11. MySQL http://www.mysql.com
12. Chen, P. P. S. (1976). The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems (TODS), no. 1, pp. 9-36.
13. Shafahi, M., Afsarmanesh, H., & Sargolzaei, M. (2014). A Coopetition Space for Complex Product Specification. In Collaborative Systems for Smart Networked Environments, pp. 83-97. Springer Berlin Heidelberg.
14. Afsarmanesh, H., Sargolzaei, M., & Shadi, M. (2014). Semi-automated software service integration in virtual organisations. International Journal of Enterprise Information Systems, (ahead-of-print), pp. 1-28. Taylor & Francis.
15. Sargolzaei, M., Santini, F., Arbab, F., & Afsarmanesh, H. (2013). A tool for behaviour-based discovery of approximately matching web services. In Software Engineering and Formal Methods, pp. 152-166. Springer Berlin Heidelberg.

ANNEX I- AN EXAMPLE SCENARIO AS A GUIDELINE FOR THE PSS

A few notes upfront:

As a guideline for the usage of the PSS sub-system, this document briefly addresses a demonstration plan for version 3 of the PSS. Further to the guideline of PST demonstration in Deliverable 4.2, this Final version has some extra functions that are related to SST and PSDR.

Also, in this demonstration we have over simplified the specification process at each stage due to time constraints. Please note that at each stage specifications are made gradually and threw an iterative process.

Please also not that PST, SST and PSDR are equipped with "smart" autocomplete/suggestions for input fields.

A. Amsterdam solar power plant – *PRE-PLC Stage*

1. An EPC (iPLON) has received a tender for constructing a solar power plant in Amsterdam

2. In order to prepare bid, iPLON should first select its initial partners for the project for which it needs to evaluate and select the needed technologies.

3. This can only be achieved after a rough specification is made and the critical features of the complex product are identified.

4. So, iPLON opens the "add product" form by clicking on the add option in product specification (Click) (see screenshot I.1).

    Screenshot I.1
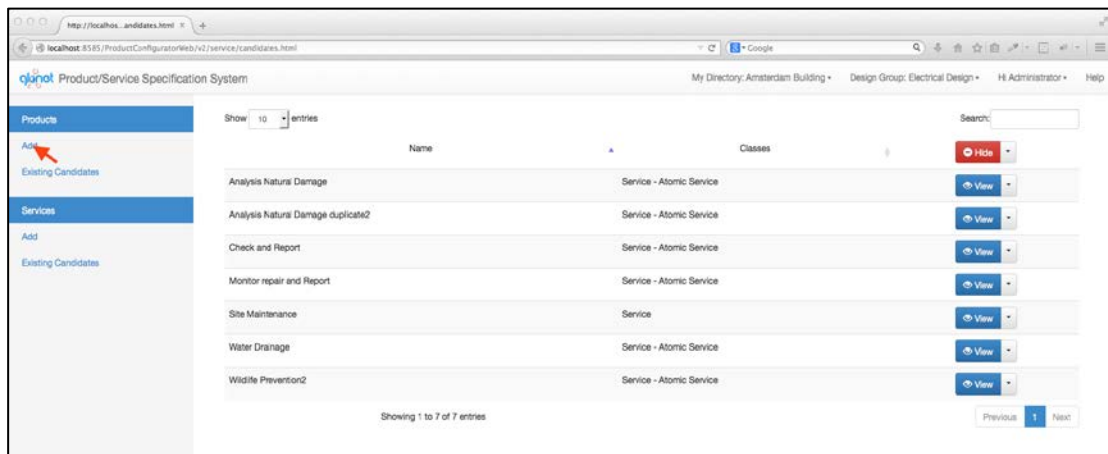
5. The first step for iPLON is to identify which high level classes of products this complex product falls under.

6. Let's imagine that iPLON notices that a class (Power Plant) that is needed is not defined in the system. iPLON requests to add the new class by clicking on the plus button in front of the class field (see screenshot I.2).

Screenshot I.2
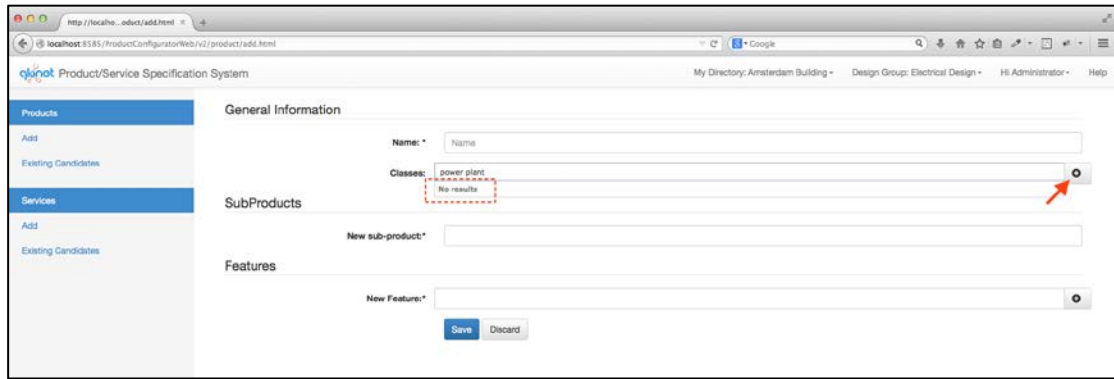
7. This opens up a window for iPLON to specify a new class.

8. iPLON introduces a name for the class (i.e. Power Plant) and specifies the possible corresponding PROCOM code (power or C27.11.43) for the class. It can also specify a number of required features (Power Capacity) for the class and clicks add to save (see screenshot I.3 and I.4).

Screenshot I.3
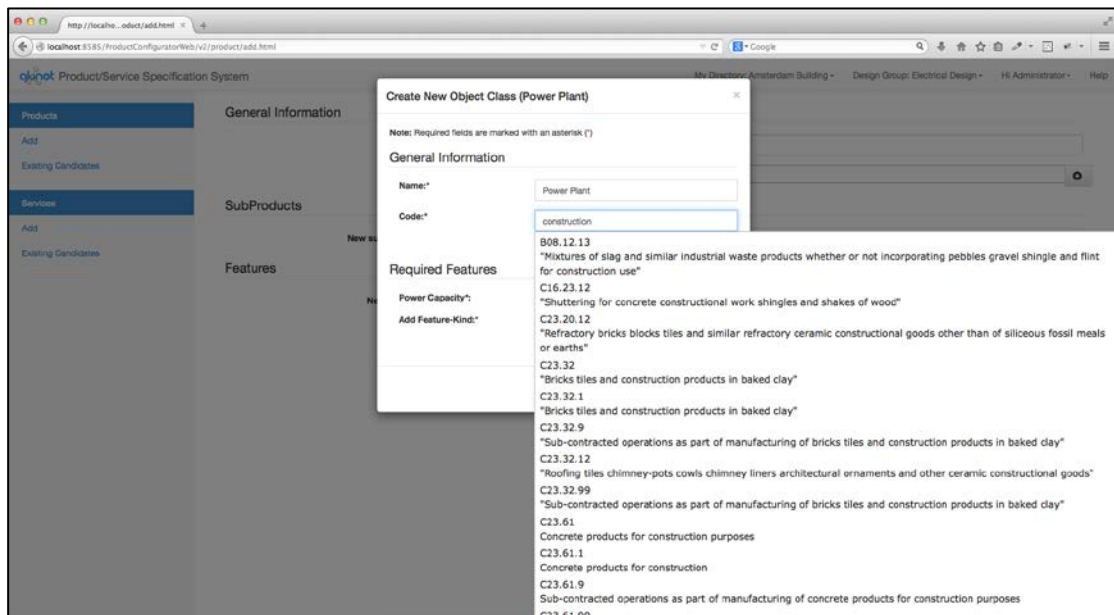
Screenshot I.4
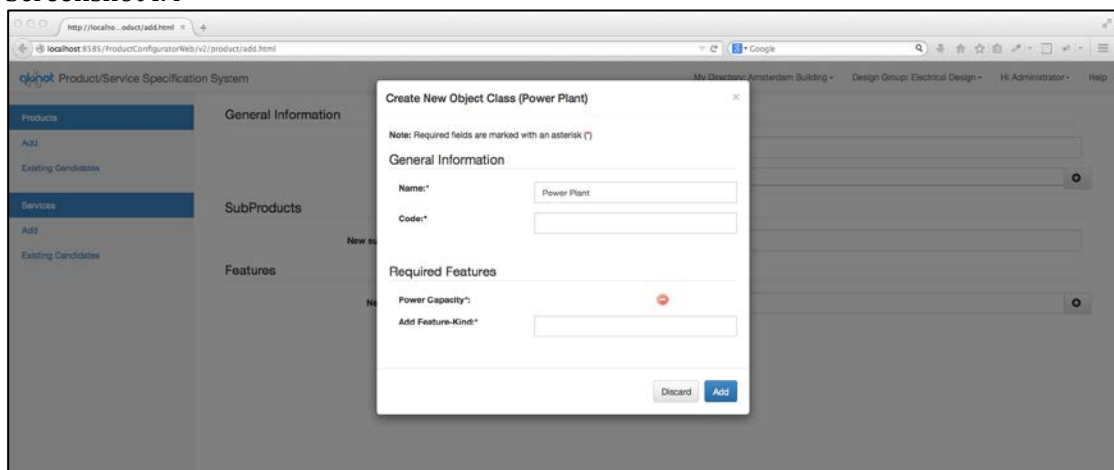
9. After saving, iPLON is guided back to the "add product" form so that it can continue with the specification.

10. Here, the just added class is introduced. iPLON can choose to introduce more classes or even removed the just specified and introduced class(Power Plant).

11. Each introduced class causes the system to ask for a set of obligatory features. So iPLON must fill in the value and units of the required features.

12. It's important to note that the fields for the features are restricted to pre-defined conditions (e.g. pre-defined units).

13. iPLON can also specify features that are not mandated by the classes by introducing them using the new feature filed (see screenshot I.5 and I.6).

Screenshot I.5

Screenshot I.6

14. Let's assume that at this point, iPLON identifies the need for a feature-kind that has not yet been introduced within the PST system (so no help is provided by autocomplete and a result it is not shown in the dropdown menu) it can add one by clicking the + button (see screenshot I.7).

Screenshot I.7

15. This opens dialog for iPLON to specify the feature-kind, iPLON can specify the name, the type of the feature-kind and the possible units for the feature-kind (see screenshot I.8).

Screenshot I.8

16. After clicking add, the system returns iPLON to the "add product" form again and this new feature-kind is introduced for the product specification.\

17. At the end iPLON specifies the value and units of all features and clicks save.

Screenshot I.9

18. At this point iPLON selects existing specifications in products and using the dropdown option beside the view button for a given specification finds more advanced functions that can be performed for the specification.

19. Here iPLON, clicks the "Request for launch" button to request a VO formation for the specified specification (see screenshot I.9 and I.10).

Screenshot I.10
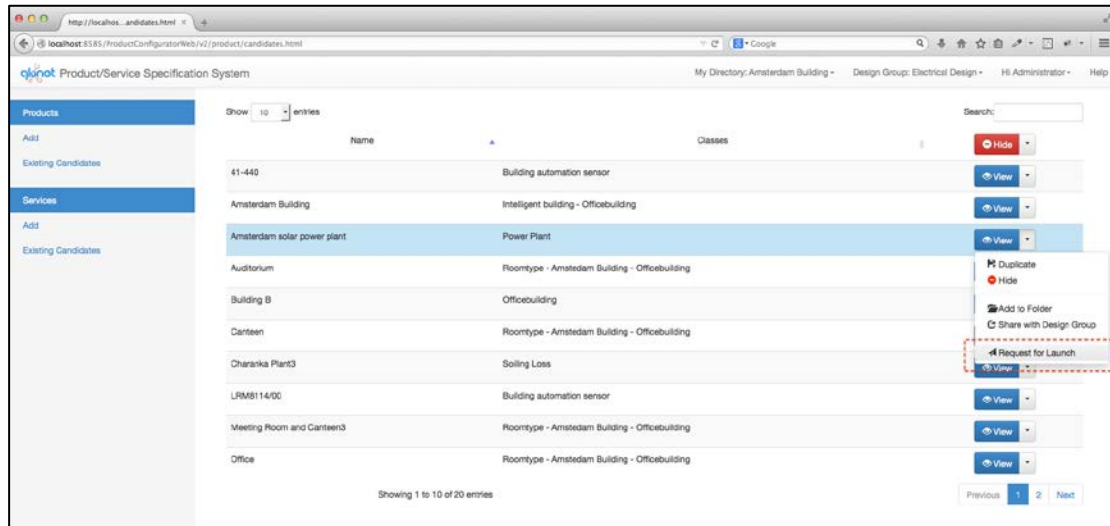
B. Amsterdam solar power plant – *Design Stage*

1. After the EPC (iPLON) has been accepted for constructing a solar power plant in Amsterdam
2. In preparation for constructing the power plant iPLON (with the help of its initial partners) should first make the detailed design/specification of the power plant in order to select partners for the construction of the power plant.
3. iPLON starts this process by creating a design group, sharing the rough specification of the Amsterdam power plant and inviting the initial partners to help in the process of specifying the detailed specification.
4. All members of the Initial VO (Amsterdam Power Plant-Bid) constitute the members of a design group (called Amsterdam PP design)
5. At this stage some of the initial partners might ask other partners to join the **design group**(s) in order to contribute to the detailed specification.
6. If there is time I can show you how members are added later
7. iPLON Shares the initial designs with the Design group
8. ***Please not that any user can also create a new design group independent of the VO in specific cases.***
9. After being part of the design group, each partner can view the design group he is a member of and can specify detailed specifications of sub-products and services required for the complex product (i.e. Amsterdam power plant) by reviewing the initial specification.
10. For example Prolon re-uses and customizes an existing specification of a sub-products (Pyranometer) for the Amsterdam power plant by duplicating it and shares it with the design group.

11. This also applies to services for example SKILL can specify a business service (wildlife prevention).

12. By clicking the add service option in service specification Skill can specifies an atomic business service (see screenshot I.11).

Screenshot I.11

13. After all partners have specified the sub-products and services, iPLON duplicates the previously defined complex product (see screenshot I.12).
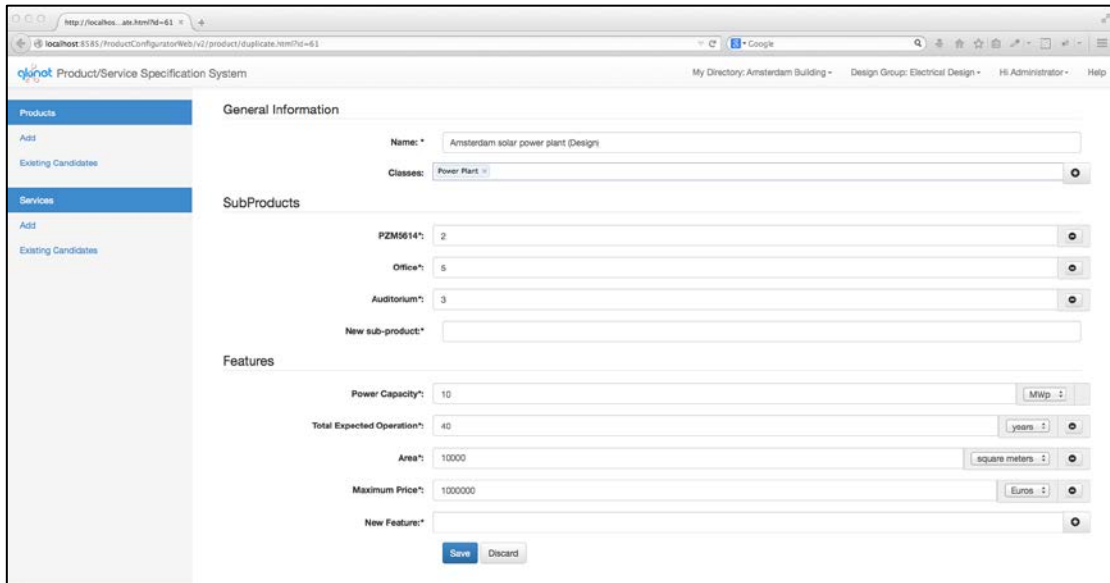
Screenshot I.12

14. iPLON adds the newly defined/excising sub-produces and some features to the duplicate and renames it (see screenshot I.13 and I.14).

Screenshot I.13

Screenshot I.14
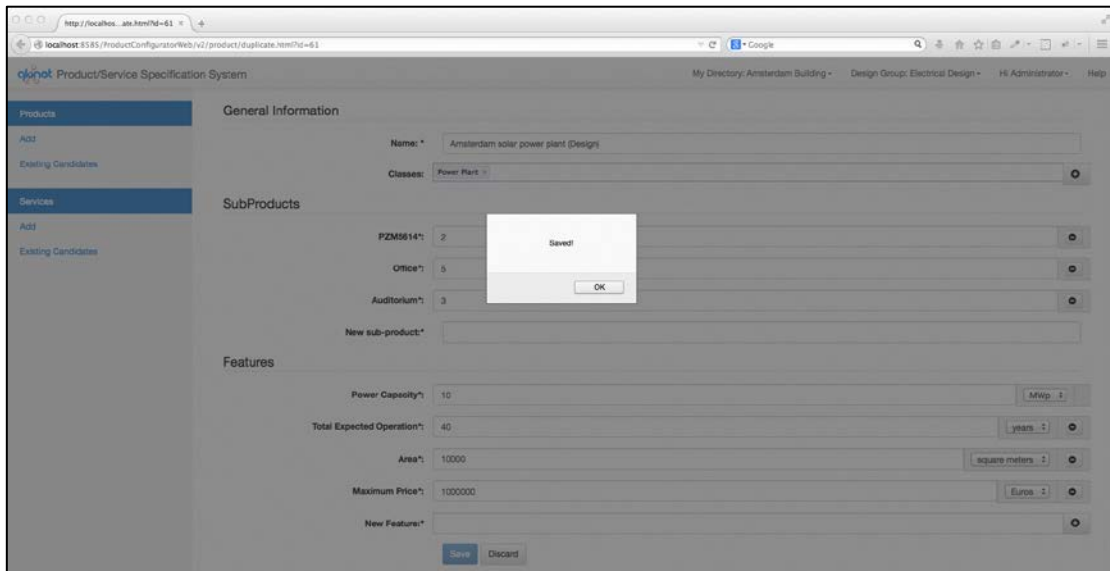
15. iPLON request a launch of the complex product (see screenshot I.15).

Screenshot I.15
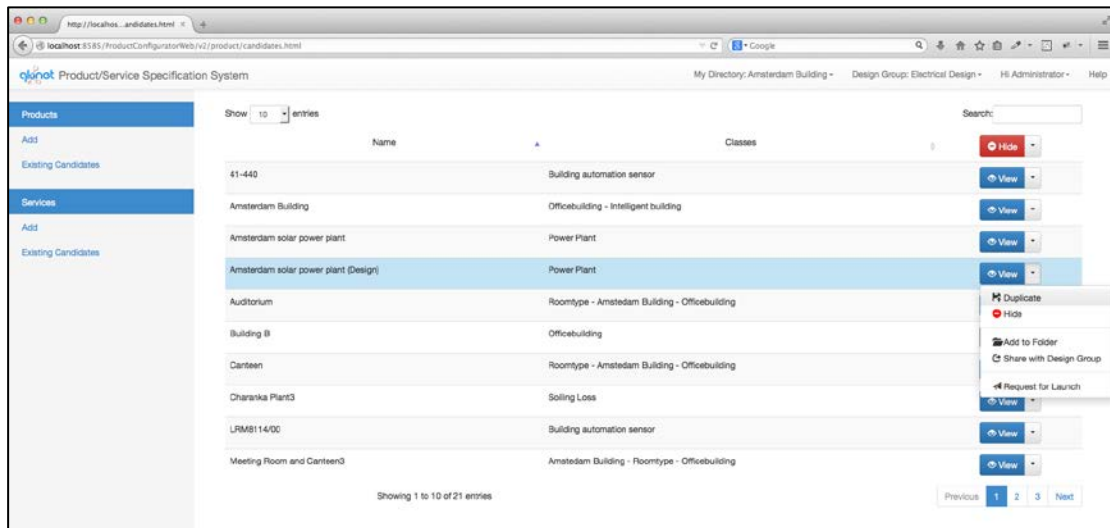
C. **Amsterdam solar plant** – *O & M Stage*

1. A partner involved in the power plant (SKILL) identifies the need for a new composed business service to support the power plant (Site maintenance).

2. In order to achieve this SKILL identifies the existing services it can use to implement the composite business service.

3. It first designs the process description (Workflow) using the GloNet BPMN modeler and uploads the workflow to the platform (see screenshot I.16).

Screenshot I.16
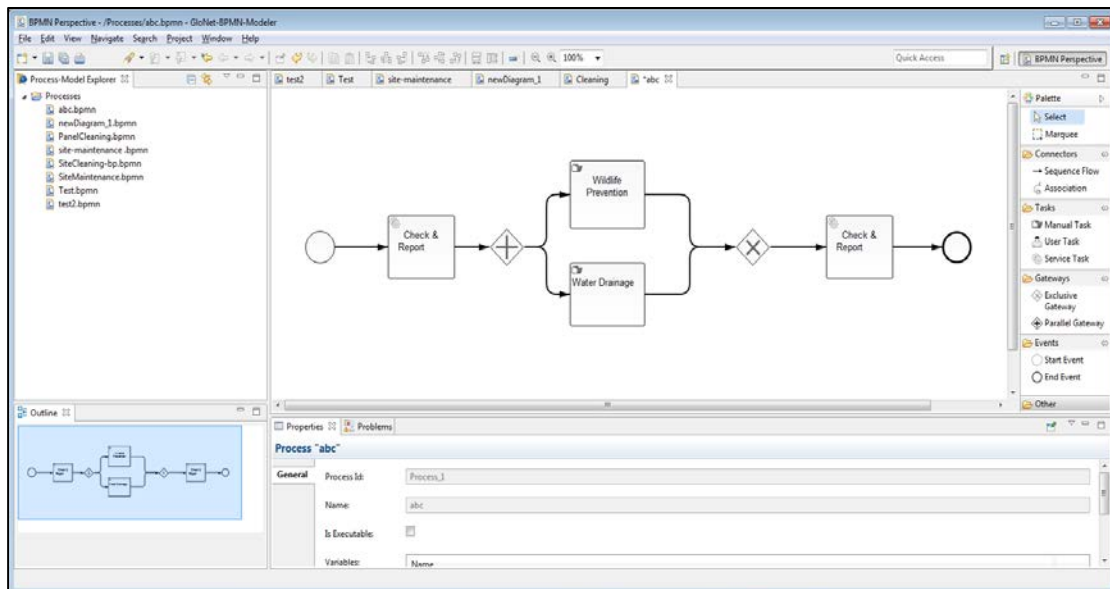
4. Then it specifies the composite business service using the add service function.

5. Here it specifies a name for the service and adds the class "composite service" and other classes he finds relevant.

6. Then he adds the constituting services to the specification. And fills in the required features for composite services. One of these is business process, in this feature he selects the workflow he has uploaded previously from the menu (see screenshot I.17).

Screenshot I.17

7. Finally after registering the service specification, it chooses the existing specifications and launches the service (see screenshot I.18).

Screenshot I.18

CONSORTIUM

.38

CAS Software AG, Germany

Project coordinator: Dr. Bernhard Koelmel

UNINOVA – Instituto de Desenvolvimento de Novas Tecnologias, Portugal

Technical coordinator: Prof. Luis M. Camarinha-Matos

Universiteit van Amsterdam, Netherlands

iPLON GmbH The Infranet Company, Germany

Steinbeis GmbH & Co., Germany

SKILL Estrategia S.L., Spain

Komix s.r.o., Czech Republic

Prolon Control Systems, Denmark

Member of the:

www.glonet-fines.eu