 THE RESEARCH LEADING TO THESE RESULTS HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S SEVENTH FRAMEWORK PROGRAM (FP7/2007-2013) FOR CRYSTAL – CRITICAL SYSTEM ENGINEERING ACCELERATION JOINT UNDERTAKING UNDER GRANT AGREEMENT N° 332830 AND FROM SPECIFIC NATIONAL PROGRAMS AND / OR FUNDING AUTHORITIES.

CRitical S**YST**em Engineering **A**cce**L**eration

Meta-Model specification V1
D602.011

Project	CRYSTAL
Grant Agreement No.	ARTEMIS-2012-1-332830
Deliverable Title	Meta Model Specification V1
Deliverable No.	D602.011
Dissemination Level	PU
Nature	R
Document Version	V1.00
Date	2015-11-04
Contact	Guilherme Baumgarten
Organization	OFFIS
Phone	+49 441 9722-176
E-Mail	guilherme.baumgarten@offis.de

AUTHORS TABLE

Name	Company	E-Mail
Anna Todino	ALA	anna.todino@alenia.it
Guilherme Baumgarten	OFFIS	guilherme.baumgarten@offis.de

CHANGE HISTORY

Version	Date	Reason for Change	Pages Affected
V0.1	2013-08-29	Initial version	All
V0.1.1	2013-10-16	Addition of initial CRYSTAL meta-model concepts	“Meta-model for Platform Builder” chapter
V0.1.2	2013-10-25	Addition of ALA contributions	All
V0.1.3	2013-10-30	Add of Process activity diagram	“Meta-model for Platform Builder” chapter
V0.1.4	2013-11-06	Update of the sections and activity diagram	All
V0.2.0	2013-12-05	Restructuring of deliverable sections, complementation of content, explanation of terms and entities (glossary)	All
V0.3.0	2013-12-13	Integration of feedback provided during internal review phase	All
V1.0.0	2014-01-29	Integration of feedback provided during external review phase	All

CONTENT

List of Figures

List of Tables

Acronyms

The first column acronyms stand for the text in the second column:

CMM	Common Meta-Model
IT	Information Technology
MM	Meta-Model
RTP	Reference Technology Platform
SEE	System Engineering Environment

1 Deliverable Scope

Industrial experts from different domains agree on the necessity to enhance and simplify the specification and deployment of product design and development environments. CRYSTAL aims to define the so called System Engineering Environment (SEE) framework as an environment where a tool-chain is instantiated, taking into consideration the IT infrastructure and project data, and set up in order to perform a desired process needed in a specific project scope. From CRYSTAL project point of view, the SEE is an embedded system product development environment and the Platform Builder is a solution for realizing the instantiation of the SEE based on configuration information of a specific development process. The main objective is then to define innovative solutions and methods in order to improve the instantiation of this System Engineering Environment.
The present document defines the CRYSTAL Platform Builder Meta-Model concepts for a correct System Engineering Environment configuration and instantiation. Thus, the CRYSTAL Platform Builder Meta-Model must be able to detail the development process to be instantiated in a way rich enough as required by companies, including used tools, their interactions, needed project data and required IT infrastructure. Thereby, CRYSTAL Platform Builder Meta-Model addresses all the concepts that are necessary for covering the aspects exposed in the last sentence as well as concepts concerning tool description, providing thus a standard for tool providers. Figure 1.1 depicts the main information types that are covered by the CRYSTAL Platform Builder Meta-Model.

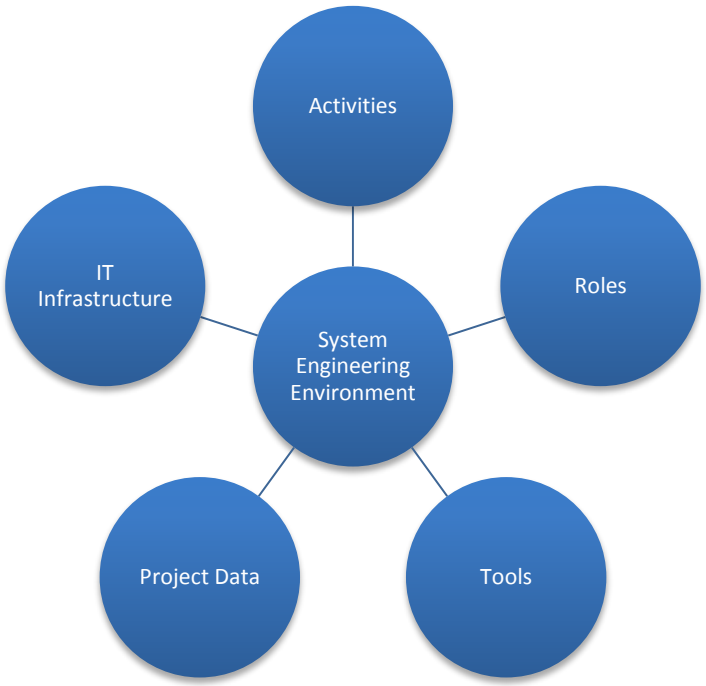

Figure 1.1: Identified items within the Platform Builder workflow

1.1 Deliverable Role

The objective of this document is to describe a meta-model which denotes relevant concepts for the SEE configuration and instantiation. The representation of these concepts in the meta-model level is necessary in order to have a common understanding of the System Engineering Environment from both industrial and tool vendor partners' point of view, facilitating the SEE instantiation process. This document contains data requirements for SEE configuration and Platform Builder tasks and it represents the main input for the Platform Builder specification and prototyping tasks.

1.2 Relationship to other CRYSTAL Documents

This document can be seen as an input in terms of requirements that are relevant for SEE configuration and instantiation as well as it is an input for the Platform Builder specification document (D602.021) and for the prototyping task (D602.031).

1.3 Structure of this document

In order to make clear the way this document should be studied, a short explanation of its structure is essential. It is composed by three main sections:

- *Platform Builder Overview* introduces to the reader an overview of the Platform Builder in order to make easier the understanding of the next sections and of the decisions that were made during their development;
- *CRYSTAL Platform Builder Meta-Model requirements* exposes the meta-model requirements, as inputs coming from the different partners, which were the basis for the creation of the meta-model;
- *CRYSTAL Platform Builder Meta-Model* presents the meta-model itself, explaining exhaustively its elements and their relationships, which aggregates the idea of a meta-model for the whole identified Platform Builder workflow, including tool specification and validation of the configuration.

2 Platform Builder Overview

The so called Platform Builder is a solution for improving the configuration and instantiation capabilities of the System Engineering Environment (SEE). Based on a Business Process description (see Glossary), the Platform Builder aims to set up a SEE for a specific domain development project. The required SEE configuration has to be validated against the available tools and the IT infrastructure. Figure 2.1 shows in a simplified way the idea of the Platform Builder: what is the needed input in order to have the desired output.

Figure 2.1: Platform Builder function

In the scope of a business process instantiated for a project/product giving a validated SEE configuration as result, the Platform Builder has to accomplish three main activities: tailor the process for a specific project and configure and validate the SEE. Figure 2.2 shows the sequence in which these activities are executed:

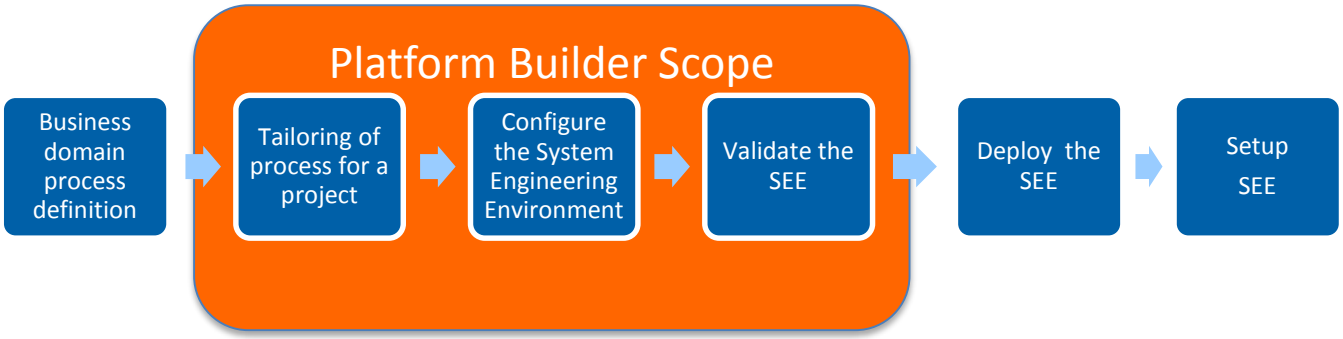

Figure 2.2: Platform Builder workflow

As it can be seen in Figure 2.2, the scope of the Platform Builder is limited to define, starting with a business process, which are the required elements to establish a properly and achievable SEE.

2.1 Platform Builder Baseline Process

The preconceived instantiation workflow for the Platform Builder process is the first and main input for the creation of the CRYSTAL Platform Builder Meta-Model. The process of configuring and instantiating a System Engineering Environment (SEE) in the scope of the CRYSTAL Platform Builder can be seen in Figure 2.3:

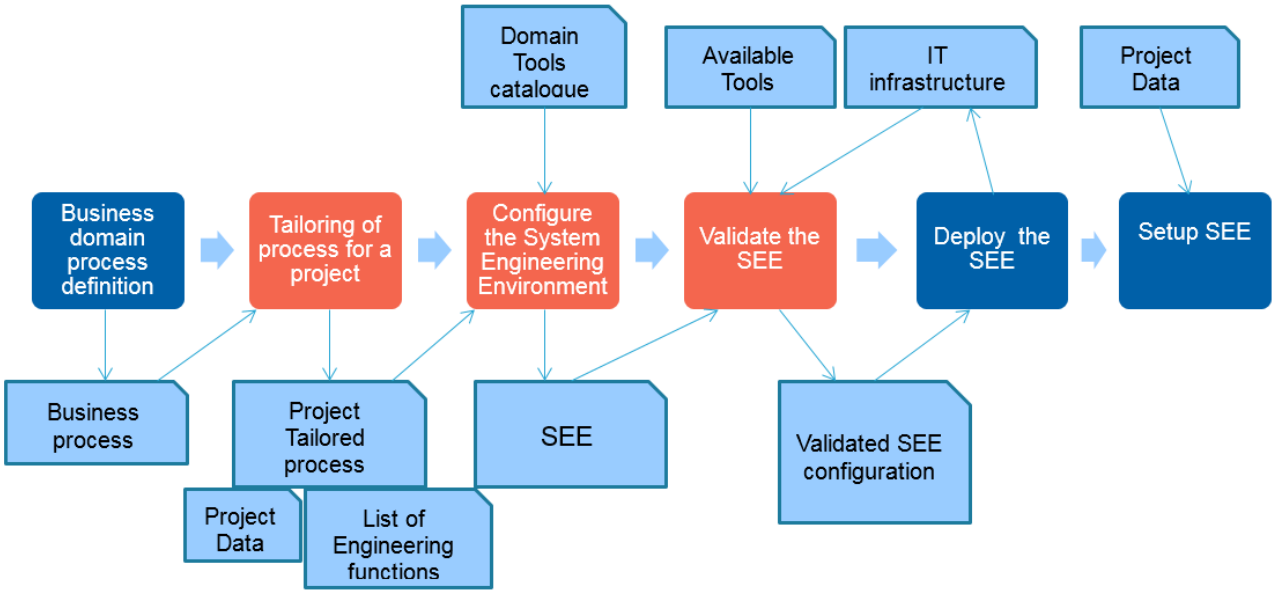

Figure 2.3: System Engineering Environment configuration and instantiation workflow[1]

From Figure 2.3 (for definitions, see Glossary), it is possible to notice that the basic idea is to execute a set of different activities one by one, having as input a business process and providing then as output a validated SEE. The activities that are explored in the scope of the Platform Builder, as exposed before, are to tailor the process and to configure and validate the SEE. The involved activities can be better depicted as follows:

1. Tailor the process for a project: this step consists in identifying and defining the process activities and their sequence, the artefacts (inputs, outputs, etc.) used in each activity, the process engineering functions to be used and the roles (define who uses the artefacts and his user rights access).
2. Configure the SEE: here, different process activities and their engineering functions (process engineering functions) are mapped to engineering tool functions (provided in the domain tools catalogue[2]), a model to represent work products in a repository is defined and IT infrastructure constraints (as required by the tailored process) and the needed IOS properties are identified.
3. Validate the SEE: this last stage is responsible for mapping the process engineering functions onto available tool engineering functions (checking the needed engineering tool functions against the available engineering tool functions), comparing the required IT infrastructure properties with the company current IT infrastructure and verifying if the required IOS properties are available.

Consequently the CRYSTAL Platform Builder Meta-Model has to cover all the required aspects in order to successfully perform sub-activities within each one of these main activities.

[1] The light blue boxes in Figure 2.3 represent artefacts which are accessed or created by the different activities and therefore are defined in the chapter *Glossary*.

[2] Engineering Tool Functions (provided by different domain tools) might not be available in the company. This configuration phase considers only what is necessary to perform the activities but does not consider if the engineering tool functions are available. This evaluation is in charge of the validation phase.

3 CRYSTAL Platform Builder Meta-Model requirements

This section presents the identified meta-model requirements, which were provided by the different domain and expertise partners. These requirements are relevant to the Platform Builder needs in terms of exchanging/managing/generating data.

3.1 Identified process elements

Taking Figure 2.3 as reference, a necessary step to the definition of the requirements that fulfil the Platform Builder Meta Model is to make explicit the breakdown elements and their relationships that compose the Platform Builder process. Table 3.1 details the different elements that are either used as input, output or are generated in this process, taking into consideration that in some cases elements comprise other elements. The definition of these relationships is important for the establishment phase of the meta-model.

Element	Generated in	Used as input in	Description	Data Format
Business process	Business domain process definition	Tailoring of process for a project	The business process provides generic information of a specific business domain process.	Business process descriptor
Project tailored process	Tailoring of process for a project	Configure the System Engineering Environment	The project tailored process details the process activities and their sequence, the artefacts being used, the participating roles and used engineering functions for a specific project. It comprises: • Project Data • List of Engineering Tools	Tailored process descriptor
Project Data	Tailoring of process for a project	Configure the System Engineering Environment Setup the SEE	Project Data describes the artefacts and roles used in the tailored process. It is part of the Project tailored process.	Project data descriptor
Engineering Functions (List)	Tailoring of process for a project	Configure the System Engineering Environment	This is the list of the engineering functions being used in the project. It is part of the Project tailored process.	A list of engineering function descriptors
Domain Tools Catalogue	See previous row	Configure the System Engineering Environment	It contains information about the existing tools in the RTP. For each tool must define the provided tool functions, the artefacts being used - in a per function basis - and IOS support. It comprises tools descriptors.	List of Tool descriptors

Element	Generated in	Used as input in	Description	Data Format
Configured SEE	Configure the System Engineering Environment	Validate the SEE	SEE maps activities to tool functions, defines a model to represent work products in a repository, identifies IT infrastructure constraints and the needed IOS properties. It comprises: • Tool Chain • IT Project Infrastructure • IOS properties	SEE descriptor
Tool Chain	Configure the System Engineering Environment	Validate the SEE	Details the tools being used and their relationships, mapping outputs of tool functions to inputs of other tool functions. Establishes the IOS needs. It is part of the SEE. It comprises: • Project Tools • IOS requirements	Tool Chain Descriptor
Project Tools	Configure the System Engineering Environment	Validate the SEE	Details the needed tools for a project. It is part of the Tool Chain. It comprises a list of tools.	List of tool descriptors
IOS requirements	Configure the System Engineering Environment	Validate the SEE	Details the IOS needs of the tool chain. It is part of the Tool Chain.	IOS descriptor
Available Tools	See previous row. Also, it is updated during the (re)evaluation of the tool catalogue.	Validate the SEE	Information about the tools available in the company. For each tool it is required to define the provided tool functions, the artefacts being used - in a per function basis - and IOS support. It comprises a list of tools.	List of tool descriptors
IT Project Infrastructure	Configure the System Engineering Environment	Validate the SEE	This data provides information about IT infrastructure constraints for the project: needed repositories, security and safety constraints.	IT descriptor (IT Infrastructure Descriptor)
IT Company Infrastructure	See previous row. Also, it is updated during the (re)evaluation of the IT Infrastructure	Validate the SEE	This data must provide information about IT infrastructure constraints in the company: existing repositories, security and safety constraints.	IT descriptor (IT Infrastructure Descriptor)

Element	Generated in	Used as input in	Description	Data Format
Validated SEE configuration	Validate the SEE	Deploy the SEE	This is the SEE obtained from the confrontation of the “initial” SEE against Available Tools and IT Company Infrastructure.	SEE descriptor

Table 3.1: Platform Builder elements

Table 3.2 regroups the previous descriptors and extends the idea in order to derive additional descriptors to be used in the specification of the required elements, which compose the different phases of the Platform Builder workflow.

Descriptor name	Description	Shall detail	Comments
Business process descriptor	It is a formalized description of Business Process.	It shall detail the process activities and their sequence.	
Tailored process descriptor	It is a formalized description of tailored process. It contains: • Project data descriptor • A list of tool function descriptor	It shall be based on the CRYSTAL Platform Builder meta-model, enriching the Business Process Descriptor, linking activities to Project Data Elements and to Engineering Tool Function descriptors.	The linking information is part of the Tailored process descriptor. It is not part of the Project data descriptor or the Tool Function descriptors.
Project data descriptor	It is a formalized description of Project Data in terms of Artefacts and Roles.	This descriptor is part of CRYSTAL Platform Builder meta-model and contains two different elements: Artefact and Role. The set of information to be provided to each one of these elements is different.	Each main element of this descriptor must be linked to the activities in which it is needed. When linked to the activity, it must be detailed as inputs, outputs or internal documentation. A section defining security aspects and identifying which roles have access to which artefacts and the access level is probably necessary.
Engineering Tool function descriptor	This descriptor has to support the unambiguous identification of the engineering tool function. It also has to contain some description about it.	This descriptor shall detail the unique identifier, the name and the engineering domain of the engineering tool function.	Each engineering tool function must have a unique identifier at the RTP level. Those tool function descriptors are linked to the activities where they are used.

Descriptor name	Description	Shall detail	Comments
SEE descriptor	It contains: • Tool chain descriptor • IT descriptor	This descriptor is composed by the Tailored Process descriptor, the Tool Chain descriptor and the IT descriptor. See concerning descriptors descriptions.	This descriptor shall be defined to contain information about Tool Chain and IT infrastructure.
Tool chain descriptor	This descriptor is composed by: • A list of Engineering Tool Function descriptors • IOS descriptor	The Tool Chain descriptor in an extension of the previously obtained List of Engineering Tool Function descriptors detailing information about the IOS needs for this Engineering Tool Functions in the context of the Service Engineering Environment being modelled. See Engineering Tool Function descriptor and IOS descriptor.	In this case IOS needs, if required, must be specified by activity – engineering tool function relationship level.
Tool descriptor	It contains: • A list of Engineering tool function descriptors • IOS descriptor	Each tool descriptor must contain: Vendor, Version, IT requirements and Engineering domain. Moreover, it also contains: an Engineering Tool Function descriptor for each Tool function provided, and an IOS descriptor for defining Tool function IOS support. See Engineering Tool Function descriptor.	Both catalogues (Domain Tools and Available Tools) are a set of Tool descriptors. In the case of conflicting information between an Engineering Tool Function Descriptor and the information of the Tool descriptor where it is contained, the one defined at the Engineering Tool Function prevails. In this case IOS support must be provided at Tool Function Level.
IOS descriptor	It represents aspects of interoperability based on the IOS specification between tools in a selected tool-chain	This descriptor serves either for defining IOS needs in a Tool Chain and defining IOS support in the Tools. The fields to be supported are: IOS specifications and versions.	When used in a Tool Chain, it is related to activity – engineering tool function pairs. When used in a Tool descriptor it is applied to the Engineering Tool Functions provided by the Tool.
IT descriptor	It defines a combined set of hardware, software, networks, and services/facilities where the tool-chain will be deployed.	All the important information concerning IT aspects and useful for characterizing the infrastructure where the tool-chain will be deployed.	

Table 3.2: Platform Builder descriptors

4 CRYSTAL Platform Builder Meta-Model

In this section, the conceived CRYSTAL Platform Builder Meta-Model is presented. The idea is to get the requirements which were made available by the different partners and, from an existing baseline meta-model (SPEM 2.0 – see *References*), develop the CRYSTAL Platform Builder Meta-Model. The chapter *CRYSTAL Platform Builder Meta-Model requirements* presented these requirements whereas information about the baseline meta-model is given in this chapter. The CRYSTAL Platform Builder Meta-Model is based on the SPEM Meta-Model and it is enriched in order to exhaustively support the specification and deployment of a fully integrated tool-chain.

The establishment process of the CRYSTAL Platform Builder meta-model followed the steps below:

1. Evaluation of the baseline meta-model in order to find out if it could be used, at least partially, as the start point for the conception of CRYSTAL Platform Builder Meta-Model;
2. Evaluation and classification of available requirements and different inputs coming from several partners;
3. Identification of relevant elements for the CRYSTAL Platform Builder Meta-Model and for the Platform Builder descriptors;
4. Initial mapping of previous identified elements to the baseline meta-model;
5. Gap analysis between specific Platform Builder needs and not-covered points from the mapping process (previous step);
6. Enrichment of the CRYSTAL Platform Builder Meta-Model in order to cover missing aspects from SPEM;
7. Consolidation of the meta-model.

The result of the Meta-Model creation process is presented in this chapter under the several following sub-sections.

4.1 Meta Model Baseline

The meta-model which was selected as baseline for the CRYSTAL Platform Builder Meta-Model was SPEM 2.0. Its specification can be found under the chapter *References* together with further information about it.

SPEM stands for Software & Systems Process Engineering Meta-Model and, as its name makes explicit, it is a process meta-model with focus on engineering processes. SPEM has been selected because it is widely used for process definition, becoming a de facto standard that allows companies to define highly personalized processes. In this way, SPEM can represent the basic items that compose processes, which are used to represent some of the elements identified in the PB Meta-Model. Figure 4.1 shows SPEM structure and the relationship between the different elements/packages which compose the whole specification:

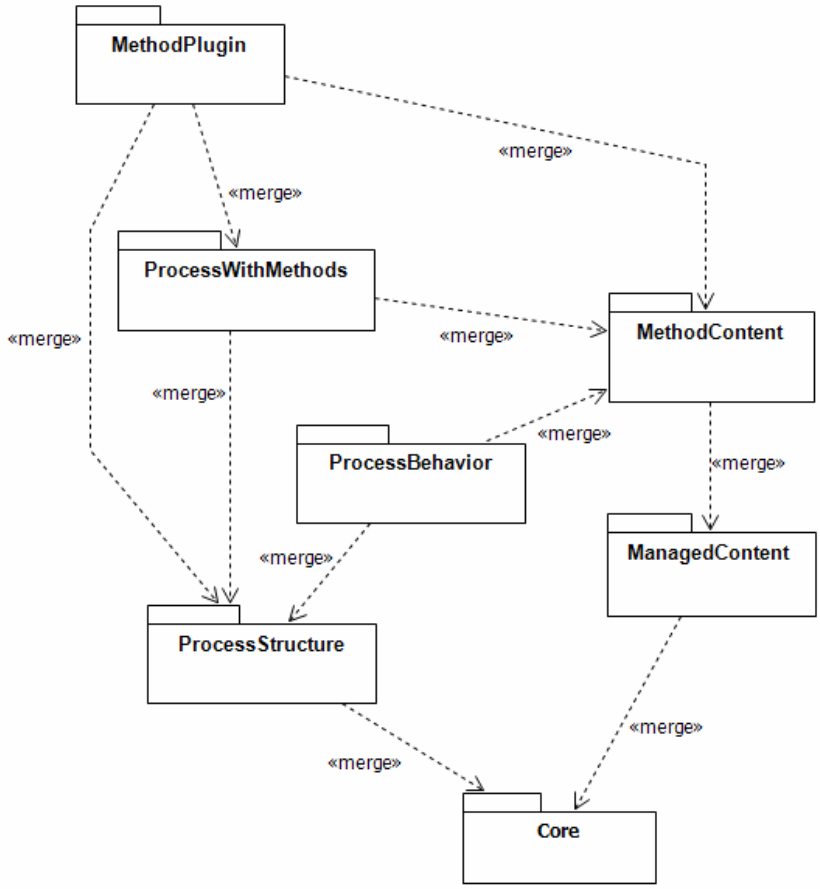

Figure 4.1: Structure of SPEM 2.0

SPEM general idea is that there are two main branches which are respectively called “Process” and “Content”. They are based on a Core specification and are developed in a way that allows users to select which packages they need/want to use, giving some freedom of application of the specification. There are also *heterogeneous* packages within SPEM structure which provide a way to integrate both branches concepts in order to multiply specification possibilities. Each existent package has its own elements which are consistent with the whole specification as well as with each other i.e. they respect the specification hierarchy and are then naturally compliant to SPEM guidelines.

4.2 Alternative Platform Builder process view

In order to represent the SEE configuration and instantiation workflow process (Figure 2.3) as a SPEM compliant *activity diagram* which explicitly presents the main required elements for the process and their relationships, the activity diagram in Figure 4.2 was created. This new diagram can also be understood as a development of the diagram presented in Figure 2.3, for here some more elements are added and specific execution steps are set.

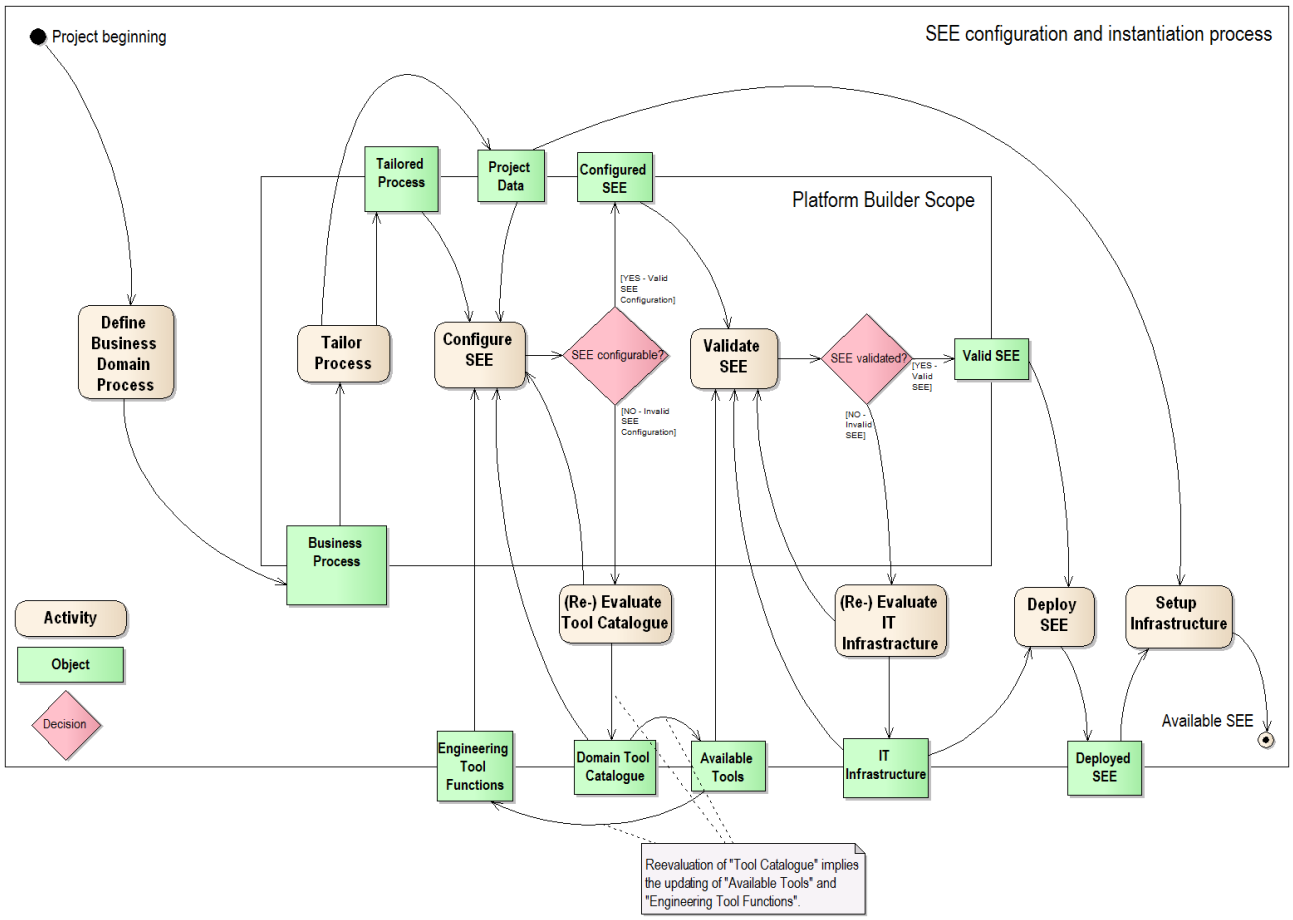

Figure 4.2: SEE configuration and instantiation activity diagram

In Figure 4.2, it is possible to recognize the workflow or sequence of activities that defines the SEE configuration and instantiation process of the Platform Builder. This diagram has a well-defined flow of activities which starts with a definition of the required Business Domain Process and has as output a configured and validated SEE. Platform Builder scope is presented in the centre of the diagram having as activities the tailoring of the process and the configuration and validation of the SEE. Some verification is made in order to check if all necessary tools or IT infrastructures aspects are available but the activities of updating or completing these objects are not necessarily in the scope of the Platform Builder. The green boxes are representing some of the elements listed in Table 3.1.

Comparing this activity diagram to the one presented in Figure 2.3, it is easy to see that some elements were added in Figure 4.2 in order to better describe the process (different activities) to be executed by the Platform Builder. These new elements (or aspects) are presented below:

- Evaluate Tool Catalogue is an activity placed between two of the main previous identified activities (Configure SEE and Validate SEE). In order to configure the SEE, it is required to know all needed Engineering Tool Functions for a given project (depending consequently on what is available in the domain tools catalogue), while for its validation it is necessary to have them available in the company (it might be necessary to ask to tools providers to upgrade the functionalities in order to cover project needs);
- Tailored Process contains Activities and their related Process Engineering Functions;
- Here, IT Infrastructure changes its meaning depending on which activity it is referred:
    - IT Infrastructure seen as input of the activity Validate SEE is the existing or updated IT infrastructure available in the company;
    - IT infrastructure as required by the process is *implicitly* described in the object Configured SEE;
    - IT Infrastructure seen as input of the activity Deploy SEE is the validated IT Infrastructure.

This diagram was created using UML language which means that it is SPEM compliant i.e. the existing elements can be mapped to SPEM.

4.3 Identified Elements for the CRYSTAL Platform Builder Meta-Model

Based on collected requirements and on the Platform Builder workflow process, the diagram of Figure 4.3 was created. This diagram represents the identified required elements for the CRYSTAL Platform Builder Meta-Model and how they are related.

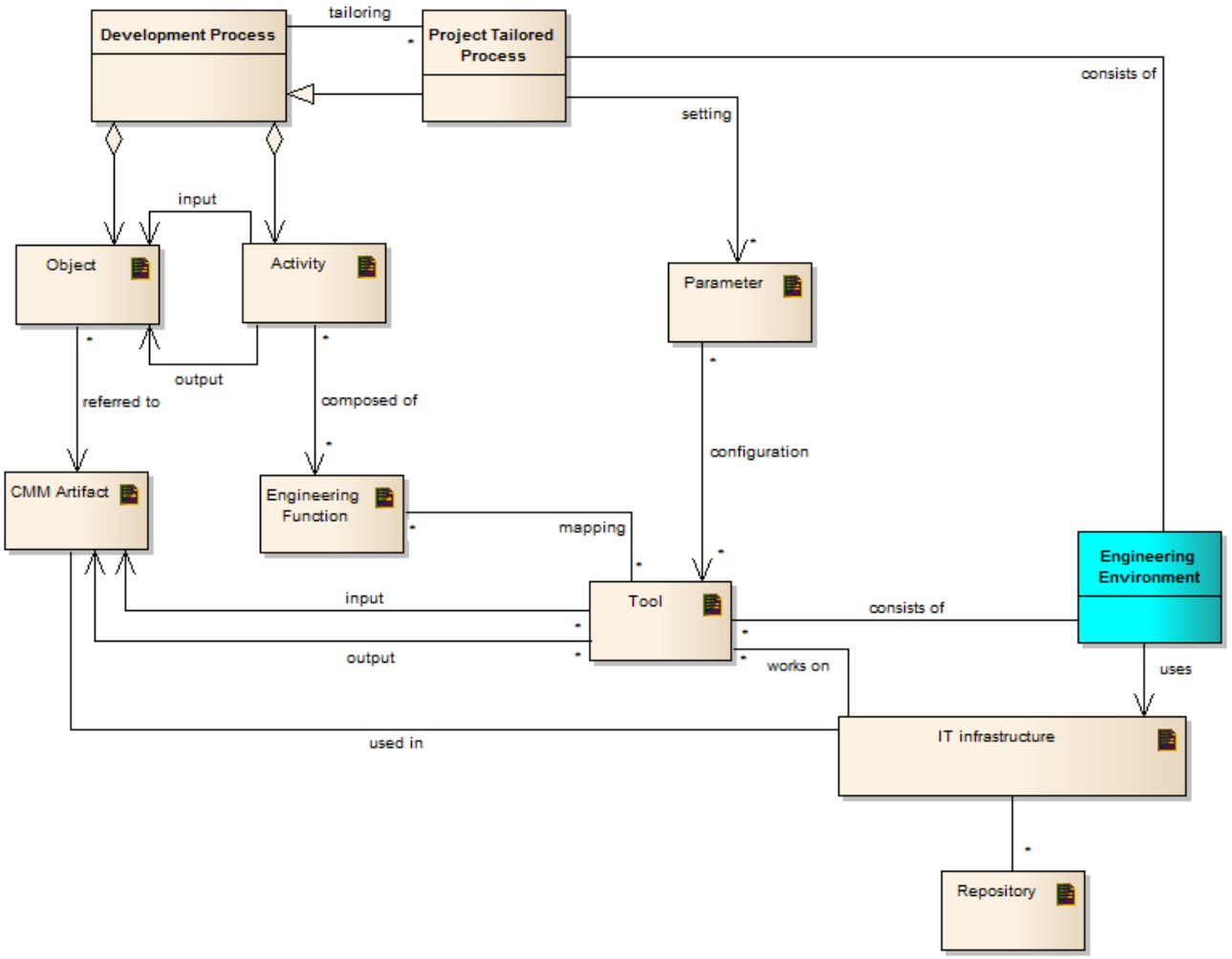

Figure 4.3: Identified required elements for the CRYSTAL Platform Builder Meta-Model

In Figure 4.3, several different artefacts are used to define the meta-model:

→ *Activity* and *Engineering Function*: the meta-model relates process activities with tools by the relation between *Activity* and *Engineering Function*. A business process defines activities to be performed for the Platform Builder. From these activities, it is then required to identify Engineering Functions. In the scope of a business process, different activities can be identified and each one might be characterized by different sequences of engineering functions, for that would depend specifically on the selected use case. Engineering function refers to an action performed by a tool and it is used to detail an activity in the business process, for an activity is a concrete work identified within a given business process and it represents a general unit of work assignable to a specific user which is logically able to perform it.

→ *System Engineering Environment*: the SEE is a framework where a tool-chain is instantiated, taking into consideration the organization IT infrastructure and the IT SEE needs. Considering the tailored process as input, a SEE descriptor has to be derived in order to contain needed information to set-up the SEE itself. Actually, needed information is sensed to be part of the tool-chain, which encompasses tools information for deployment and interoperability about information services and IT infrastructure. The SEE descriptor has then to be validated checking the tool-chain descriptor against the available tools catalogue and checking the

needed IT infrastructure descriptor against a given organization IT infrastructure. A System Engineering Environment configuration, which has its characteristics exposed in Figure 4.4, depends on:

- Process definition, that describes project data and workflow to be applied in terms of project roles, tasks, work products, tools and guidance;
- Available tools in the business domain and in the company organization;
- IT Infrastructure, that describes the environment where the RTP Instance will be deployed.

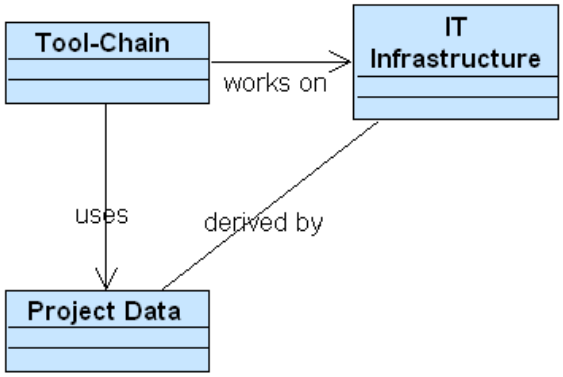

Figure 4.4: Required data to properly represent the SEE

→ *Tool* and *Tool-Chain*: Tool-chain is defined as a set of tools employed to a specific use and their interoperability aspects in the respective system life cycle. The tool-chain has to be defined by the tool-chain descriptor that contains information about different tools and interoperability aspects between them. Tool information specifies the necessary data used to classify tools in terms of provided and consumed services which are full described by the engineering tool functions. Tools catalogue is a container of tool descriptors. In the Platform Builder workflow, which is exposed in Figure 2.3, there are two different categories of tools catalogues:

- Domain tools catalogue and
- Available tools catalogue.

The domain tools catalogue contains all the identified tools available in a given domain which are able to support interoperability aspects. The available tools catalogue contains a list of the tools that are available for use by a given company and it also has relevant aspects concerning the IOS (available tools catalogue encompass description that is relevant for interoperability aspects). The associations between the different catalogues are presented in Figure 4.5.

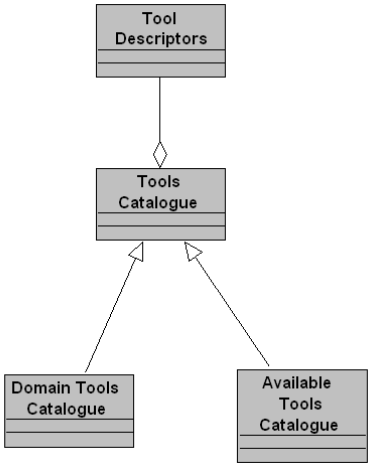

Figure 4.5: Relation between the different catalogues

The provisioning of the detailed specification for tool providers with the objective of allowing them to describe their tools in a standardized way is also foreseen within Platform Builder scope. Indeed, to stipulate a SEE configuration, the Platform Builder needs to map engineering functions, as in the tailored process, to engineering tool functions, as contained in the domain tools catalogue and defined using the meta-model (for tool description purposes). For the validation of the SEE configuration, the available tools catalogue is then applied. From the Figure 2.3, it is easy to see that the engineering functions mapping is a task required for configuring and validating the SEE. Figure 4.6 demonstrates these ideas.

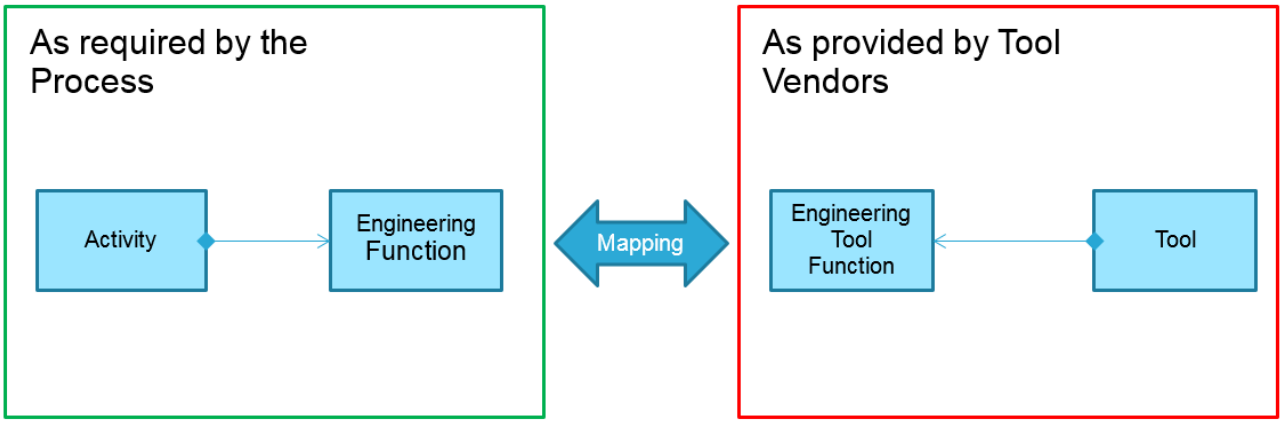

Figure 4.6: Mapping between Engineering Functions and Engineering Tool Functions

→ *IT Infrastructure:* IT infrastructure can be seen as two different-in-time elements (IT infrastructure currently available in the company and IT infrastructure required for the project) and it deals with storing, retrieving, exchanging and manipulating data. IT infrastructure addresses also aspects that are relevant to data structure, data security and user access rights, these elements being identified in the tailored process. IT Infrastructure properties have to be derived from the tailored process in order to properly describe needed repositories and their locations, data views and security and safety constraints (access rights and access control level). IT infrastructure also depends on work products (mainly for data views and repositories) and roles (mainly for access rights and access control level). The IT infrastructure descriptor has to define:

- Artefacts with properties (which and how the artefacts are used concerning data security levels, format for data views, repository location);
- Roles (define who uses the artefacts and his user rights access);
- Tools constraints (IT infrastructure has to take into account tool constraints for tool constraints could also depend on Artefacts (which they use or access). Generally, tool constraints could concern hardware properties/system properties but also network properties to access data (artefacts). Depending on the artefacts that a tool uses, the IT infrastructure could have its network and repository properties changed, which would depend on the relationship between artefacts and tools relationship).

The IT infrastructure (as required for the project) defines the IT infrastructure to where the tool-chain will be deployed. This IT infrastructure is the platform which provides the basis for the data structure, roles, applications and functionality being delivered to the end users in order to allow them to perform activities using the plugged-in “tool-chain”.

→ *Project Tailored Process:* The Project Tailored Process is the result of tailoring the project Development Process taking in consideration its specificities.

→ *Parameter:* The Project Tailored Process is able trough the Parameter artefact to configure the used tools by setting specific parameters, having an impact on the tools behaviour and consequently on the Engineering Functions.

As shown in Figure 4.7, the artefacts identified in Figure 4.3 are assumed to be inherited from a general “Platform Builder Entity” which has specific roles and contract properties as Guarantee and Assumption and can also cover, but not only, aspects as Safety and Security constraints (as needed for defining IT infrastructure properties).

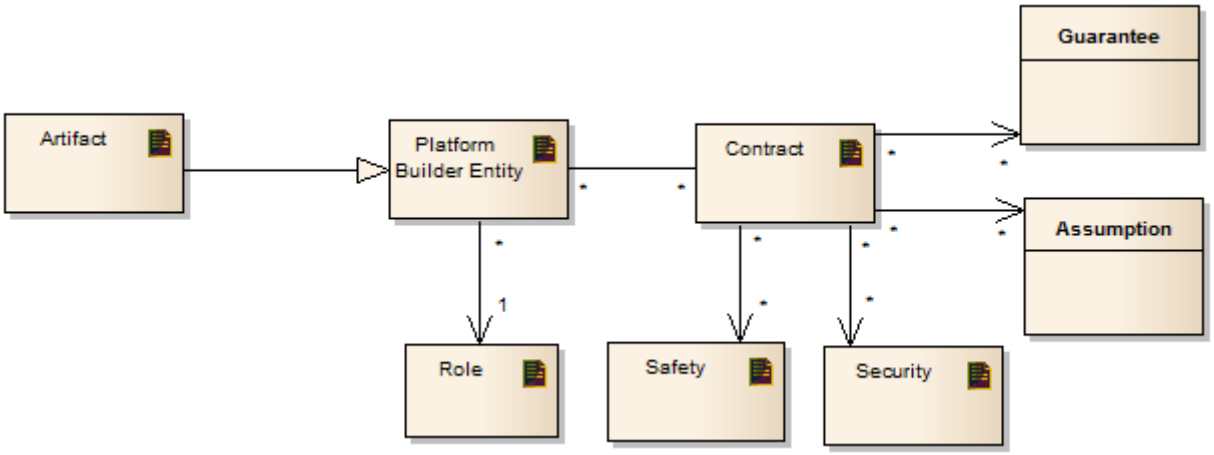

Figure 4.7: Inheritance characteristics of CRYSTAL Platform Builder Meta-Model artefacts

From the identified elements presented in Table 3.1 and the Platform Builder workflow itself (see Figure 2.3), a deeper representation of the meta-model consisting of descriptors was developed. For the System Engineering Environment configuration infrastructure as designed in the Platform Builder workflow, the descriptors properties were summarized and are presented in Figure 4.8.

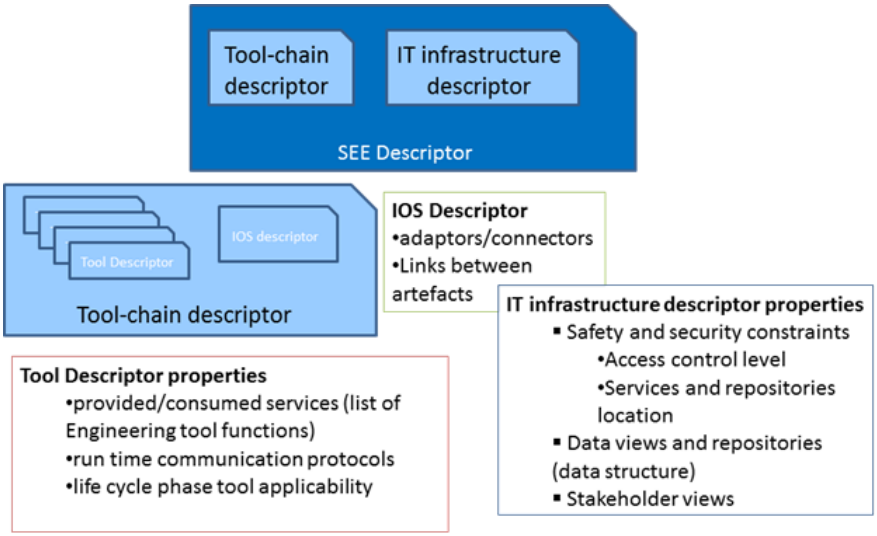

Figure 4.8: SEE configuration descriptors

The different descriptors, which are directly related to the elements of the meta-model (see Figure 4.3), are related to each other and to other artefacts as presented in Figure 4.9.

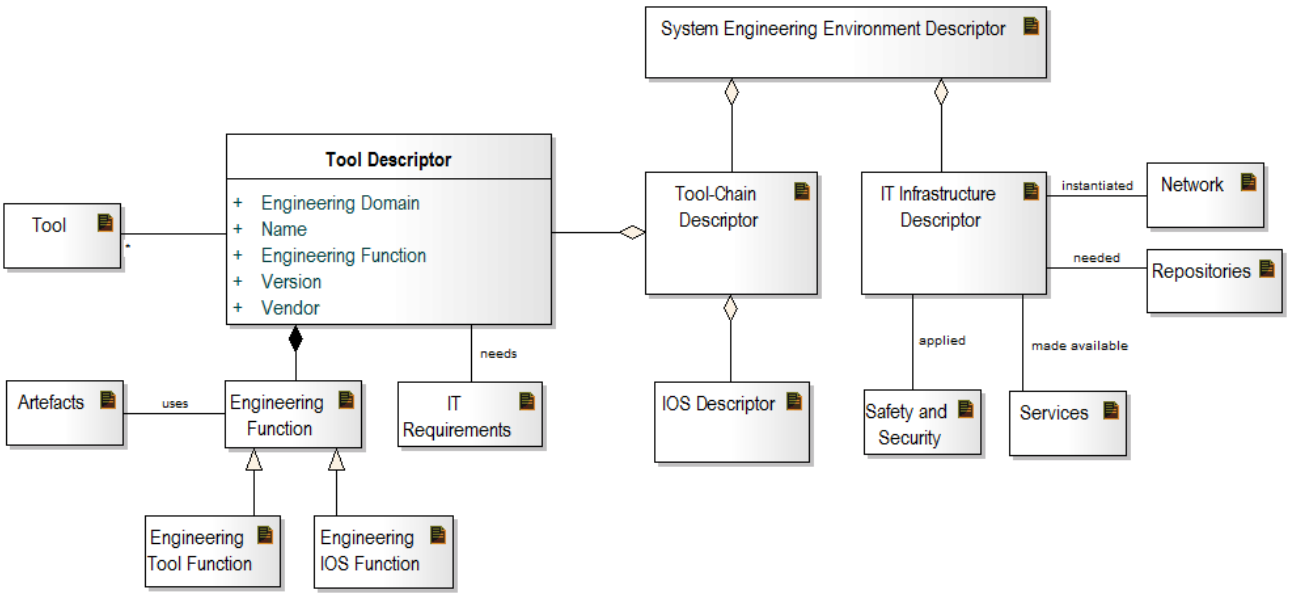

Figure 4.9: Relation between different descriptors and artefacts

A given Tool Descriptor has a set of Engineering Functions that is provided by the designated Tool. Generally, Engineering Functions use Artefacts; when Engineering Functions perform their actions on Artefacts that are produced for external use (by another Tool for instance) then these Engineering Functions are called Engineering IOS Functions.

4.4 Mapping identified elements to SPEM

Once the relevant elements were identified, a mapping of these elements to SPEM was realized. The main objective of this mapping was to find out if the elements needed for the CRYSTAL Platform Builder Meta-MODEL were covered by the elements present in SPEM or if a further enrichment and expansion of SPEM were needed in order to cover all the missing aspects of the Platform Builder needs.

The next sections present the performed mapping both in relation to the Process and Content packages.

4.4.1 Mapping concerning process aspects (Process Structure package)

A preliminary mapping and gap analysis was performed concerning the process point of view. The basic employed approach in the execution of the mapping was to study the existing elements that compose the SPEM Process Structure package and then try to fit the CRYSTAL Platform Builder Meta-Model elements to them. Figure 4.10 shows the original studied package structure:

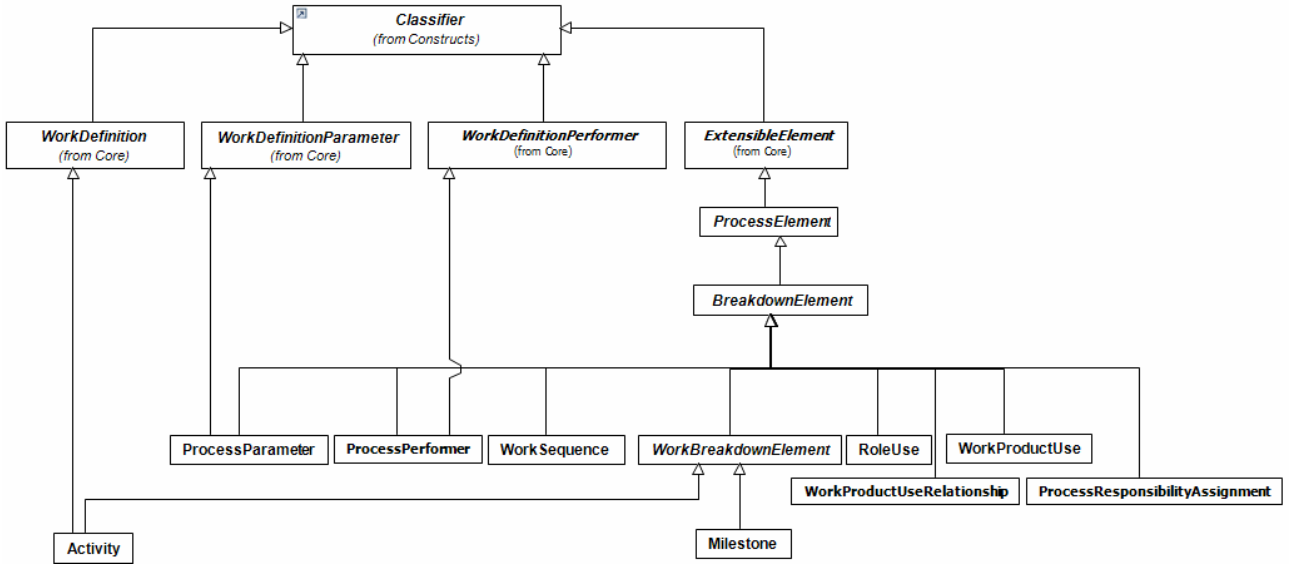

Figure 4.10: Complete taxonomy of all process elements defined in the Process Structure package

The performed mapping can be seen in Figure 4.11.

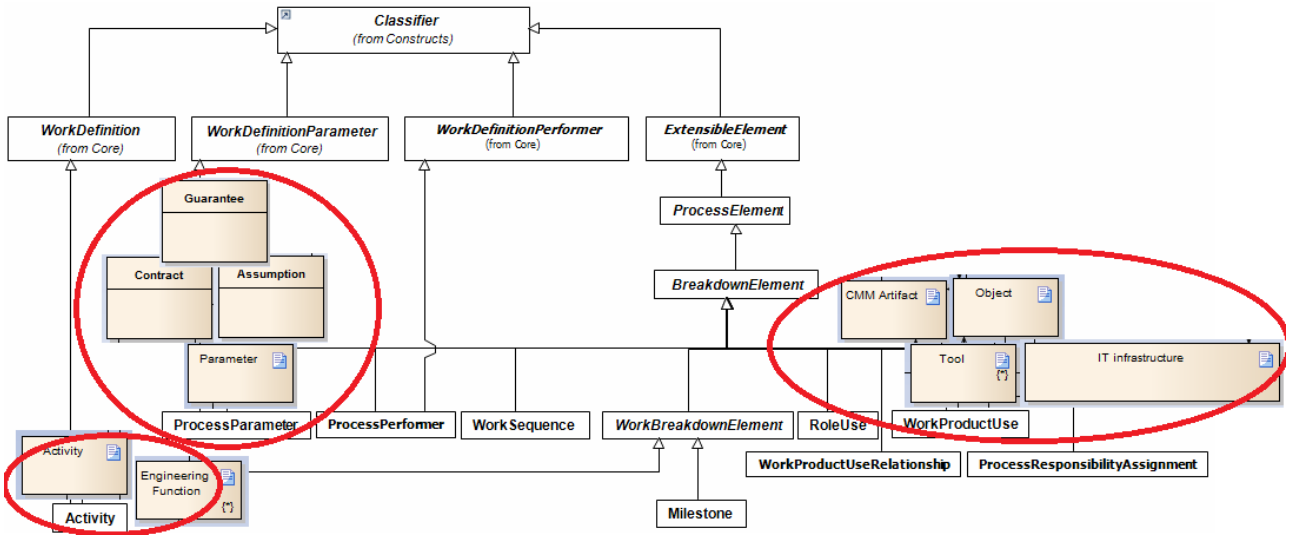

Figure 4.11: Mapping CRYSTAL Platform Builder Meta-Model elements to the Process Structure package

The following short descriptions that come from SPEM specification support the choices of such mapping:

- **Activity:** "An *Activity* is a Work Breakdown Element and Work Definition that defines basic units of work within a Process as well as a Process itself. In other words, every Activity represents a Process in SPEM 2.0."
  - *Activity* and *Engineering Function* were mapped to *Activity* in SPEM Process Structure package.
- **Process Parameters:** "A *Process Parameter* is a Work Definition Parameter and Breakdown Element that is used for process definitions. It defines input and output meta-types to be Work Product Uses."
  - *Assumption*, *Contract*, *Guarantee* and *Parameter* were mapped to *Process Parameter* in SPEM Process Structure package.
- **Work Product Use:** "A *Work Product Use* is a special Breakdown Element that either represents an input and/or output type for an Activity or represents a general participant of the Activity. If it is an input/output, then the Work Product Use needs to be related to the Activity via the Process Parameter class."
  - *CMM Artefact*, *IT Infrastructure*, *Object* and *Tool* were mapped to *Work Product Use* in SPEM Process Structure package.

4.4.2 Mapping concerning methods aspects (Method Content package)

Following the same idea from the previous section, a preliminary mapping and gap analysis was performed concerning this time the method point of view. Again, the employed approach consisted in executing the mapping by starting the examination of the existing elements that compose the SPEM Method Content package and then trying to fit the CRYSTAL Platform Builder Meta-Model elements to them. Figure 4.12 shows the original studied package structure:

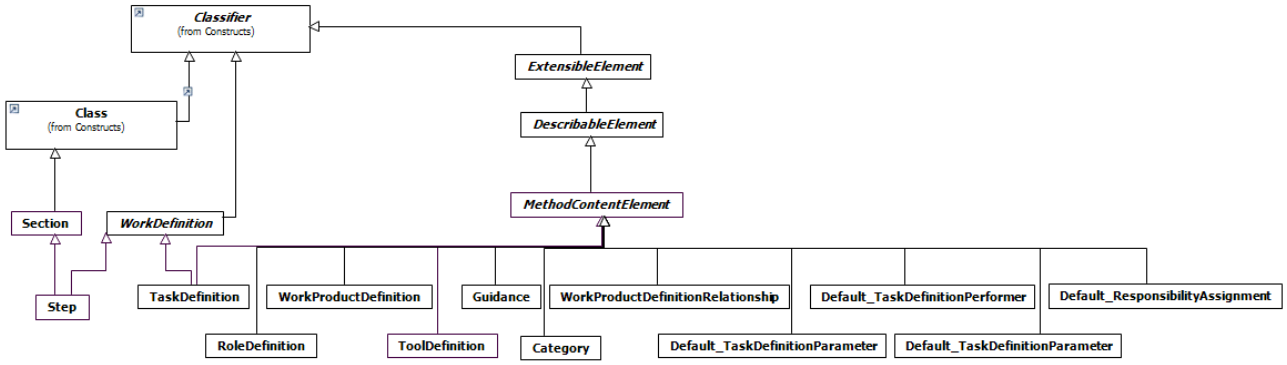

Figure 4.12: Complete taxonomy of the elements defined in the Method Content package

The resultant mapping can be seen in Figure 4.13.

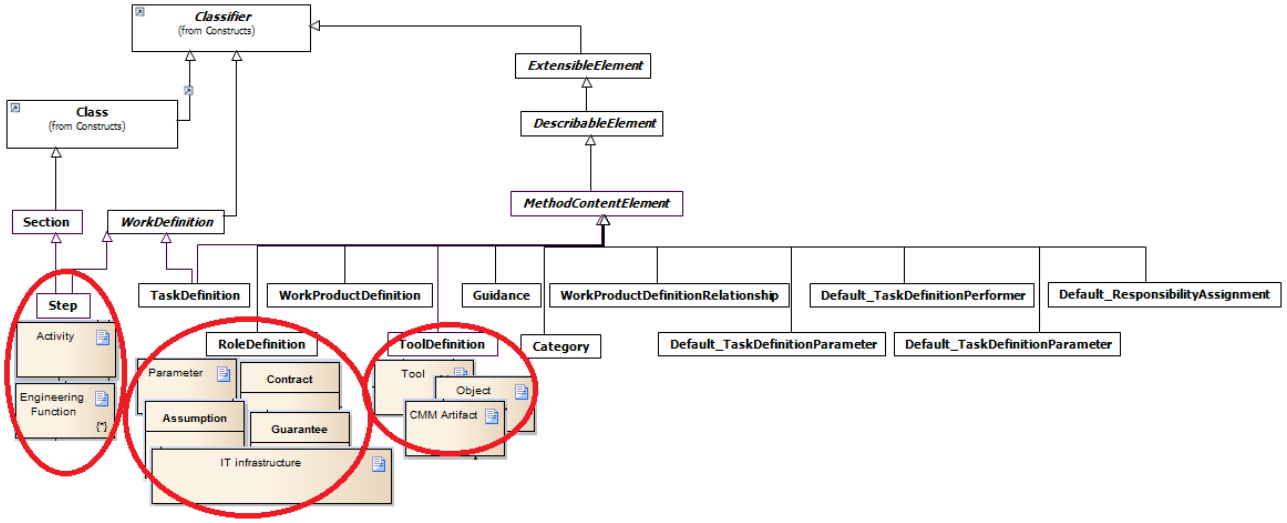

Figure 4.13: Mapping CRYSTAL and SPEM MMs concerning the Method Content package

The following short descriptions that come from SPEM specification support once again the choices of such mapping:

- **Role Definition:** "A Role Definition is a Method Content Element that defines a set of related skills, competencies and responsibilities. Roles are used by Task Definitions to define who performs them as well as to define a set of Work Product Definitions they are responsible for."
    - *Assumption*, *Contract*, *Guarantee*, *IT Infrastructure* and *Parameter* were mapped to *Role Definition* in SPEM Method Content package.
- **Step:** "A Step is a Section and Work Definition that is used to organize a Task Definition's Content Description into parts or subunits of work. Steps inherit the subSection decomposition from Section and can therefore describe sub-Steps nested into Steps."
    - *Activity* and *Engineering Function* were mapped to *Step* in SPEM Method Content package.
- **Task Definition:** "A Task Definition is a Method Content Element and a Work Definition that defines work being performed by Roles Definition instances. A Task is associated to input and output Work Products."

- **Tool Definition:** “A *Tool Definition* is a special Method Content Element that can be used to specify a tool’s participation in a Task Definition.”
    - *CMM Artefact*, *Object* and *Tool* were mapped to *Tool Definition* in SPEM Method Content package.

5 Glossary

Term	Description	Term/ description inherited from
Activity	Activity is a concrete work identified within a Business Process and it represents a general unit of work assignable to specific user able to perform it. An activity is detailed using Engineering Functions.	WP 602 Glossary
Available Tools/ Available Tools Catalogue	Available Tools are the tools in the company which have relevant aspects to the implemented IOS.	
Business Process	It is the system life cycle development and management process that represents a whole given process, covering development phase and product management phase. Life cycle management comprises tasks of product management process as: traceability, configuration and versioning, user and process management. In the other hand, life cycle development covers tasks of development process (as requirements elicitation and formalization, design, implementation, validation and verification).	
Domain Tools Catalogue	Domain Tools Catalogue contains all the Tools adapted to work within the IOS in CRYSTAL RTP.	
Engineering Functions	Engineering Function is a detailed function of an Activity and it refers to an action performed by means of a Tool.	WP 602 Glossary
Engineering Tool Function / Tool Function	Engineering Tool Function is an engineering function that a Tool is able to provide.	WP 602 Glossary
Engineering IOS Function	Engineering IOS Function is an engineering function that foresees interoperability between tools.	
IT Infrastructure	IT infrastructure is a combined set of hardware, software, networks, and services/facilities where a tool-chain is deployed and needed network is instantiated, services are made available, and security and safety constraints are applied. There are three different IT infrastructure descriptions: IT infrastructure as required by the process, available IT infrastructure in the company and validated IT infrastructure.	Partially from WP 602 Glossary
Plugged-in Tool-Chain	It is the instantiation of a tool-chain within the IT infrastructure.	
Project/Process Data	SEE framework shall be tailored using process and project data from specific use cases. System Engineering Environment is a Collaborative Engineering Development Environment guided by a defined process tailored for a project (from a specific use case). Project Data are relevant data to the project which is necessary to setup up the SEE. Project Data are Work Products and Roles defined in the Tailored Process. Roles will be assigned to Users identified within a	Partially from WP 602 Glossary

	company organization which (Users that play a Role) uses the SEE.	
Role/Role Use	A Role Use represents a Role in the context of one specific Activity. Every breakdown structure can define different relationships of Role Uses to Task Uses and Work Product Uses. Therefore, one role can be represented by many Role Uses, each within the context of an Activity with its own set of relationships. Every Role Use can reference only one Role Definition. However, a Role Definition can be represented by many Role Uses. A Role Use can select a sub-set of valid Qualifications defined for the Role Definition for this one use of the Role Definition in the context of a particular Activity.	SPEM 2.0
RTP Instance	RTP Instance is a set of *Engineering Tools* and *Basic Services* interacting together according to a domain-specific application/product development process throughout the *Engineering and Lifecycle Management Phases.* RTP Instance definition is a tool-chain with interoperability aspects defined within the System Life Cycle. Tools could be Development Tools and/or Management Tools. Life Cycle Management Tools are involved in the tasks of managing process (i.e. traceability, configuration and versioning, user and process management) while Development Tools are involved in tasks of developing process (i.e. requirements elicitation and formalization, design, implementation, validation and verification).	CESAR Project
SEE / SEE configuration	System Engineering Environment is a framework where a RTP instance will be instantiated within the organization's IT infrastructure to perform a process relevant to a project scope. This framework will be set up with the project's data. SEE in the Platform Builder can be configured or validated.	
Tailored Process/ Tailored Project Process	Tailored Process has to define all aspects relevant to a specific Use-Case and it defines in detail the workflow in terms of: Tasks to be performed and their sequence, work products to be produced, Roles that perform tasks, and tools to be used to perform tasks. Tailored process represents an input of the Platform Builder workflow.	
Tool	This concept is used for referring to a Software Tool that is used in the development process of critical embedded systems.	
Tool Chain	Tool chain is the set of tools and their interoperability aspects in the system life cycle. Tools could be Development Tools and Management Tools. Life cycle Management tools are involved in the tasks of management process (i.e. traceability, configuration and versioning, user and process management) while Development tools are involved in tasks of development process (i.e. requirements elicitation and formalization, design, implementation, validation and verification).	WP 602 Glossary
Tools Catalogue	Two different tools catalogue exist in the Platform Builder Process. See Available Tools Catalogue and Domain Tools	

	Catalogue.	
Validated SEE configuration		
Work product	Work Products are in most cases tangible work products consumed, produced, or modified by Tasks. They may serve as a basis for defining reusable assets. Roles use Work Products to perform Tasks and produce Work Products in the course of performing Tasks. Work Products are the responsibility of Role Definitions, making responsibility easy to identify and understand, and promoting the idea that every piece of information produced in the method requires the appropriate set of skills. Even though one Role Definition might “own” a specific type of Work Product, other roles can still use the Work Product for their work, and perhaps even update them if the Role Definition instance has been given permission to do so.	SPEM 2.0

Table 5.1: Terms, Abbreviations and Definitions

6 References

[Author, Year]	Authors; *Title*; Publication data (document reference)
[OMG,2008]	OMG – Object Management Group; SPEM 2.0 – Software & Systems Process Engineering Meta-Model Specification Version 2.0; April 2008 (http://www.omg.org/spec/SPEM/2.0/PDF)