D4.2
Prototype of services supporting iterative complex product specification

Edited by

UvA
February 2014

Project funded by the European Commission under the ICT – Factories of the Future Programme

Contract Start Date: 1 Sep 2011
Duration: 36 months
Deliverable data

<table>
<thead>
<tr>
<th>Deliverable no &amp; name</th>
<th>D4.2 – Prototype of Services supporting iterative complex product specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Contributors</td>
<td>UvA: Mohammad Shafahi, Hamideh Afsarmanesh, Mahdi Sargolzaei, Ozgul Unal</td>
</tr>
<tr>
<td>Other Contributors</td>
<td></td>
</tr>
<tr>
<td>Internal Reviews</td>
<td>Luis Camarinha-Matos (Uninova), Thomas Maltesen (PROLON)</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>Public</td>
</tr>
<tr>
<td>Date</td>
<td>February 2014</td>
</tr>
<tr>
<td>Status</td>
<td>Final</td>
</tr>
</tbody>
</table>

Deliverable summary

This deliverable addresses the prototype that implements the design presented in D4.1 deliverable. The Complex Product Specification System (in short product specification system) constitutes one main sub-system developed within the GLONET, to support proper specification of different aspects of the complex products.

Specifically, this deliverable addresses the implementation aspects related to iterative specification of complex products. While the complete design of functionality for this system is provided in deliverable D4.1, we also address some enhancement over that design in this deliverable. The main implemented features and functionalities of the system are described. Also, a set of examples of uses, with screenshots from the developed subsystem are provided.

D4.1 and D4.2 serve as the base for the work in tasks T4.3 and T4.4 of WP4. Furthermore, these four WP4 deliverables provide one important input needed in WP5, for creation of those VOs that will be formed around developing and provision of different equipment and devices, needed for complex products.
TABLE OF CONTENTS

PROJECT-RELATED SUMMARY .......................................................................................................... 4

1 INTRODUCTION ............................................................................................................................... 6

2 REALIZATION OF NON-FUNCTIONAL REQUIREMENTS FOR PRODUCT SPECIFICATION SYSTEM .................................................................................................................. 10

  2.1 Security ................................................................................................................................................... 10
  2.1.1 Authentication .................................................................................................................................. 10
  2.1.2 Authorization .................................................................................................................................... 11
  2.2 Scalability ................................................................................................................................................ 12
  2.3 Portability ............................................................................................................................................... 13

3 REALIZATION OF FUNCTIONAL REQUIREMENTS FOR PRODUCT SPECIFICATION SYSTEM .................................................................................................................................................................................. 14

  3.1 Granular and customized specification of complex products ............................................................ 15
  3.2 Capturing product specification perspectives using classification ......................................................... 16
  3.3 Sub-product re-specification .................................................................................................................. 16
  3.4 Complex product/sub-product Launch Request..................................................................................... 17

4 IMPLEMENTATION APPROACH – GENERAL SYSTEM DESCRIPTION ................................... 18

  4.1 The Entities Layer ................................................................................................................................... 19
  4.2 The DAOs (Data Access Object) Layer .................................................................................................... 21
  4.3 The DAO Implementation Layer ............................................................................................................. 21
  4.4 The Controllers Layer ...................................................................................................................................... 21
  4.5 The Views Layer ...................................................................................................................................... 21

5 EXAMPLES OF USING PRODUCT SPECIFICATION SYSTEM ................................................... 23

  5.1 Adding a new product specification ....................................................................................................... 26
  5.2 Adding a new Launch Request ............................................................................................................... 28
  5.3 Adding a new feature-kind ..................................................................................................................... 30
  5.4 Adding a new unit ................................................................................................................................... 31
  5.5 Adding new classes ................................................................................................................................... 31
  5.6 Viewing / managing existing product specifications .............................................................................. 32

6 CONCLUDING REMARKS ............................................................................................................. 35

7 REFERENCES ................................................................................................................................ 36

ANNEX A – INTRODUCTION TO USING PRODUCT SPECIFICATION SYSTEM ..................... 37
**PROJECT-RELATED SUMMARY**

This deliverable (D4.2) is the second deliverable of WP4, and plays a main role both for the next deliverables of the workpackage, as well as for interconnecting WP4 to other workpackages, e.g. WP5. D4.2 is produced within Task 4.2, and represents the implementation of several base functionalities that support the specification of a complex product by its designer(s).

Being complex, the specification of the so called complex products in GLONET, e.g. the solar power plants and intelligent buildings, is typically not performed at one time during one session, rather it is done iteratively during several sessions and potentially involving a number of different stakeholders, varying from the equipment manufacturers, to the suppliers, or the EPC members, who are experts in specification of different components related to such complex products.

Our findings and design results reported in deliverable D4.1 (“Design report on approach and mechanism for effective customized complex product specification”) are used as the main inputs for this deliverable. As such, this deliverable addresses only the Prototype developed for the services designed in D4.1, which support iterative complex product specification. Although, when and if necessary, we also revisit some design aspects that have evolved since the preparation of D4.1.

Later on, task T4.3 will address the next main steps in complex product specification. Namely, the designs described in D4.1 and the implementation addressed in D4.2, will be further enhanced with some intelligent tools. These tools assist designers with some automated recommendation of equipment and services for their complex product specification. The approach that will be introduced in T4.3 therefore aims at automating certain predication of the needed sub-products, and recommending equipment and services, which best fulfill the complex product requirements.

The following figure summarizes different tasks and deliverables in WP4.
Considering that the system implemented in T4.2, will be further extended within the next tasks of WP4 (namely: tasks T4.3 and T4.4), the deliverable D4.2 provides the implementation of the foundation, which is needed to develop the full functionality of specification system for complex products in WP4. As such, it develops the required underlying database, as well as related data manipulation services for this system.

Establishing proper implementation in this deliverable is then imperative to the success of the other two deliverables of WP4. At a later stage, the specifications generated by stakeholders using the WP4’s developed system, constitute one important input for the VO creation process, as addressed in WPS.

This deliverable also provides a number of screenshots from the complex product specification system, while being used by Prolon partner in the project, for entering data related to the specification of an intelligent building, as the complex product.
1 INTRODUCTION

As the results of the GLONET’s requirement analysis stage indicated in [D1], the peculiarities and complexities embedded in our targeted complex products (e.g. solar power plant and intelligent buildings) make them both one of a kind in their design specification and massively customized. Furthermore, the Product Life Cycle (PLC) of these complex products requires attention, as addressed below. Their PLC of complex products run over several decades and can be divided into the following three main phases [D2], each with its own peculiar features:

(i) **design and engineering**,  
(ii) **construction and commissioning**, and  
(iii) **long term operation and maintenance**.

In this section we address these three phases primarily in relation to their specific needs to use the product specification tool. But before we provide this description, please note that the product specification tool is also required to be used before the PLC of the complex product starts. This is mainly due to the need for preparation of the bid for the targeted complex product, for instance in response to a call for tender. Please see Figure 1 below indicating the use of product specification tool during different phases of the complex products’ PLC, as well as during the pre-phase of bidding for the complex product.

Please also note that the letters “a” to “j” that appear on the arrows, as well as the numbers 1 to 10 that appear inside some squares in Figure 1, represent only the order of time where different activities related to product specification occurs, but with no time scale. Also note that the two red (almost horizontal) arrows on top of this figure represent this same PLC scenario, but without using any product specification tool to help with specification of complex products.
**Design and Engineering phase.**

Being the first phase in the PLC of the complex products, the design and engineering phase plays an important role in the success of the later PLC phases of the complex products. Activities during this phase are typically divided into the three steps of: Project Assessment, Project Design, and Project Implementation.

**Project Assessment step** includes the complete analysis of the site and the technical assessment of the entire project, at this step a *high-level specification of the complex product* is made in order to assess the feasibility of the project.

After the Project Assessment has been successfully preformed, during the **Project Design step** the early engineering, and selection of technology takes place. These include the following activities, which each need to be reflected in *some detailed specification of complex product*:

- (i) Pre-engineering (e.g. achieving the initial specifications of the complex-product),
- (ii) Evaluation/selection of the technology or equipment (e.g. evaluating for selection and/or extension of the existing devices and equipment suitable for the complex product), and
- (iii) Selection of sub-product specifications (e.g. adding sub-product specifications as components of the complex-product specification).

Finally after the Project design step has been successfully preformed, during the **Project Implementation step**, the *planned specification of the complex product is finalized* and the product specification is used for selecting the relevant organizations and for sub-product procurement.

**Construction and Commissioning phase.**

Typically during this phase, the product specification tool is not used for any design activity, rather it might be used only to access/retrieve some detailed design information.

**Operation and Maintenance phase.**

Although as discussed above, the product specification is fundamentally required during the entire design and engineering phase of the product life cycle, it is also needed to be used later during the long-term *operation and maintenance phase* of the product life cycle, but rather infrequently. The later occurs mostly due to the continuous need for evolution of the complex product, and/or to innovate and provide new products in response to the identified needs and/or the raised problems in the operation of complex products. An example case is when a device enhancement is required and/or the replacement of some existing components of the complex product become necessary. Another example case is when innovation or design of certain new component (e.g. equipment) is necessary for solving an emerged problem in the complex product, etc.

Further to the above, being complex, the specification of a complex product is typically not performed in one session. It is rather done iteratively in a number of sessions, and potentially involving a number of different stakeholders at each time, varying from equipment manufacturers or suppliers to the experts at the EPC (Engineering, Procurement, and Construction) company, who may collaborate and gradually and incrementally specify different components and their sub-components related to the complex product.

Additionally, based on our findings in this area, and our gathered expert opinions in the field, due to the industrial nature of our targeted complex products and being a relatively young industry, design and engineering phases of these complex products cannot be simply resulted through the mere searching and identification of their needed components among the existing products in the market.

In other words, although knowing about the existing product details as provided by the manufacturers or suppliers of related equipment and devices are the necessary starting point for the complex product designer, the mere existence of these product details are not sufficient to fully
specify the **Project Design** of the complex product. Rather, the nature of our targeted complex products, mandate a long and delicate process of designing and customizing, and therefore specifying details of the needed sub-products, including the needed specification of required equipment, devices, and their enhancing services, and further involving different stakeholders in this process.

As such, while complex products specification is mostly performed by experts from some consulting or EPC firms in interaction with the customer, there are several other stakeholders that can get involved in this process and directly provide the sub-product specifications. These include: product designers, developers, and providers of equipment and services related to the complex product.

Figure 2 shows a partial view of the involved stakeholders and the centric role that is typically played by the EPC in complex product specification area.

It is therefore necessary that the design and development of the Complex Product Specification system in GLONET supports all the above mentioned requirements, to be used during different PLC phases of the complex products by all the involved stakeholders. Additionally this system must support and integrate with other relevant software systems running in the environment, which constitute other forms of users for product specification system. To accomplish these aims, Product Specification System has focused on the development of three main sets of services/functionality, as also indicated in Figure 3 under the service-enhanced product support header. These include: the **Product Specification & Registration**, the **Service Specification & Registration**, and finally the **Product/Service Discovery & Recommendation**.

In this deliverable the first set of the above functionality, the **Product Specification & Registration** is addressed. We discuss the implemented prototype that provides a set of web-services developed within GLONET, supporting product specification & registration functionality to assist the user with modular and iterative specification a of the components of a complex product, in terms of its sub-products. The other two sets of functionalities are the subject of the next two deliverables.
Please note that the design of this prototype has been detailed out previously in [D3]. So, in this document we will only revisit this design when and if necessary to augment or enhance it.

Following sections of this deliverable provide more details on a number of important implementation aspects related to the product specification system. These aspects are structured as follows in the next sections:

- Section 2 – Realization of non-functional requirements for product specification system,
- Section 3 - Realization of functional requirements for product specification system,
- Section 4 - Implementation approach - General system description, and
- Section 5 - Examples of using product specification system,

Then concluded with the Section 6 on Concluding remarks, and Section 7 including references.
2 REALIZATION OF NON-FUNCTIONAL REQUIREMENTS FOR PRODUCT SPECIFICATION SYSTEM

At its base, system requirements address why a system is needed, what are the functions it must provide, how the system must be constructed and implemented, and what conditions must be satisfied by the system. The two main types of requirements to be considered are the non-functional and the functional requirements [1]. This section addresses the non-functional requirements for our product specification system. Non-functional requirements are those that are not directly related to the functions of the system, such as the security, scalability, availability, etc. In the follow up subsections, we will address these requirements and look into how the product specification system has realized the most important non-functional requirements, namely, the security, scalability and portability.

2.1 Security

Security plays a very important role in systems, for instance when dealing with profit-based industries. This is due to the fact that improper access to a system might bring loss and even bankruptcy to the organization using the system. Proper prevention of threats via the competitors, both from the outside world and from the inside of the system, is a must. The first step in this process is enforcement of proper authentication.

2.1.1 Authentication

Authentication means that the system must make sure that the user of the system is the one that he/she claims to be. There are three different main techniques used for authentication, including [2]: what you are, what you have, and what you know.

“what you are” makes sure that you are whom you claimed to be, by looking into your biometrics. “what you have” makes sure that you own the proper physical key or card that only you can have. And finally “what you know”, makes sure that you know what you should know, by asking you questions that only you would know how to answer. A very common form of “what you know” is your credentials (e.g. username and password).

Among the above, and when considering on one hand the usage/user of the product specification system and on the other hand the fact that this system comprises one sub-system within the complete GloNet system, we have chosen to apply a specific type of “what you know” technique, that is to use a token. In this approach instead of product specification sub-system asking every user to provide his/her credentials, this sub-system receives a token for each user, which is generated through the “Single Sign-On” mechanism implemented at the GloNet level. As such, this token is provided by the GloNet platform, which authenticates the user at the stage of its first sign-on into the system, prior to its use of any of the sub-systems.

The advantage of deciding to choose the single sign-on approach by the GloNet platform is that if the user is authenticated once by the GloNet platform, there is no need for the user to redo the authentication process for using any GloNet sub-system, including the product specification system. While single sign-on is an advanced convenient security provision functionality for users, clearly the threat of this approach has a domino effect. Meaning that in the rare case that the GloNet platform fails proper authentication of the users, the product specification system will also fail to do so.
2.1.2 Authorization

One of the most important requirements of a system that deals with multiple stakeholders (specifically within a VBE/VO environment) is its secure and proper information sharing. This is due to the fact that although these stakeholders cooperate to achieve a common specific goal, they can be potential competitors on other goals.

To preserve the users of our product specification system, against unwanted data access to their private information, we must prohibit any endangering access that might threaten the privacy of the involved users. To do so we have implemented three different data spaces for the users (levels of access) in this system, that limit who can access what. These spaces include the following:

- Private – only for personal use of the user
- Restricted – specified by the user to be shared within the VO related to one specific project
- Public – to be shared within the VBE

Figure 4 shows these spaces.

Please note that each product specification is owned by the person who defines that specification within the system. Therefore only the user that has defined the specification is allowed to move a specification from one data space to another, e.g. from private either to public or to restricted, in order to allow access, which is needed to either view it or use it for speciation of new products by others. In the upcoming sub-sections we will introduce the public, restricted, and private data spaces in more details, and later on, in section 5 we will provide examples of the functionality that supports the user with specifying and/or changing the type of data space (private, restricted, public) related to each specification that he/she provides, and therefore owns.
2.1.2.1 Private data space

The first step in the process of product specification is to specify sub-products of the complex product. To accomplish this task, the designers should have a private space to do their specifications before making them available to other stakeholders involved in specification process of the complex product. This space is called the private space and any specification done in this space is only accessible by its creator. It is important to note that any specification made by the user is by default put in this space and can afterwards be moved to the restricted or public spaces.

2.1.2.2 Public data space

After one has specified a product he/she might want to share the specification with the public. This is mostly exemplified by a sub-product provider who wishes to promote the use of an already built sub-product (e.g. equipment) by its organization, to be used for building other complex products. Although there are also other users nowadays who are involved and interested to participate in open access movements. To enable this possibility, the system provides a public space that the user can use to provide access to specifications, and all users can access such specifications.

2.1.2.3 Restricted data spaces

Within the process of specifying one complex product, multiple stakeholders might be together involved. In such a situation the user would be interested in sharing a certain specification, with a given set of other organizations (other involved stakeholders), which are typically part of a virtual organization (VO). To enable this feature, one user can indicate a separate restricted space for a product specification related to the VO, through which different users can share some previously private specifications within this space. Please note that every time a VO is created a restricted data space is created for it.

2.2 Scalability

Due to the nature of complex products, specification of such products might involve a large number of stakeholders. This means that during its PLC, a complex product might potentially be dealing with hundreds of users. To support such a user base and to enable possible expansions of both the user base and well as the product specifications, we have leveraged the possibilities supported by the cloud that allows allocation of more resources on demand, when and if needed. This leveraging has been done through applying three different techniques. First, different components of the system (i.e. the executable building blocks of mainly the Controller layer and the DAO implementation layer, as described in Section 4), e.g. web service controllers, web interface controllers, Hibernate DAO Implementations, GloNet DAO Implementations, etc., are decoupled from each other. This means that none of these components is dependent on how another component is implemented or executed, which in turn enables the execution of different components on different physical/virtual machines. Second, the implemented product specification system is layer-based (e.g. Application, Data, etc., as described in Section 4), while each layer is transparent for the other layers. Consequently, not only the components of the system are decoupled but also the different layers are decoupled and can run on different physical/virtual machines. And third, the implemented system can take advantage of load balancing mechanisms for data. This is supported due to the separation of the data access for objects, and how the data access is now implemented.
2.3 Portability

The product specification system has been developed as a web-based application using the Java programming language that enables the server side code to run independent of the platform. The server side program generates standard outputs (such as HTML 5 [3]) that could be consistently and easily rendered by different browsers.

The client side (Browser side) of the program is based on JavaScript and has been written using the jQuery framework, insuring that the JavaScript code is compatible with different browsers.

The combination of Java as the programming language of the server side and the compatibilities of the client side, such as compatible JavaScript and standard HTML 5, makes both the client (browser) and server side of the system highly portable.
3 REALIZATION OF FUNCTIONAL REQUIREMENTS FOR PRODUCT SPECIFICATION SYSTEM

The complex product specification system supports two main functionalities of product specification and product registration. This tool is designed and implemented to primarily assist the design & engineering phase of the complex product lifecycle. Nevertheless, it may also be needed from time to time to specify new sub-products during the operation and maintenance phase of the complex product life cycle.

The system facilitates specification of new customer-oriented products, and enables involvement of different stakeholders, e.g. equipment providers as well as the EPC designers. Within the context of the data level at the PLM framework [4] for complex products, three main product models need to be captured throughout its lifecycle. These include: the Geometric-oriented product models (e.g. CAD), the Structure-oriented product models (e.g. DMS – Digital Manufacturing System), and the Meta-data-oriented product models, as also illustrated in Figure 5. The product specification system primarily captures, handles, and manages the detailed meta-data about the complex product and all its sub-products, and assists users with their specification. Furthermore, the product specification system captures and stores links to a set of files that represent the other two product models, which are mainly produced in certain industry specific systems, e.g. CAD, CAM, and DMS software.

Figure 5 - Main product models for PLM

The aims of the functionality provided in the product specification system within the GloNet are three-fold. The first goal is to support gradual specification of the complex products. This is needed to reflect the reality of complex products that are neither defined in one session and nor by one stakeholder. Therefore, detailed specifications that capture and transform customer requirements for a complex product into discrete sub-product specifications, can be gradually defined by the involved multi-stakeholders, using the developed product specification system. The second goal is to
properly capture the classification of all relevant sub-products in a *granular and modular* manner in the complex product environment, e.g. distinguishing and capturing both the electrical and mechanical aspects of a sub-product. This will in turn support effective multi-perspective retrieval/discovery of information related to sub-products, as well as creating their concise presentation, needed for common understanding among different related stakeholders. The third goal is to capture all the details related to sub-products in a *reusable* form. As such, the existing specifications of already introduced sub-products can be either fully or partially (e.g. at the level of certain detailed feature-kind) be reused for the specification of other sub-products.

In the following subsections we address different steps involved in defining details of a product and/or sub-product in a modular manner, through the use of the product specification system.

### 3.1 Granular and customized specification of complex products

Having different levels of granularity and customizability for complex products is a must, due to their dynamic and complex nature. At the lowest level of granularity, the features of a specific sub-product can be defined. Every feature is an instance of a feature-kind. Through the granular definition of feature-kinds and instantiating features, the system enables the user to specify any sub-product from scratch and without being limited to for instance only defining sub-products as instances of a predefined type of product, with a pre-defined set of fields/attributes.

#### Feature-kind and features

A feature-kind (e.g. weight) is a characteristic of a sub-product (e.g. an equipment), which may be specified with multiple units (e.g. Kg, pounds).

One specific *value* for a feature-kind together with the *unit* for it, related to a sub-product, constitutes a *feature* tuple for that sub-product. The combination of all the features of a sub-product makes that sub-product unique.

For example a sub-product lamp may be defined with the following feature-kinds: input current, amount of energy it consumes, amount of light it produces, the color of its light beams, etc. But the definition of one physical lamp entity would become unique as we indicate the values for the above features, namely that it has an input voltage of 220 volts, consumes 100 watts of energy, and provides 800 lumens, of blue light. Therefore, the latter definition of lamp (through giving its features) fully specifies the real physical lamps, while the former definition of lamp (only specifying its feature-kinds) specifies the meta-data for the type lamp.
3.2 Capturing product specification perspectives using classification

When specifying a sub-product related to a complex product, due to the multi-disciplinary nature of the complex-products, it is important to enable the user with providing different perspectives of the sub-products, based on their related features. This can be made possible by the product specification system by means of classes. Other than that, classes can also guide the user to provide proper feature information related to the sub-products. For example one can define feature-kinds that are obligatory for a given class, so that later on if a user identifies a product as belonging to certain class of products, the user is warned to also provide the features for its obligatory feature-kinds.

### Classes

As addressed before, the feature-kinds defined for a class of sub-products, together represent a specific classification for that sub-product in the system. We introduce a **Class** as the basic classification means defined to model generic categorization of all sub-products, where in turn each class of sub-products has a specific set of feature-kinds associated with it. Classes can also be defined on top of other classes. Therefore, complying with the generalization / specialization (sub-class / super-class) abstractions.

Furthermore, we can enforce a set of obligatory feature-kinds on sub-products. For example, definition of a class “Electrical” (a class representing electrical sub-products) can enforce the definition of an obligatory feature-kind, such as “voltage”, to this sub-product specification. In other words, any sub-product class that has “Electrical” as super-class, must have the “voltage” feature-kind and therefore any product specification with that class is obliged to have a feature (e.g. feature value) for its voltage.

The figure below shows four different example classes that are defined for sub-product classes in PV power plants, and each represents a set of features-kinds that are specified for those classes, e.g. sensor class has a feature-kind accuracy.

![Diagram of classes](image)

3.3 Sub-product re-specification

When dealing with complex products the user may wish to slightly re-specify a sub-product, or customize a new sub-product specification based on an existing sub-product specification. Here the needed functionality from the product specification system is provided by this system by means of duplication. As such, the duplication functionality supports the above requirements.
3.4 Complex product/sub-product Launch Request

After the designer has specified a product, its specification should be sent to the VO planner in order to configure and establish the needed VO for its realization. This step is represented in Figure 1, in relation to different stages of the product life cycle. For instance, it is indicated through the arrow labelled as “e”, taking the specification results from box 4 to the box 5 where the product/service VOs are planned and established. It is similarly shown with “b” from box 2.1 to 2.2, as well as with “i” from box 8 to 9. Also Figure 4 illustrates this functionality as the request for sub-product launch. This request can be issued by a designer or advisor, and must result the packaging of the sub-product specification, and sending it through the cloud to the system that supports the product VO formation.

The product specification system needs to support this functionality to enable the user with requesting the initialization/launching of the process that can realize the targeted specification. In other words, this request shall trigger the process of planning a goal-oriented VO, as addressed by the consortium formation and operation support in GloNet. To exemplify this process, Figure 6 shows the business process of how the packaged specification generated by the product specification system is passed over from the Broker to the VO planner, to be processed toward the formation of its needed VO.

![Figure 6 - Launching a characterized new product - Planning goal-oriented VO for its realization](image)
4 IMPLEMENTATION APPROACH – GENERAL SYSTEM DESCRIPTION


More details about the implementation architecture of this system and its main components are presented in figure 7, and described in the following sub-sections.

As the general framework for its implementation, the product specification system follows the MVC (Model–View–Controller) software design pattern [8]. MVC enables modular development of software systems, which can benefit the implementation of our product specification system, as addressed later in this section. This model is divided into five layers of:

- Entities,
- DAOs (Data Access Object),
- DAOs’ implementation,
- Controllers, and
- Views,

as also depicted in Figure 7. In the remaining of this section, in a bottom-up manner we will describe the details of each of these Layers.
4.1 The Entities Layer

This layer consists of the main entities introduced in the system. The implementation of this layer is done using the POJO (Plain Old Java Object) concept [9], and as such the implementation in this layer is totally technology independent, and is only based on the plain Java. Nevertheless, while the implementation is POJO-based, the code in annotated with Hibernate [6] annotations, which enables Hibernate [6] to understand the relations between classes and tables in the database.

Figure 8 shows all the defined entities in product specification system, and their inter-relations’ cardinality.

![Class diagram and cardinality of inter-relationships among the entity sets](image)

Furthermore, Figure 9 shows the detailed UML class diagram for the product specification sub-system.
Figure 9 - Class diagram and cardinality of inter-relationships among the entity sets

Here we provide a short description of every entity defined in the UML diagram. The Product entity is represents all products/sub-products, which will be defined in the system. The Feature-kind entity represents all different kinds of features that can be defined. The Feature entity, characterizes different aspects of the Products, 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 launching of a new element in the system, including the launching/establishment of for instance a new project (e.g. Building-Amsterdam), or a new compound product (e.g. a control-box for controlling the sensors in the building-Amsterdam). Please note that the Request entities generate one of the outputs of the Product Specification System, which starts a business service for the VO creation functionality of GloNet. Definition of this entity and its corresponding business service, which will create its corresponding VO, are the subject of the next task on “dynamically customizable services enhancing products”, and will therefore be addressed in details later in next deliverables. The Organization entity briefly characterizes the organization, which issues a request for launch. The Class entity is the generalization of the Product, Feature-kind, and Request entities.
4.2 The DAOs (Data Access Object) Layer

This 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. DAO layer enforces that higher layer implementations in the product specification system access all these entities only through the provided minimum interfaces.

This layer also in turn enables the possibility to build and access entities, independent from both the used technologies and the data sources. In this direction, in the layer above the DAOs, two different sets of implementations are developed in product specification system as described next.

4.3 The DAO Implementation Layer

The DAO implementation layer implements the DAOs addressed in the previous layer, thus providing access to each of the defined entities at the entity layer. This implementation is done partially as web-based services developed on top of the GloNet platform. These web-services enable access to entities that are provided through the GloNet platform database (GLONET - Data Model Layer), e.g. access to the organization entity, as well as the name and other information about the user who is logged in, etc. The other part of the DAO implementation provides Hibernate-based access to the MySql database through the cloud, e.g. to access the product and feature specifications, etc.

4.4 The Controllers Layer

The interactions between the users (human or software) and the system are handled through two separate sets of controller components, which are implemented within the controller layer. Therefore, users interactions with the product specification system are divided into: (i) the web service interactions (i.e. by software) and (ii) the web interface interactions (i.e. by human).

The web service interactions supporting the software systems, may execute a web service provided by the system, for registering a new product specification. As such, the web service controllers component handles the request & respond actions, which will be preformed among the calling software system and the specific web service defined for the registration purpose in the product specification system. Furthermore, human users in the environment can also directly use through the web interface, certain functionality provided by the products specification system, e.g. viewing the list of product specifications or feature kinds. As such, the web interface controllers handle the actions requested by users and provide the proper response to the human users’ requests.

4.5 The Views Layer

This layer provides a set of different interfaces on the exchanged data/information between the product specification system and its users (human or software system). The view layer decouples the request & response for data from the actual data itself, which is sent and received by the system. Three formats are currently considered and implemented for the transferred data in product specification system, including: HTML, JSON and XML, where HTML is implemented for the human interactions, while XML and JSON are used for interaction with software systems. It is important to note that some of these data views might produce more than one output file/type, for example the HTML view also produces the javascript and the css to further enable the HTML document.
For the user interface the product specification system uses the AJAX technology and the JQuery framework to enable producing a smooth and highly interactive experience for the user. Ajax enables the user interface (providing the HTML view), to directly call the services provided by the product specification system, and helps with reducing unnecessary interactions with the user.
5 EXAMPLES OF USING PRODUCT SPECIFICATION SYSTEM

This section aims at providing some examples and snapshots from the running product specification system, while illustrating the interfaces developed for current use of this system. Please note that at this stage, the developed system and its interfaces provide only the base, which in the coming time will be enhanced with the results of other planned tasks in GloNet. The current development has addressed specifically the functional/non-functional requirements addressed in Sections 2 and 3 of this document. As such an emphasis has been also on developing needed data manipulation operations, e.g. Add, View, Duplicate, and Alter, on sub-products, classes, feature-kinds, and units.

Below, we first provide an introduction to the general view (header and menus) of the product specification system, as presented in Figure 10.

- On the top right, after Hello, it indicates the name of the user (who is logged in), followed by the name of the project (optional) on which the user is working. Provision of this optional project name assists the user to better organize his/her folders of design specifications, and provides the possibility of switching between different projects. The project name is then followed by the name of a virtual organization (VO) to which this work belongs, providing the

Figure 10 - Product Specification System
possibility to also switch among the VOs. Finally a link for log out from the system is provided.

- On the top left, two main functionalities are represented, each opening a set of menu items, that give access to different functionalities related to either specification of products or requesting a product launch, which are described further below and in the following sub-sections.

As mentioned earlier, although the main usage of the product specification system is to specify products (as addressed in section 3.1 to 3.3), it is also used to request the initialization/launching of the process that can realize the targeted specification (as addressed in section 3.4). The request for launch triggers the process of planning a goal-oriented VO, as being addressed by the consortium formation and operation support in the GloNet system. Therefore, at the top level when using the product specification system, there are two main menus provided by the system, namely: Product Specifications and Requests.

Under the Product Specifications menu, there are two sub-menus. Namely, the user either can Specify Product or view the Existing Specifications, where the latter is exemplified in figure 11, and the former is described in more details below in this section.

![Figure 11- Existing specifications (related to project Amsterdam Building)](image-url)
Existing product specifications can be presented to users either from the viewpoint of a specific user’s project, or from the viewpoint of a specific VO. In either case, the user will specify the name of either the project or the VO respectively, for this purpose. Please note that the semantics of these viewpoints are very different, and while the first one indicates one optionally defined folder of the user (to serve as a working area for him/her), the other one is in fact dedicated to all specifications belonging to one VO.

As illustrated in Figure 11, the list of specified products is provided with their: names (that may include in front of it a “(-)” for private, and “(#)” for restricted data, or nothing for public, as explained earlier in section 2), classes, and a set of icons representing different actions that can be performed on them, which are described later.

Under the Requests menu, the user can add a new launch request for one of the already specified products, as shown in figure 12. This option is also further explained in a sub-section with more details.

![Figure 12- New Launch Request](image)

It is important to mention here that the current version of the product specification system is under the trial stage by Prolon partner of the GloNet consortium. Therefore, all the screenshots provided in this section present different cases of the usage of this system for specification of different sub-products related to a complex product within the intelligent building domain.

The remaining of this section describes different features supported by the system, and illustrates them with some screenshots. Furthermore, we have provided some general information for the
users of the product specification system in ANNEX A, that can be used as an introductory guideline for specification of products/sub-products in this system.

5.1 Adding a new product specification

Figure 13 shows the snapshot of the interface window for product specification. This window is used for specifying a new product, e.g. a sub-product of a complex product, as well as to iteratively specifying further sub-products for it. As such, this interface can be used to specify/register/add a new (sub-)product at any level of abstraction/granularity, including the complex product itself.

To simplify the labelling of presentations in this section, this interface window is called “New Product” window. In New Product window, the user can add a product specification by first providing a unique name for the product. The user can then optionally define one or more classes for the product, as well as one or more sub-products and/or features relevant to the new product, and depending on its classification, for which their detailed definitions have been provided earlier.

![New Product specification window](image)

Figure 13 - New Product specification window

It is important to note that if the user, who is specifying a sub-product for an intelligent building, defines a class for it, e.g. the conference room type or the office room type, and then through the system all the features defined for that specific class will automatically pop-up in this window. In other words, the user is assisted with receiving the names of the features, which he/she should fill and specify for that sub-product. Furthermore, while specifying 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 13 since the user has added the class “Building automation sensor” in relation to the “LRM8114/00” product, which he/she is specifying, a set of feature-kinds for this sub-product have automatically popped up at the bottom of the page, to be further specified. These include a set of feature-kinds, namely: “Supply input”, “Supply power”, and “maximum mounting height”, that show up on the screen. Please also note here that providing input (i.e. features) for the Supply input and Supply power feature-kinds are obligatory, since the remove button option (red circle) is not indicated on their right end. Therefore the user is obliged to define the value and unit for these two features. At the same time, the third feature appearing there has the remove button in front of it, and therefore providing input for it is not obligatory.

There are several other aspects involved in specifying a product that can be defined through this interface. For instance, if a product is composed of other sub-products, they can be all indicated in its specification. Consider the example shown in the Figure 14. Here, the user has indicated all the needed sub-products for the product Auditorium, and for each sub-product, it is specified the needed quantity of that sub-product. As an example, note that the composed product “Auditorium” consists of 4 “PSW5036”s, so the user indicates “PSW5036” with the quantity 4 as one of its sub products.

Finally, if the user wishes to add any other new features to a product specification, the New Feature item within the Features section of the interface can be used for this purpose, and there the user
should first indicate the feature-kind to which the feature corresponds. In fact, within the presented example in Figure 13, the user has entered “maximum mounting height” 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, as represented in Figure 13. 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 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 product specification process, the specification can be saved or discarded, through the two buttons appearing at the end bottom of this window.

5.2 Adding a new Launch Request

When a designer completes the process of specifying a product and/or sub-product (at any level of granularity), typically he/she wishes to initialize the process of realizing that product. For this purpose the product specification system supports the functionality to Launch Request (also referred to in this document in short as request) for starting this product realization process. As such, when and if a designer wishes that a certain product specification should be realized, he/she can announce this fact through building a request. For instance the designer of a sub-product (Pyranometer – see Figure 15), after specifying all its details, can build/make a launch request for this sub-product, while providing its sub-product specification. In fact designers of innovative sub-products are interested and curious to check if their specifications can in fact be realized, meaning that they wish to know if their specifications are buildable and can be developed, and by which potential consortium of companies it can be done.

In order to build a launch request, the user can use the interface developed by the product specification system, as indicated in Figure 16, and labelled as the New Launch Request window. Please note that before a new Launch Request can be built for a product specification, the product must be first properly specified, as addressed in Section 5.1. Then through the interface presented in the New Launch Request window, the user will identify one or more specified sub-products (i.e. their specifications), to be included in the package for this request of launch.

Besides the identification of sub-products, when defining and packaging a launch request it must be clear by which member of the VO this design specification has been made as well as for which project and customer the design has been made.

In Figure 16, the sub-product called Auditorium is requested to be launched. The default value of the “by” field of the request would be indicated based on the user-name who is logged into the system (e.g. Prolon). The default value of the “for” field of the request, would be indicated based on the project-name and VO-name specified on top of the screen. Figure 16 indicates a request specified by Prolon Control Systems for the Amsterdam Building project where the customer is UvA.

In case, no project name is specified on top of the screen, it in turn indicates that the specification is done by a manufacturer or provider of a sub-product, and is aimed for public. In such cases the keyword “public” appears as default in the “for” field.
It is in principle not obligatory to provide any other piece of information for a launch request, however if the user wishes, he/she can include more information in the package of this request. For instance, he/she can add the class “Design” to this request, to indicate that this is a specification for a controller product to be designed within a VO. If so, the selected class(es) may in turn oblige the user to also specify certain other features for the request. In the example of Figure 16, identifying the class “Design” obliges the user to specify value and unit for the feature “Date” for this launch, here indicating the design-deadline for example.

Please also note that in order to assist the users of the product specification system, for some rather generic features such as the date, an assisting function is developed. For instance, for date a pop-up window will appear to assist users with selecting a date, instead of having to enter the date with certain format.

Furthermore if desired, the user can also add other optional features to this request, through the use of New Feature. All specified features, including the specification of the customer (on behalf of whom the request for launch is triggered by the user), in this example being UvA, become a part of the package for the launch request, and will be dealt with at the side of the system receiving the launch request, which is the consortium formation and operation sub-system of the GloNet system.
At the last stage of the launch request process, the request definition can be saved or discarded through the buttons at the bottom of the window.

5.3 Adding a new feature-kind

Several entities defined in the product specification system, e.g. product specification and Launch Request are characterized through a set of features. Any identified feature in this system requires that its feature-kind is defined a 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 system is supported through a pop-up window, as indicated in the “Create New Feature-Kind” window in Figure 17.

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 add a number of sub-feature-kinds to it, when a feature-kind is composed. Another option is to indicate one or more possible units for the domain values specified for the feature-kind.

For the purpose of defining a new feature-kind, a popup window will open, as seen in Figure 17, which can be triggered when defining the new product specification or the launch request.
At the last stage of the feature-kind specification process, the feature-kind can be 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.

5.4 Adding a new unit

Users can introduce a new unit (e.g. Volt) for the features instantiating a feature-kind, through the “Create New Feature-Kind” interface, as indicated in Figure 17. 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 (See Figure 17). 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 saved or discarded.

5.5 Adding new classes

As discussed before, each product specification or Launch Request could be associated to one or more classes. As such, classes define the meta-data for products and Launch Requests, 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” window in Figure 18. To add a new class, the user must provide a unique name for the 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.
5.6 Viewing / managing existing product specifications

Once products are specified, they can be viewed by selecting the “Existing Specifications” item under the product specification menu. As such, depending on the selected VO-Name or project-name (as indicated in the upper right corner of the screen), their associated existing specification window will appear, showing the list of all relevant existing product specifications (sorted by their names), which the user is authorised to view. In other words, the specifications that are included in this window are all those related to either the specified VO or the specified project.

Please also note the following three cases:

- If neither a specific project nor a specific VO is identified by the user, then all public product specifications in the system, plus all the private specifications of that user, will be shown in the window.

- If no project is specified by the user, but a VO is specified, for which the user is authorized, then the restricted specifications for that VO, plus the public products related to that VO will be shown.

- If the user specifies no VO, but a project is selected, then all the private, restricted, and public product specifications that the user has associated to this project folder will be shown.

In the example of Figure 19, the Prolon user has selected/indicated the VO name “Electrical Design”. Consequently in this example, all restricted product specifications that belong to this VO, as well as the those public sub-products used for this VO are shown. Please note while Prolon might own some of these restricted products, some others might be owned by other users.
Other than viewing the product specifications, authorized users can also manage these specifications by performing the following set of actions:

- **Duplication** (Shown with the icon ![icon](image)) action, which takes the user directly to a pre-filled “New Product” window. This simplifies the task of users, since in that window the specification information about the selected product is duplicated, which can then be edited by the user, thus defining a new similar product specification.

- **View** (Shown with the icon ![icon](image)) action, which takes the user to the view window of the product specification.

- **Delete** (Shown with the icon ![icon](image)) action, which allows hiding certain specification from the users screen, for instance the user finds a sub-product useless for him/her use and so the user deletes this specification from his/her view.

- **Share** (Shown with the icon ![icon](image)) action, which provides the user with the option to change the access rights/sharing status of a certain product specification that he/she owns. The share options are available through existing products window, when the user clicks on its icon. Please note that when defining a new product specification, the access right to that specification is made private by default, that is if the user has not indicated a VO on the top right corner of the screen, otherwise the specification will become restricted to that VO by default. At any point in time, the owner of the product 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 either restricted within a VO, or public. But product specifications that are for instance restricted to a VO or available to public, their access cannot be reduced to private. In other words once the owner of a product specification grants certain access rights to others (to view the specification) he/she cannot withdraw that right later.

To summarize, the following changes can be made to the access rights of a product specification: i) from private to restricted to a VO, ii) from restricted to public, and iii) from restricted (to one or a set of VOs) to a larger set of VOs, which also includes the original VOs.
• **Assign to Project** (Shown with the icon 📥) action, which provides the possibility to assign an already defined specification to which user has access, and might be already indicated as private, public, or restricted, to an existing project folder of the user. This is mainly to assist the user with organizing his/her product specification folders.

This means that by default when specifying a product, if the user has not indicated a project on the top right corner of the screen, the specification does not get assigned to any specific project. However, when a project is indicated on top right corner while specifying a product, then that product will be allocated to that project folder. Nevertheless, through this action which is provided in the Existing Specification window, products may be assigned and reassigned to different project folders.
6 CONCLUDING REMARKS

This report describes the prototype developed for the product specification system. It takes as a main input the design specification and introduced mechanism for the specification of customized complex products, as addressed in deliverable D4.1. Also, this prototype implementation uses a number of base services provided by the 2nd release of the GloNet platform.

It is expected that with the forthcoming releases of the GloNet platform, an enhancement of the developed functionalities can take place, e.g. in relation to user authentication. These enhancements as well as the implementation of other advanced functionalities planned in T4.3 and T4.4, will then be reported in deliverables D4.3 and D4.4.

The advantage of the currently developed prototype reported in D4.2 is that it is both generic and reusable. It therefore serves as the base for:

- other developments in WP4, now being designed in task T4.3, addressing the customized service-enhanced product specification.
- other developments in WP5, namely for providing and packaging the needed input for the functionality of consortium formation and operation support.
7 REFERENCES


Referenced GloNet deliverables:

- [D1] GloNet D1.1 deliverable – Detailed Requirements for GloNet use case and Domain Glossary
- [D2] GloNet D2.1 deliverable – Required Information/Knowledge Provision Services Specification
- [D3] GloNet D4.1 deliverable – Design report on approach & mechanism for effective customized complex product specification
- [D5] GloNet D2.4 deliverable - Mechanisms for defining composed services to support collaboration
ANNEX A – INTRODUCTION TO USING PRODUCT SPECIFICATION SYSTEM

The first window you arrive at when browsing to the URL is the “New product” window (Please note that at this stage you are already signed in through the single sign-on feature of the GloNet platform).

As a rule of thumb, please remember that the best approach for complex product specification is to do so in a bottom-up manner, to the extent possible. That said; do not worry if you are not always able to do so. In other words, you can always add sub-product specifications at any level of details. Furthermore with the use of duplicate and alter, you can easily copy existing specifications, and glue them to each other, and/or enhance earlier design specifications.

Use the above window to specify/define a new product and its sub-products iteratively. Our suggestion is that you start by specifying the smallest sub-products (for instance the smallest piece of equipment) first, and then specify the more compound products, and the higher levels.

When adding the product specifications, it is optional to further add for them: classes, sub products, and even features. But, if you wish to do so, and if possible please first add the required features for the sub-product, and then define classes related to the product, including those features.
Nevertheless, whenever relevant, features-kinds that are defined previously in the system will show up as suggestions (similar to auto-complete), to assist you while trying to type the name of the feature.

Please note that you can define new feature-kinds from the window by clicking the green plus sign in front of the new feature input.

While defining a new feature kind you can provide for it: a name, its type, and the possible units. Please note that a few default types are already defined in the system, e.g. color and date, but this set is continuously enlarged by users who define more types into the system. Furthermore, during the development of D4.4, we would further expand the default types, to better support the applications of intelligent buildings and PV power plants. Please note that if a unit has been defined previously, it would be suggested to you when defining the feature-kinds, otherwise you can add new units by clicking the green plus sign in front of it.

It is also possible in the system to define sub-features kinds, but our suggestion is that you keep the features as simple as possible, and avoid having a hierarchy of features at this point in time.

After defining the sub-products, you can then define more compound products on top of them that contain those sub-products.

Also note that the sub-product names will be suggested to you, while you try to type their names in the field.

Please note that after saving a product specification, it becomes permanent and you will not be able to edit or remove it. But you can always duplicate a product specification by going into the product
specification menu and choosing the existing specifications option. There, you can choose to duplicate a product, by clicking on the “duplicate” icon in the row presenting the product, which takes you directly to the new product window, to start with the currently existing specification, and allowing you to edit and extend that definition. The existing specification window, related to one project folder is also shown in the figure below as an example.
CONSORTIUM

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