D5.2 FINAL STANDARD DATA MODEL

Eurostep

31-08-2017

This project has received funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement No 603843

Dissemination Level:

☒ PU Public
☐ PP Restricted to other program participants
☐ RE Restricted to a group specified by the consortium
☐ CO Confidential, only for members of the consortium

Lead beneficiary	Eurostep AB
Author/s	Xin Ye
Reviewed	Torbjörn Holm, Christoph Velte, Patricia van Loon, Farazee Asif
Approved	Amir Rashid

Project Number	603843
Project Acronym	ResCoM
Project Title	Resource Conservative Manufacturing - transforming waste into high value resource through closed-loop product systems
Start Date of Project	1 November 2013
Duration	48 months
Programme / Topic	FP7 / ENV.2013.6.3-1
Type of activity	Other
Deliverable reference number	603843 /
WPs contributing to the deliverable	WP7
Due date	31st Aug 2017
Actual submission date	
Revision	
Abstract	This is a documented and ResCoM verified domain information model based on ISO 10303-239 and OASIS PLCS ed.2 standard supporting the envisioned ResCoM processes and information. Similar as in deliverable D5.1, it also include a defined terminology and UML class diagrams defining the used information structures.
Key Words	Data model, Product multiple lifecycle management, PLCS, standard, integration

Table of Contents

1. ResCoM standard data model

Defining and implementing data models to support closed-loop multi-lifecycle product definitions based on ISO 1303 standards is one of the most important tasks of WP5 collaborative software platform. This deliverable is a documented and ResCoM-verified (through developments in WP2, WP3, WP4 and WP6) domain information model based on ISO 10303-239 and OASIS PLCS ed.2 standard and supporting the envisioned ResCoM processes and information. Similar as in deliverable D5.1, it includes a defined terminology and UML class diagrams (the introduction of the SysML block definition diagram was provided as Annex in D5.1) defining the used information structures.

1.1 Introduction

ResCoM aims to shift the current open loop manufacturing model to a seamless closed loop manufacturing model, which enables a product to have multiple lifecycles for maximizing the utilization of raw material, minimizing the consumption of energy and recapture the utmost value-added. While implementing a series of meticulous and collaborative works of product design, business model, closed loop supply chain and remanufacturing, numberless information will be generated from the collaborative work by using different IT tools which include design and analysis tools for multiple lifecycle product. Hence, a wide range of product-related information has to be tracked and archived properly throughout the lifecycles of a product for the closed loop manufacturing. Therefore, a collaborative software platform is needed for implementing the ResCoM approach from the information management point of view. The overall solution architecture of the ResCoM collaborative software platform is shown in the figure 1.1.

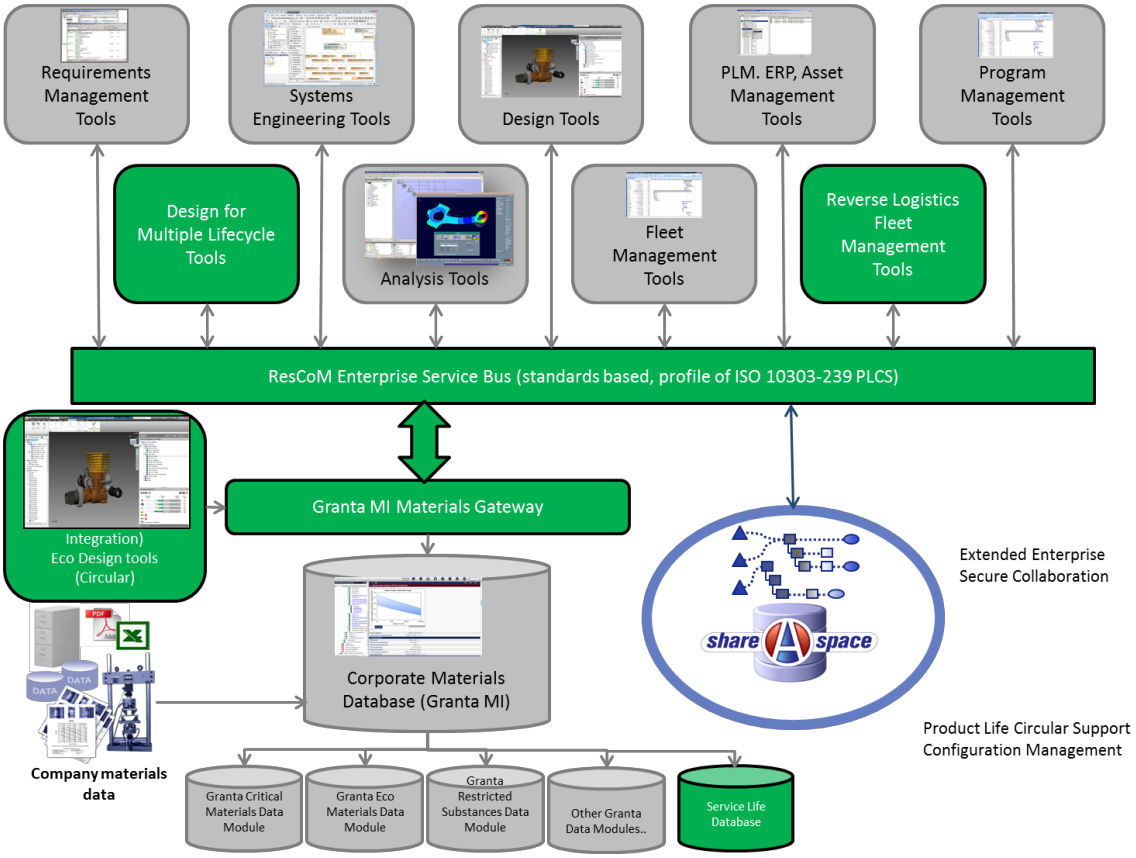


Figure 1.1 Overall solution architecture of the ResCoM collaborative software platform (In green: new components or new interfaces developed in the ResCoM project)

The platform is based on the ShareAspace, an IT tool for secure PLM collaboration that is developed further to support closed loop multiple lifecycle applications. It provides two functions: product life circular support configuration management (i.e. product multiple lifecycle management) and extended enterprise secure collaboration (i.e. tools integration, the tools including the decision making tools used/developed by ResCoM researchers and other tools used by 4 OEMs). It manages all the product-related information from the early conceptual design phase to the end-of-life (EOL) of a product.
In the platform the data and information has to be well-structured, thus data models are needed that provide a standard for mapping and structuring the data. By having such standard data models, the following benefits can be obtained:

- Clearer scope: the data models help business sponsors and developers to agree on precisely what data and information has to be managed in the ResCoM project.
- Better documentation: the data models document important concepts of ResCoM and provide a basis for long-term maintenance and efficient communication.
- Higher data quality: the data models structure the data in a logical way, so to keep a "single source of truth (SSOT)" of the data.
- Faster performance: a well-constructed database typically runs fast.
- Better applicability: the data models specify the rules for structuring data for different products and they are applicable to all products that apply the ResCoM approach.

The data models are the basis of the ResCoM collaborative software platform development, including developing the solution for managing products with multiple lifecycles and developing the ResCoM Web API and mappers for integrating ResCoM tools to the platform. These models were validated using the platform and industrial test cases.

1.2 Standards-based data model

The development of ResCoM data model was started from industrial proven life cycle standards for product life cycle information management i.e. ISO 10303- 239 Product Life Cycle Support

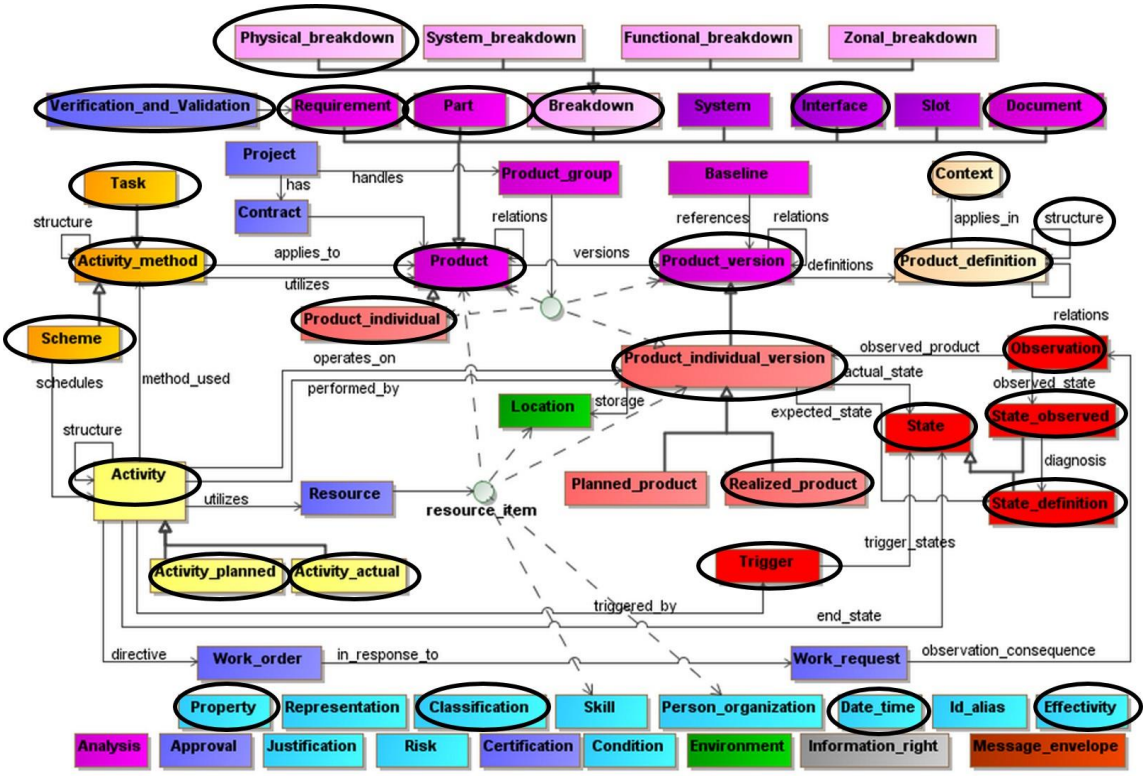


Figure 1.2 PLCS concept model

and ISO 10303-233 System Engineering, so that the final data model is ready for communication with the OASIS and ISO standards. The main concepts which are outlined in the PLCS concept model (figure 1.2), influence the ResCoM data model.

1.3 ResCoM requirements on the data model

The data models have to support closed-loop multi-lifecycle product definitions, that is, to support the envisioned ResCoM processes and information. The data model aims to specify an information environment for the integration and exchange of the data that are required by the processes described in ResCoM requirement framework. The requirements were collected from WP2, WP3, WP4, and the data mode was applied to the ResCoM collaborative software platform and finally verified and evaluated by WP6. The highlevel requirements on the ResCoM collaborative software platform are:

- Decision making tools should support decisions such as component-level recycling vs. product-level recycling based on the analysis of the product design and the reverse value chain capabilities.
- Modules should include new processes linked to remanufacturing (disassembly, cleaning, testing) and integrated (forward + reverse) supply chains
- Product changes should be traced during the multiple lifecycles (upgrades or customer own configurations that may not be obvious from the outside, but have to be known so the products can be correctly processed upon return)
- Product evolution throughout the multiple lifecycles needs to be traceable and documentable (product-level recycling = remanufacturing, then component-level recycling, then material-level recycling)

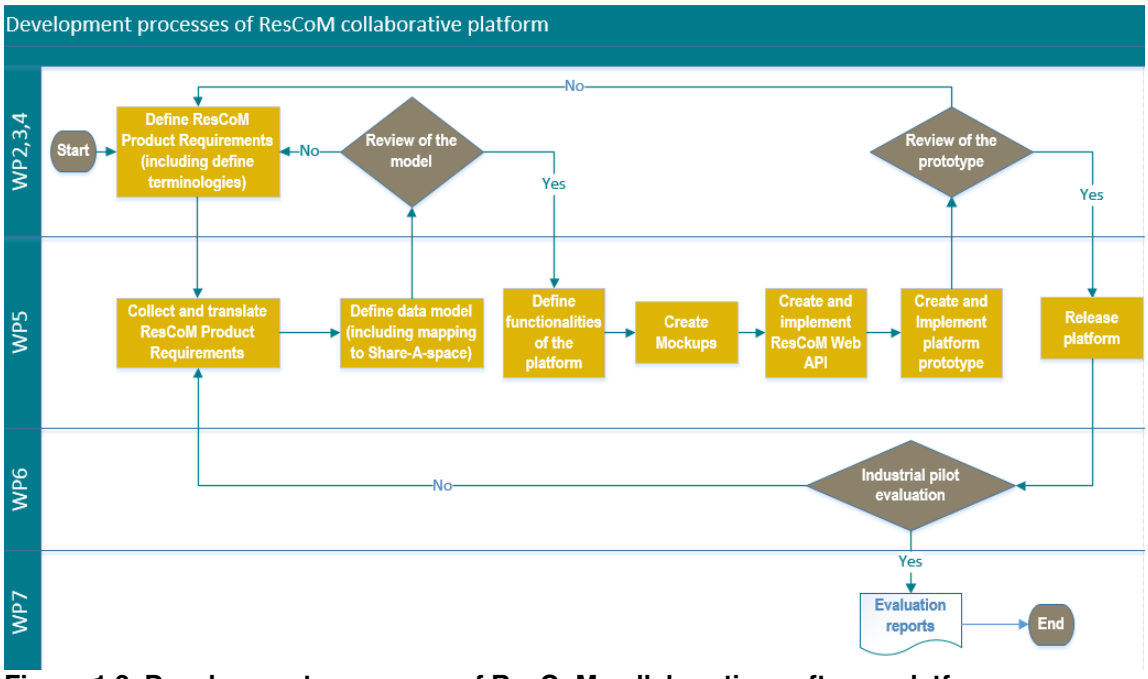


Figure 1.3 Development processes of ResCoM collaborative software platform

The development processes of ResCoM collaborative software platform is shown in figure 1.3. From the processes one can see that the data models are the basis of the ResCoM collaborative software platform development, including the development of a solution for managing products with multiple lifecycles based on ShareAspace (the collaborative PLM software of Eurostep), the ResCoM Web API, and mappers for integrating ResCoM tools to the software platform. It was an iterative process and 3 iterations were done for developing the data model: In the first iteration, D5.1 initial standard data model was created. In the second iteration

the data model was further developed to support modularization, and the data model was tested and validated through an integration with a standalone modularization tool called PALMA. This configuration software is not a “ResCoM tool” to be included in the ResCoM software platform, but it tested and proved that the ResCoM software platform has the capability to integrate any other standalone modularization tool based on the data model. In the third iteration the data model was updated according to the feedback of the industrial pilot validation and evaluation.

2. Final standard data model

In this chapter the final standard data model is documented. The model also includes a defined terminology and UML class diagrams defining the used information structures. The data model consists of four distinct domains, and each of them focuses on different aspect of the model and is represented by one SysML block definition diagram: ResCoM_Requirements, ResCoM_ProductRCBreakdown_and_Design, ResCoM_InLife, and ResCoM_Activities_and_Methods.

2.1 ResCoM_Requirements

The ResCoM_Requirements model in figure 2.1 aims to collect and store the ResCoM requirement on the Design of the multi-life product including business requirements, product design requirements, closed-loop supply chain requirements and information technology requirements. Those requirements are supposed to be listed, stored and structured in some requirement management tools and then imported to the collaborative software platform or

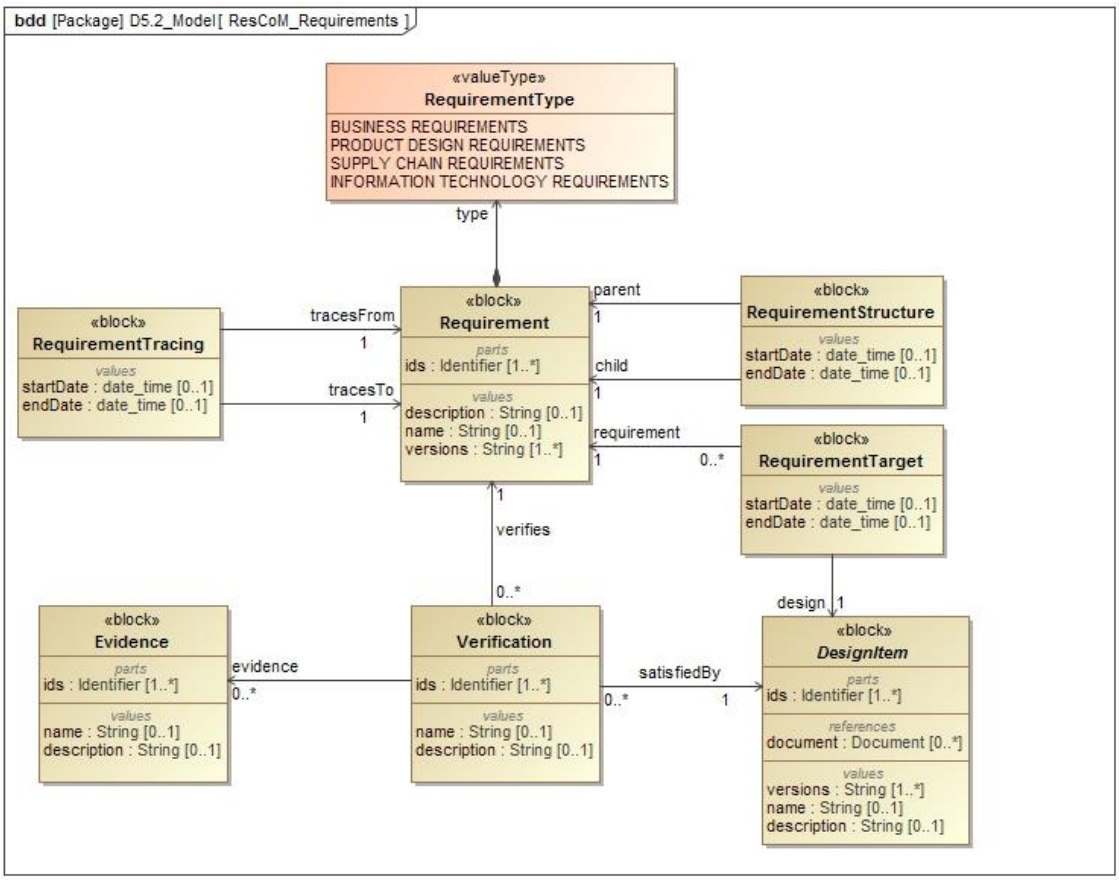


Figure 2.1 ResCoM_Requirements model as a SysML block diagram

directly be created in the collaborative software platform.
As a result one can easily document and trace a requirement (**Requirement** in the data model) in the platform, and define:

- other requirements which are relating or related to this requirement (**RequirementStructure** and **RequirementTracing** in the data model),
- the object e.g. product or components (***DesignItem*** in the data model) that the requirement is target to,
- the verifications (**Verification** in the data model) applied to verify the requirement,
- the evidence (**Evidence** in the data model) that support the verifications.

Note that a block can participate in several SysML block definition diagrams for different purposes, e.g. the "*DesignItem"* block, which has been shown in this diagram, also appears in other diagrams to represent relationships with other blocks.
The concepts, relationships and properties of the ResCoM_Requirements model are described in Table 1.

Table 1. The concepts (block), relationships and properties of the ResCoM_Requirements model

Block / Relationship	Description	Concept Reference
«block» **Requirement**	A statement that identifies a necessary capability, function, characteristic, or quality of a target item, e.g. a product or service. A Requirement is used to uniquely identify a requirement.	• ResCoM: ResCoM product specific requirements [D2.3] • PLCS: Requirement [1] • System Engineer : Requirement [2]
Relationship: type	A composition represents that one Requirement is classified as one valueType: RequirementType.	
«block» **RequirementTracing**	A RequirementTracing shows tracing from one Requirement to another.	• ResCoM: essential and critical factors [D4.1 and D4.2] • PLCS: TracingRelationship, Effectivity and EffectivityAssignment [2]
Relationship: tracesFrom, tracesTo	An association represents tracing relationship from one Requirement (TracesFrom) to another Requirement (TracesTo).	
«block» **RequirementStructure**	Relationship between a parent and child Requirement.	• ResCoM: ResCoM product specific requirements [12] • PLCS: RequirementCollection-Relationship, Effectivity and EffectivityAssignment [2]
Relationship: child, parent	Associations between one Requirement and its constituent lower level Requirement.	
«block» **RequirementTarget**	A RequirementTarget is used to relate a Requirement to *DesignItem* which are affected by the Requirement.	• ResCoM: ResCoM product specific requirements [D2.3] • PLCS: RequirementAssignment, Effectivity and EffectivityAssignment [1]
Relationship: requirement	An association represents zero or multiple RequirementTarget is associated with one Requirement.	

Relationship: design	An association represents one RequirementTarget targets to one *DesignItem*.	
«block» ***DesignItem***	An abstract class. The identification of a design, a generalization of the design objects. It is a collector of data common to all revisions of the *DesignItem*.	• PLCS: Product [1]
«block» **Verification**	A Verification is an objective assertion that a claim that a requirement is satisfied by a particular item is. Verification ensures that the specified requirements have been met. Verification uses the methods of Test, Analysis, Inspection, Demonstration, and Similarity.	• PLCS: Verification [1]
Relationship: satisfiedBy	An association represents zero or multiple Verification claims that a Requirement is satisfied by a particular *DesignItem*.	
Relationship: verifies	An association represents zero or multiple Verification claims that a Requirement is satisfied by a particular *Designitem* has been verified.	
Relationship: evidence	An association represents zero or multiple Evidence supports one Verification.	
«block» **Evidence**	An Evidence is a collector of items used together to provide a single piece of evidence within a Verification.	• PLCS: Evidence [1]
Property	**Description**	**Comment**
«Value Type»		
RequirementType: BUSINESS REQUIREMENTS	Requirements on the service models that supports product returns to OEMs	ResCoM concept [DOW]
RequirementType: PRODUCT DESIGN REQUIREMENTS	Requirement on the product design for multiple lifecycles	
RequirementType: SUPPLY CHAIN REQUIREMENTS	Requirements on integrating forward and reverse supply chains solutions that can handle the dynamics of multiple lifecycles.	
RequirementType: INFORMATION TECHNOLOGY REQUIREMENTS	Requirement on the collaborative product lifecycle management software platform supporting multiple lifecycles.	
parts		
ids : Identifier [1..*]	A set of Identifiers for the Requirement /*DesignItem*/Verification/Evidence.	An identifier provides the identifying code for the product data.
values		
version : String [1..*]	A version identifies a version of a Requirement/*DesignItem*.	A version serves as the collector of the data characterizing a realizable object in various application contexts.
name : String [0..1]	A name provides the identifying name in terms of String for the Requirement /*DesignItem*/Verification/Evidence.	
description : String [0..1]	A set of text based descriptions of the Requirement /*DesignItem*/Verification/ Evidence.	

startDate : date_time [0..1]	A date is assigned to a relationship/structure between two objects showing when this relationship/structure begins.	Relationship/structure between: • Requirement (TracesFrom) and Requirement (TracesTo) • Requirement and its sub requirements • Requirement and its target *DesignItem*
endDate : date_time [0..1]	A date is assigned to a relationship/structure between two objects showing when this relationship/structure end up.	

2.2 ResCoM_ProductRCBreakdown_and_Design

The ResCoM_ProductRCBreakdown_and_Design model in figure 2.2 aims to capture and store the information regarding the lifecycle definitions and product design in the conceptual design and engineering design phase. Those information are supposed to be stored and structured in some design tools and then imported to the collaborative software platform or directly be created in the collaborative software platform.

As a result one can easily document and check the product design (***DesignItem*** in the data model) and product lifecycles definition (**ResourceConservativeBreakdown** in the data model) in the platform, and define:

- a product design (**ProductDesign, Material** and **Substance** in the data model) from the multi-lifecycle viewpoint,
- a product design is developed by the modular function deployment (**ModuleSystem, ArchitectureNode, ArchitectureNodeStructure, Module, ModuleVariant, Interfact, StandardInterface, InterfaceType, ProductProperty and GoalValue** in the data model),
- structure (**DesignStructure** in the data model) of the product, as well as the attributes e.g. mass and quantity in the structure (**Property** in the data model)
- the lifecycle definitions of a product (**ResourceConservativeLevelDefinition** in the data model) and product structure in each lifecycle (**RCLDefinitionStructure** and **RCLElementStructure** in the data model)

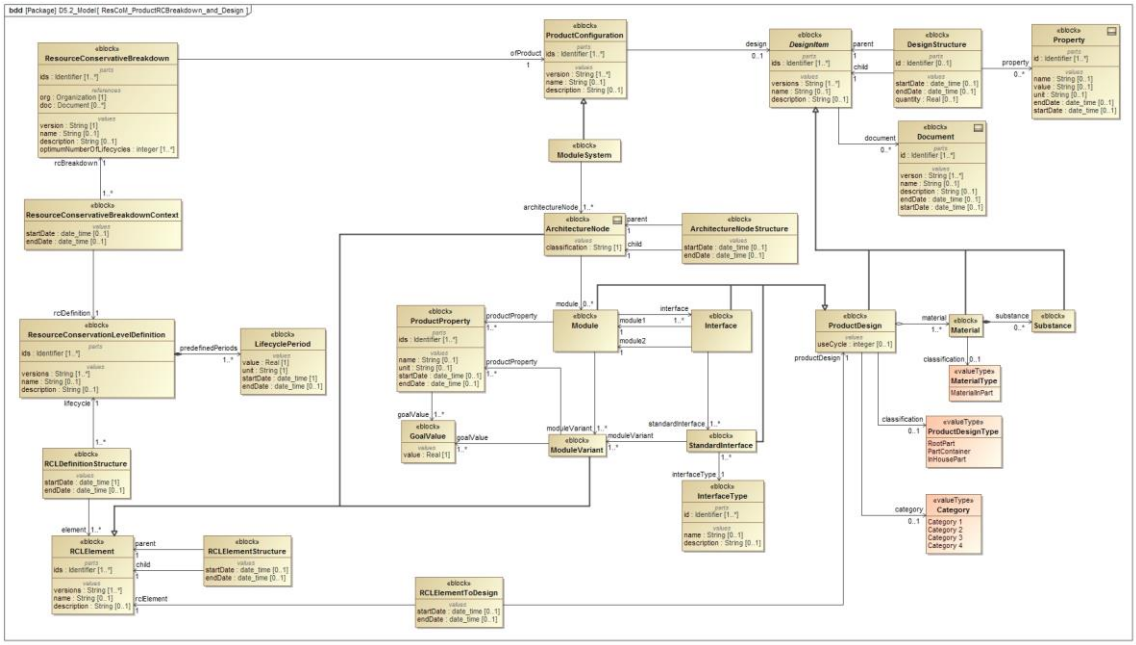


Figure 2.2 ResCoM_ProductRCBreakdown_and_Design as a SysML block diagram

The concepts, relationships and properties of the ResCoM_ProductRCBreakdown_and_Design model are described in Table 2.

Table 2. The concepts, relationships and properties of the ResCoM_ProductRCBreakdown_and_Design model

Block / Relationship	Description	Concept reference
«block» **ProductConfiguration**	The identification of a product concept (multiple lifecycles design) as a configuration.	• ResCoM: Standardization and coompatibility, upgradability and adaptability [D2.3] • PLCS: ProductConfiguration [1]
Relationship: design	An association represents that one ProductConfiguration is associated to zero or one *DesignItem*.	
Relationship: *Generalization* Note: a child block inherits all properties that its parent block has, but may have additional properties that make the child block special.	A generalization where ProductConfiguration is the parent block and ModuleSystem is the child block. It represents that one ProductConfiguration could have one specialization which is ModuleSystem.	
«block» **ResourceConservativeBreakdown**	A ResourceConservativeBreakdown provides a mean to subdivide a RCP (Resource Conservative Product) into a set of related RCLs (Resource Conservative Levels) to which additional information can be attached. This usually takes the form of a tree.	• ResCoM: RCP [DOW] • PLCS: Breakdown [1]
Relationship: ofProduct	An association represents that one ResourceConservativeBreakdown is a breakdown of one ProductConfiguration (product concept).	
«block» **ResourceCoversativeBreakdownContext**	Membership relationship between ResourceConservativeLevelDefinition and ResourceConservativeBreakdown of which the definition is a member.	• PLCS: BreakdownContext, Effectivity and EffectivityAssignment [1]
Relationship: rcBreakdown	An association represents that one to multiple ResourceConservativeBreakdownContext are associated to one ResourceConservativeBreakdown.	
Relationship: rclDefinition	An association represents in each ResourceConservativeBreakdownContext one ResourceConservativeLevelDefinition (i.e.RCL) is defined.	
«block» **ResourceConservationLevelDefinition**	Identification of RCL definitions i.e. lifecycles definitions of a RCP. RCLi: Resource Conservation Level, where i= 0,1,2…RCL0 represents RCP in its 1st, 2nd, 3rd…designed lifecycles	• ResCoM: RCL [DOW] • PLCS: PhysicalElement [1]
Relationship: predefinedPeriods	A composition represents one ResourceConservationLevelDefinition defines one to multiple LifePeriod.	
«block» **LifecyclePeriod**	Predefined length (time or performance) of a RCL, which is a Property of RCL.	• ResCoM: Predefined length [3]
Block: **RCLDefinitionStructure**	Relationship between a ResourceConservativeLevelDefinition and a RCLElement where RCLElement is regarded as a child.	• PLCS: PhysicalElementUsage, Effectivity and EffectivityAssignment [1]

Relationship: lifecycle	An association represents that in one ResourceConservativeLevelDefinition one to multiple RCLDefinitionStructure are defined.	
Relationship: element	An association represents that one RCLDefinitionStructure consists of one to multiple RCLElement.	
«block» **RCLElement**	A basic element of a ResourcConservationLevelDefinition that represents a physical sub part (module or component) of RCP.	• ResCoM: Module [D2.3] • PLCS: PhysicalElement [1]
Relationship: *Generalization* Note: a child block inherits all properties that its parent block has, but may have additional properties that make the child block special.	A specialization where RCLElement is the parent block and ArchitectureNode, and ModuleVariant are the children blocks. It represents that one RCLElement could have two specializations which are ArchitectureNode and ModuleVariant.	
«block» **RCLElementStructure**	Relationship between a parent and child RCLElement.	• PLCS: PhysicalElementUsage, Effectivity and EffectivityAssignment [1]
Relationship: child, parent	Associations between RCLElement and its constituent lower level RCLElement.	
«block» ***DesignItem***	An abstract class. The identification of a design, a generalization of the design objects. It is a collector of data common to all revisions of the *DesignItem*.	• PLCS: Product [1]
Relationship: *Generalization* Note: a child block inherits all properties that its parent block has, but may have additional properties that make the child block special.	A generalization where *DesignItem* is the parent block and ProductDesign, Material, and Substance are the children blocks. It represents that one *DesignItem* could have three specializations which are ProductDesign, Material, and Substance.	
Relationship: document	An association represents that one *DesignItem* is associated to zero to multiple Document.	
«block» **Document**	A Document is used to identify documentation data of ResourceConservativeBreakdown/ *DesignItem*.	• PLCS: Document [1]
«block» **ProductDesign**	Specialization of *DesignItem* that collects the definitional information of the versions of a product design.	• ResCoM: Product/component indexes [DOW] • PLCS: Part [1]
Relationship: material	An aggregation represents one ProductDesign consists of one to multiple Material.	
Relationship: classification	An association represents that one ProductDesign is classified as zero or one valueType: ProductDesignType.	
Relationship: category	An association represents that one ProductDesign is classified as zero or one valueType: Category.	

Relationship: *Generalization* Note: a child block inherits all properties that its parent block has, but may have additional properties that make the child block special.	A generalization where ProductDesign is the parent block and Module, Interface, and StandardInterface are the children blocks. It represents that one ProductDesign could have three specializations which are Module, Interface, and StandardInterface.	
«block» **Material**	Specialization of *DesignItem* that collects the definitional information of the versions of a non-countable material.	• ResCoM: Material indexes [DOW] • PLCS: Part [1]
Relationship: substance	A composition represents one Material consists of zero or multiple Substance.	
Relationship: classification	An association represents that one Material is classified as zero or one valueType: MaterialType.	
«block» **Substance**	Specialization of *DesignItem* that collects the definitional information of the restricted substance in the material.	• ResCoM: Conformity to legislation [D2.3] • PLCS: Part [1]
«block» **DesignStructure**	Relationship between a parent and child *DesignItem*	PLCS: NextAssemblyViewUsage, Effectivity and EffectivityAssignment [1]
Relationship: child, parent	Associations between *DesignItem* and its constituent lower level *DesignItem*.	
Relationship: property	An association represents one DesignStructure has zero to multiple Property.	
«block» **Property**	A Property is the record of an attribute or characteristic that is applicable to something.	• PLCS: Property [2]
Block: **RCLElementToDesign**	Relationship between a RCLElement and a ProductDesign that is realized by the element.	• PLCS: BreakdownElementRealization, Effectivity and EffectivityAssignment [1]
Relationship: rclElement	An association represents one RCLElementToDesign is associated with one RCLElement.	
Relationship: productDesign	An association represents one RCLElementToDesign is associated with one ProductDesign.	
«block» **ModuleSystem**	Specialization of ProductConfiguration that collects the definitional information of a module system.	• ResCoM: [D3.3]
Relationship: architectureNode	An association represents one ModuleSystem is associated with one to multiple ArchitectureNode.	
«block» **ArchitectureNode**	Specialization of RCLElement that collects the definitional information of an architecture of a module system.	
Relationship: module	An association represents one ArchitectureNode is associated with zero to multiple Module.	
«block» **ArchitectureNodeStucture**	Relationship between a parent and child ArchitectureNode.	
Relationship: child, parent	Associations between ArchitectureNode and its constituent lower level ArchitectureNode.	

«block» **Module**	Specialization of ProductDesign that collects the definitional information of the versions of a module design.	• ResCoM: [D3.3]
Relationship: moduleVariant	An association represents one Module is associated with one to multiple ModuleVariant.	
Relationship: interface	An association represents one Module is associated with one to multiple Interface.	
Relationship: productProperty	An association represents one Module is associated with one to multiple ProductProperty.	
«block» **ModuleVariant**	A Module designed with a set of specifications is a variant of the Module. A Module design could be realized by multiple ModuleVariant depending on the specifications.	• ResCoM: [D3.3]
Relationship: goalValue	An association represents one ModuleVariant is associated with one to multiple GoalValue.	
Relationship: productProperty	An association represents one ModuleVariant is associated with one to multiple ProductProperty.	
«block» **GoalValue**	The value i.e. specification of a ProductProperty for defining a ModuleVariant to realize a Module.	• ResCoM: [D3.3]
«block» **ProductProperty**	A ProductProperty is the record of an attribute or characteristic that is applicable to a product.	• ResCoM: [D3.3]
Relationship: goalValue	An association represent one ProductProperty can define one to multiple GaolValue.	
«block» **Interface**	Specialization of ProductDesign that collects the definitional information of the versions of an interface design.	• ResCoM: [D3.3]
Relationship: module1, module2	Associations represent one Interface is associated with two Module.	
Relationship: standardInterface	An association represent one Interface is consist of one to multiple StandardInterface.	
«block» **StandardInterface**	Specialization of ProductDesign that collects the definitional information of the versions of a standard interface design.	
Relationship: interfaceType	An association represent one StandardInterface is classified to one InterfaceType.	
Relationship: moduleVariant	An association represent one StandardInterface is applied to one to multiple ModuleVariant.	
«block» **InterfaceType**	A classification of the StandardInterface.	

Property	**Description**	**Comment**
«Value Type»		
MaterialType : MaterialInPart	A category defined by the XML schema of Granta BoM Analyzer for structuring the BoM of a product.	Integration with Granta BoM Analyzer
ProductDesignType : RootPart		
ProductDesignType : PartContainer		
ProductDesignType : InHousePart		
Category : Category1	parts to last	
Category : Category2	wear and tear part	

Category : Category3	refreshment part	A category defined by Bugaboo for categorizing the parts design.
Category : Category4	packaging or lost	
parts		
ids : Identifier [1..*]	A set of Identifiers for the ResourceConservativeBreakdown/ ResourceConsercationLevelDefinition/ RCLElement/ProductConfiguration/ *DesignItem*/DesignStructure/Property/ ProductProperty/InterfaceType.	An identifier provides the identifying code for the product data.
references		
org : Organization [1]	An Organization is the ResourceConservativeBreakdown belonged to.	PLCS: Organization [1] A hidden block "Organization" which is shown in ReferenceData.
doc : Document [0..*]	A Document is used to identify documentation data of ResourceConservativeBreakdown/ *DesignItem*.	PLCS: Document [1]
values		
version : String [1]	A version identifies a version of a ResourceConservativeBreakdown i.e. a RCP.	A version serves as the collector of the data characterizing a realizable object in various application contexts.
version : String [1..*]	A version identifies a version of ResourceConsercationLevelDefinition/ RCLElement/*DesignItem*.	
name : String [0..1]	A name provides the identifying name in terms of String for the ResourceConservativeBreakdown/ ResourceConsercationLevelDefinition/ RCLElement/ProductConfiguration/ *DesignItem*/Property/ProductProperty/InterfaceType	
description : String [0..1]	A set of text based descriptions of the ResourceConservativeBreakdown/ ResourceConsercationLevelDefinition/ RCLElement/ProductConfiguration/ *DesignItem*/InterfaceType.	
startDate : date_time [1] or [0..1]	A date is assigned to a relationship/structure between two objects showing when this relationship/structure begins.	Relationship/structure between: • ResourceConservativeBreakdown and ResourceConservationLevelDefinition • ResourceConservationLevelDefinition and RCLElement • ResourceConservatioLevelDefinition and LifecyclePeriod • RCLElement and its sub parts • RCLElement and ProductDesign • *DesignItem* and its sub parts • Designstructure and Property • ArchitectureNode and its sub parts • Module and ProductProperty
endDate : date_time [0..1]	A date is assigned to a relationship/structure between two objects showing when this relationship/structure ends up.	

optimumNumberOfLifecycles : integer [1..*]	The optimum number of lifecycles of defined for a ResourceConservativeBreakdown (RCP).	ResCoM concept [DOW]
unit : String [1]	A LifecyclePeriod/ProductProperty/ Property is specified by a unit.	
value : Real [1]	A LifecyclePeriod/ProductProperty/ Property/GoalValue is specified by a value.	
quantity : Real [1]	The quantity of a child *DesignItem* constitutes its parent *DesignItem*.	
useCycle : Integer [0..1]	A useCycle of a ProductDesign starts when it is released for use to a certain use. It ends when the user of a product changes.	Integration with Granta Enhanced eco-audit
classification : String [1] or [0..1]	A classification is a relationship to the ArchitectureNode/ProductDesign/ Material that a subject is a member of.	

2.3 ResCoM_InLife

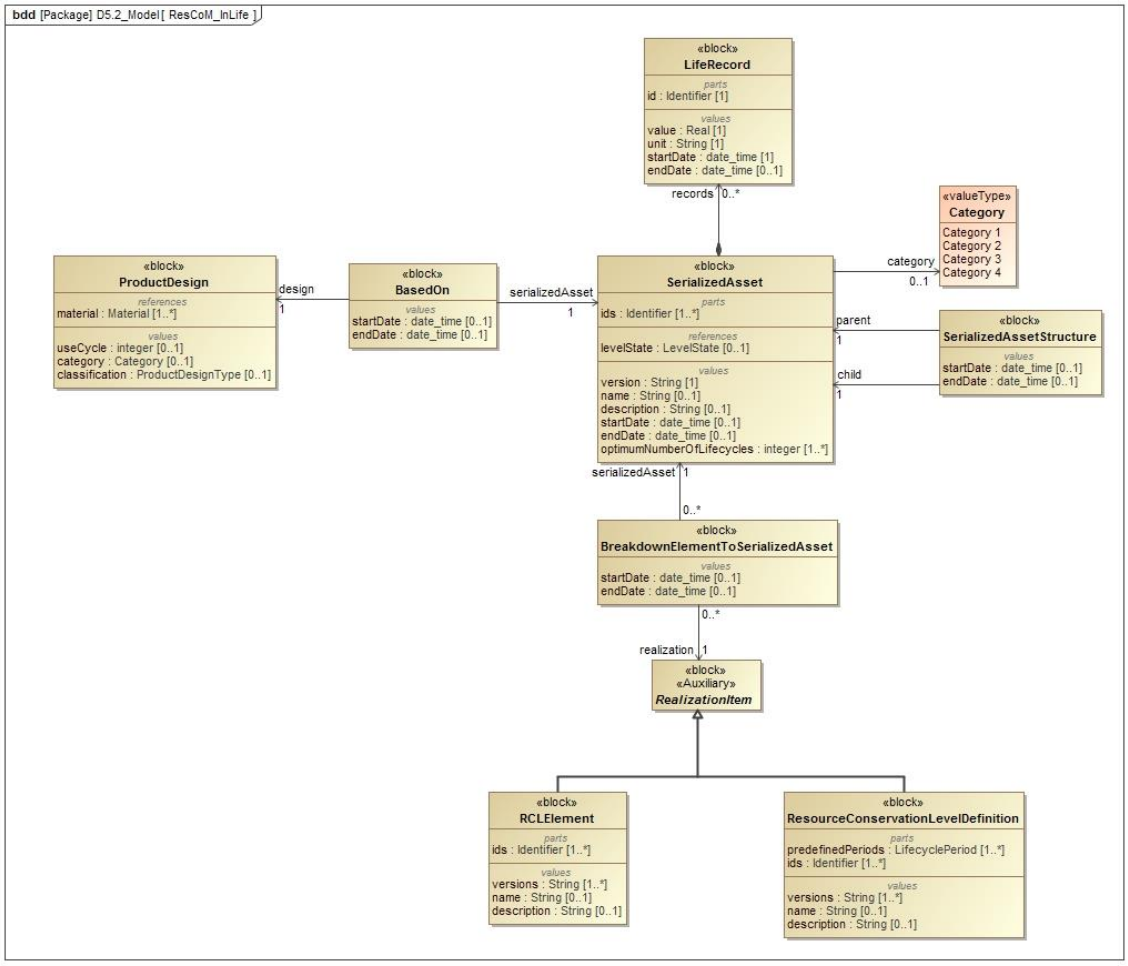


Figure 2.3 ResCoM_InLife model as a SysML block diagram

The ResCoM_InLife model in figure 2.3 aims to capture and store the information regarding the definitions of the product individuals in the manufacturing, in-service and remanufacturing phases. Those information are supposed to be stored and structured in some ERP tools and then imported to the collaborative software platform or directly be created in the collaborative software platform.
As a result one can look up and trace the following information of a product individual (**SerializedAsset** in the data model) including its subcomponents through its structure (**SerializedAssetStructure** in the data model) throughout its multiple lifecycles:

- life record of each lifecycle (**LifeRecord** in the data model),
- the product design (**ProductDesign** in the data model) that it is based on (**BasedOn** in the data model) for manufacturing and remanufacturing in each lifecycle,
- the lifecycle definition (**ResourceConservativeLevelDefinition** in the data model) that it is based on (**RCLElementToSerializedAsset** in the data model) in each lifecycle,
- predefined product structure (**RCLDefinitionStructure** and **RCLElement** in the data model) in each lifecycle.

The concepts, relationships and properties of ResCoM_InLife model are presented in Table 3**Table 1**.

Table 1. The concepts, relationships and properties of the ResCoM_InLife model.

Block / Relationship	Description	Concept reference
«block» **SerializedAsset**	A SerializedAsset identifies an individual product that has been manufactured. An identifier provides the identifying code for the product individual data.	• PLCS: ProductAsRealized [1]
Relationship: records	A composition represents one SerializedAsset defines zero to multiple LifeRecord.	
Relationship: category	An association represents that one **SerializedAsset** is classified as zero or one valueType: Category.	
«block» **LifeRecord**	The identification of a life record that records each RCL a SerializedAsset has gone through and the usage state representing by value within this RCL.	• ResCoM: traceability of each RCL [DOW] • PLCS: Effectivity and EffectivityAssignment [1]
«block» **SerializedAssetStructure**	Relationship between a parent and child SerializedAsset.	• PLCS: RealizedAssemblyRelation-ship, Effectivity and EffectivityAssignment [1]
Relationship: child, parent	Associations between SerializedAsset and its constituent lower level SerializedAsset.	
«block» **BasedOn**	A BasedOn is a relationship between a ProductDesign, and the product that has been made from the design, represented by SerializedAsset.	• PLCS: ProductDesignVersionTo-Individual, Effectivity and EffectivityAssignment [1]
Relationship: design	An association represents that one BasedOn basese on one ProductDesign.	
Relationship: serializedAsset	An association represents that one BasedOn associating with one SerializedAsset.	
«block» **ProductDesign**	Specialization of *DesignItem* that collects the definitional information of the versions of a product design.	• ResCoM: Product/component indexes [DOW] • PLCS: Part [1]

«block» **BreakdownElementToSerializedAsset**	Relationship between a SerializedAsset and its predefined structure in each RCL.	PLCS: BreakdownElementRealization, Effectivity and EffectivityAssignment [1]
Relationship: serializedAsset	An association represents that zero to multiple BreakdownElementToSerializedAsset associating with one SerializedAsset.	
Relationship: realization	An association represents that zero to multiple BreakdownElementToSerializedAsset associating with one SerializedAsset.	
«block» «Auxiliary» ***RealizationItem***	A *RealizationItem* is an abstract generalization of instances.	
Relationship: *Generalization* Note: a child block inherits all properties that its parent block has, but may have additional properties that make the child block special.	A generalization where *RealizationItem* is the parent block and RCLElement and ResourceConservationLevelDefinition are the children blocks. It represents that one *RealizationItem* could have three specializations which are RCLElement, and ResourceConservationLevelDefinition.	
«block» **RCLElement**	A basic element of a ResourcConservationLevelDefinition that represents a physical sub part (module or component) of RCP.	• ResCoM: Module [D2.3] • PLCS: PhysicalElement [1]
«block» **ResourceConservationLevelDefinition**	Identification of RCL definitions i.e. lifecycles definitions of a RCP. RCLi: Resource Conservation Level, where i= 0,1,2…RCL0 represents RCP in its 1st, 2nd, 3rd…designed lifecycles	• ResCoM: RCL [DOW] • PLCS: PhysicalElement [1]

Property	Description	Comment
«ValueType»		
Category : Category1	parts to last	A category defined by Bugaboo for categorizing the parts design.
Category : Category2	wear and tear part	
Category : Category3	refreshment part	
Category : Category4	packaging or lost	
parts		
ids : Identifier [1..*]	A set of Identifiers for the SerializedAsset/LifeRecord/ ResourceConservationLevelDefinition/RCL Element.	An identifier provides the identifying code for the product data.
predefinedPeriods: LifecyclePeriod [1..*]	Predefined length (time or performance) of a ResourceConservationLevelDefinition (RCL), which is a property of RCL.	The hidden block "LifecyclePeriod" which is shown in figure 2.2 ResCoM_ProductRCBreakdown_and_Design model.
references		
levelState : LevelState [0..1]	A LevelState is a definition of a possible or typical state levels.	The hidden block "LevelState" which is shown in figure 2.4 ResCoM_Activity_and_Methods model.

material :Material[1..*]	Specialization of *DesignItem* that collects the definitional information of the versions of a non-countable material.	The hidden block "Material" which is shown in figure 2.2 ResCoM_ProductRCBreakdown_and_Design model.
values		
version : String [1]	A version identifies a version of a SerializedAsset i.e. a manufactured RCP.	A version serves as the collector of the data characterizing a realizable object in various application contexts.
version : String [1..*]	A version identifies a version of ResourceConsercationLevelDefinition/RCLElement.	
name : String [0..1]	A name provides the identifying name in terms of String for the SerializedAsset/ ResourceConservationLevelDefinition/RCLElement.	
description : String [0..1]	A set of text based descriptions of the SerializedAsset/ ResourceConservationLevelDefinition/RCLElement.	
startDate : date_time [1] or [0..1]	A date is assigned to a relationship/structure between two objects showing when this relationship/structure begins.	Relationship/structure between: • ProductDesign and SerializedAsset • SerializedAsset and its sub parts • SerializedAsset and its property LifeRecord • SerializedAsset and ResourceConservationLeveIDefinition/RCLElement.
endDate : date_time [0..1]	A date is assigned to a relationship/structure between two objects showing when this relationship/structure end up.	
value : Real [1]	A LifeRecord is specified by a value.	
unit : String [1]	A LifecycleRecord is specified by a unit.	
optimumNumberOfLifecycles : integer [1..*]	The optimum number of lifecycles of defined for a SerializedAsset i.e. a manufactured RCP.	ResCoM concept [DOW]
useCycle : integer [0..1]	A useCycle of a ProductDesign starts when it is released for use to a certain use. It ends when the user of a product changes.	Integration with Granta Enhanced eco-audit
category : Category [0..1]	A category defined by Bugaboo for categorizing a ProductDesign.	The hidden valueType "Category" which is shown in figure 2.2 ResCoM_ProductRCBreakdown_and_Design model.
classification : ProductDesignType [0..1]	A classification is a relationship to the ProductDesign that a subject is a member of.	The hidden valueType "ProductDesignType" which is shown in figure 2.2 ResCoM_ProductRCBreakdown_and_Design model.

2.4 ResCoM_Activities_and_Methods

The ResCoM_InLife model in figure 2.4 aims to capture and store the information regarding the activities for closing the loop of the product system and supporting multiple lifecycles of the RCP. Those information are supposed to be stored and structured in some ERP tools and then imported to the collaborative software platform or directly be created in the collaborative software platform.
As a result one can look up and tract the following information of an individual product (**SerializedAsset** in the data model) throughout its multiple lifecycles:

- current state level (**LevelState** in the data model) of the individual product,
- the activities (**PlannedActivity** and **ActivityActual** in the data model) which are required for closing the product system,
- the methods (**TaskMethod** in the data model) and life plans (**LifecyclePlan** in the data model) of the activities,
- the lifecycle definition (**LifecyclePlanAssignment** and **ResourceConservationLevelDefinition** in the data model) that the life plan is applied to,
- the objects (**AppliedTaskMethodAssignment, ResourceConservationLevelDefinition, rclDefinitionStructure, RCLElement, ResourceConservativeBreakdown** and ***DesignItem*** in the data model) that the methods are applied to.

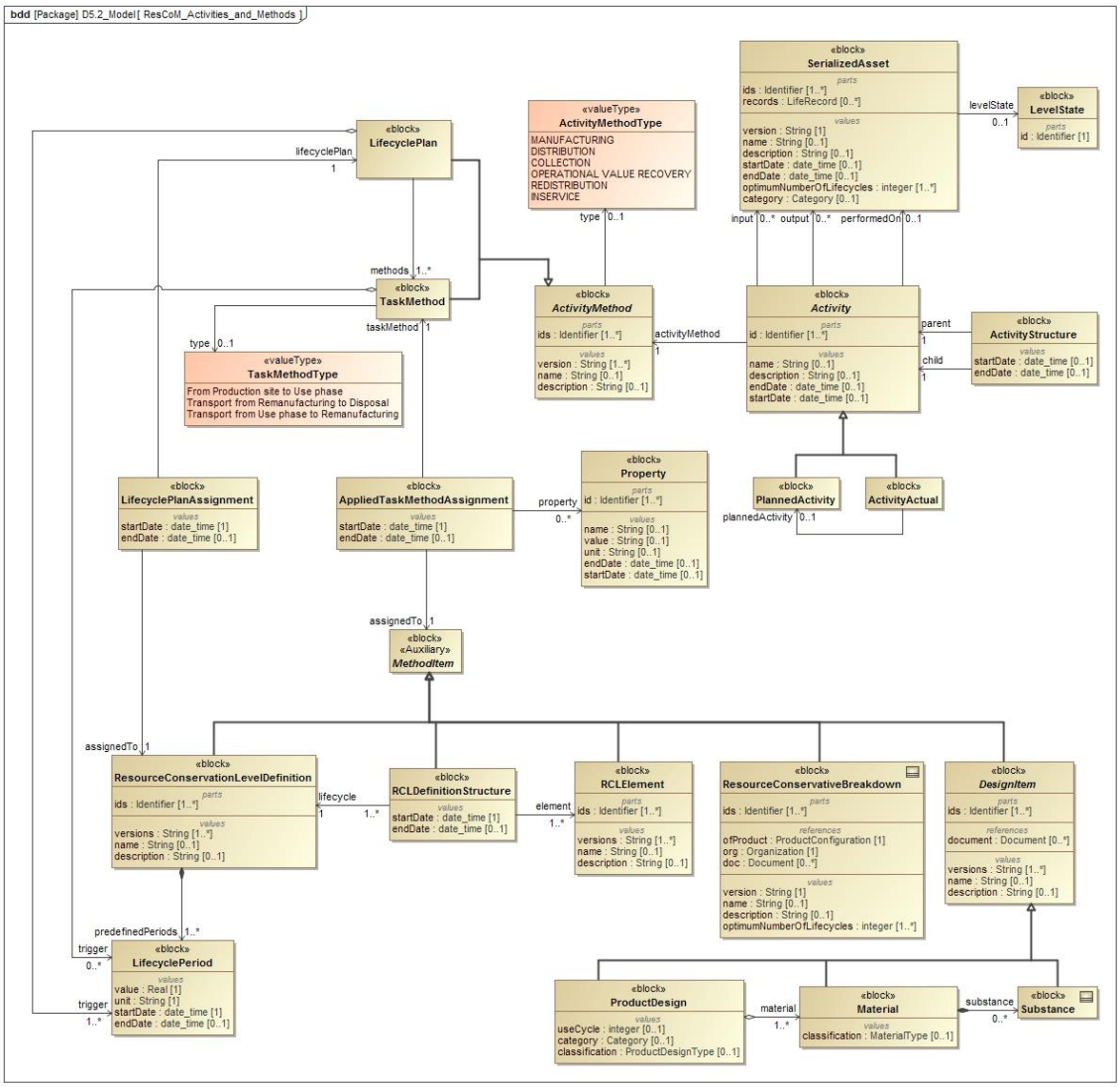


Figure 2.4 ResCoM_Activities_and_Methods model a SysML block diagram

The concepts, relationships and properties of ResCoM_Activities_and_Methods model are presented in Table 4.

Table 2. The concepts, relationships and properties of the ResCoM_Activities_and_Methods model

Block / Relationship	Description	Concept reference
«block» **SerializedAsset**	A SerializedAsset identifies an individual product that has been manufactured. An identifier provides the identifying code for the product individual data.	• PLCS: ProductAsRealized [1]
Relationship: levelState	An association represents that one SerializedAsset is associated with zero or one LevelState.	
«block» **LevelState**	A LevelState is a definition of a possible or typical state levels.	• Bugaboo defines 4 state levels for the quality of the parts returned back for remanufacturing. • PLCS: Activity [1]
«block» ***Activity***	An abstract class. An *Activity* is an action or a set of actions that consume time and resources and whose performance is necessary to achieve, or contribute to, the realization of one or more outcomes.	• ResCoM: Production activities, supply chain activities [D2.3] • PLCS: Activity [1]
Relationship: outputs	An association represents that zero to multiple SerializedAsset is the input to one *Activity*.	
Relationship: inputs	An association represents that zero to multiple SerializedAsset is the output to one *Activity*.	
Relationship: performedOn	An association represents that one *Activity* is performed on zero or one SerializedAsset.	
Relationship: activityMethod	An aggregation represents one *Activity* utilizes one ActivityMethod i.e. the specific way to carry out the *Activity*.	
Relationship: *Generalization* Note: a child block inherits all properties that its parent block has, but may have additional properties that make the child block special.	A generalization where *Activity* is the parent block and PlannedActivity and ActivityActual are the children blocks. It represents that one *Activity* could have two specializations which are PlannedActivity and ActualActivity.	
«block» **ActivityStructure**	Relationship between a parent and child *Activity*.	• PLCS: ActivityRelationship [1]
Relationship: child, parent	Associations between *Activity* and its constituent lower level *Activity*.	
«block» **PlannedActivity**	A PlannedActivity is an *Activity* which, when first defined, has yet to start and so is being planned. It is a record of the intent to perform an *Activity*.	• PLCS: ActivityPlanned [1]
«block» **ActivityActual**	A PerformedActivity is an *Activity* that has started but not necessarily finished.	• PLCS: ActivityActual [1]
Relationship: plannedActivity	An association represents that one ActivityActual is directed by zero or one PlannedActivity.	
«block» ***ActivityMethod***	An *ActivityMethod* is a specific way to carry out an *Activity*.	• PLCS: ActivityMethod [1]

Relationship: type	An association represents that one *ActivityMethod* defines one valueType.	
Relationship: *Generalization* Note: a child block inherits all properties that its parent block has, but may have additional properties that make the child block special.	A generalization where *ActivityMethod* is the parent block and LifecyclePlan and TaksMethod are the children blocks. It represents that one *ActivityMethod* could have two specializations which are LifecyclePlan and TaksMethod.	
«block» **LifecyclePlan**	A LifecyclePlan is a specialization of ActivityMethod predefines a set of activity methods for supporting a RCP (design) from first manufacturing to operational value recovery in each end of its lifecycle.	• ResCoM: Right time, Right quality, Right Quantity [D2.3] • PLCS: Scheme [1]
Relationship: methods	An association represents that one LifecyclePlan is associated with one to multiple Method.	
Relationship: trigger	An aggregation represents that one LifecyclePlan is triggered by one to multiple LifecyclePeriod.	
«block» **LifecyclePlanAssignment**	A LifecyclePlanAssignment is a relationship between a LifecyclePlan and ResourceConservationLevelDefinition that the LifecyclePlan is applied to.	• PLCS: SchemeVersionAssignment, Effectivity and EffectivityAssignment [1]
Relationship: lifecycleplan	An association represents that one LifecyclePlanAssignment utilizes one LifecyclePlan.	
Relationship: assignedTo	An association represents that one LifecyclePlanAssignment is assigned to one ResourceConservationLevelDefinition.	
«block» **TaskMethod**	A TaskMethod is a specialization of *ActivityMethod*. It is a specification of work.	• PLCS: TaskMethod [1]
Relationship: type	An association represents that one TaksMethod defines one valueType.	
Relationship: trigger	An aggregation represents that one TaksMethod is triggered by zero to multiple LifecyclePeriod.	
«block» **AppliedTaskMethodAssignment**	An AppliedTaskMethodAssignment is a relationship between a TaskMethod and *MethodItem* that the AppliedTaskMethodAssignment is applied to.	• PLCS: TaskMethodVersionAssignment, Effectivity and EffectivityAssignment [1]
Relationship: taskMethod	An association represents that one AppliedTaskMethodAssignment ultilizes one TaskMethod.	
Relationship: assignedTo	An association represents that one AppliedTaskMethodAssignment is assigned to one *MethodItem*.	
Relationship: properties	An association between AppliedTakMethodAssignment and Property represents that one AppliedTakMethodAssignment has zero to multiple Property.	
«block» **Property**	A Property is the record of an attribute or characteristic that is applicable to something.	• PLCS: Property [1]

«block» «Auxiliary» ***MethodItem***	A *MethodItem* is an abstract generalization of instances.	• PLCS: *TaksAssignmentSelect* [1]
Relationship: *Generalization* Note: a child block inherits all properties that its parent block has, but may have additional properties that make the child block special.	A generalization where *MethodItem* is the parent block and ResourceConservationLevelDefinition, RCLDefinitionStructure, RCLElement, ResourceConservativeBreakdown and *DesignItem* are the children blocks. It represents that one *MethodItem* could have five specializations which are ResourceConservationLevelDefinition, RCLDefinitionStructure, RCLElement, ResourceConservativeBreakdown and *DesignItem.*	
«block» **ResourceConservationLevelDefinition**	Identification of RCL definitions i.e. lifecycles definitions of a RCP. RCLi: Resource Conservation Level, where i= 0,1,2…RCL0 represents RCP in its 1st, 2nd, 3rd…designed lifecycles	• ResCoM: RCL [DOW] • PLCS: PhysicalElement [1]
Relationship: predefinedPeriods	A composition represents one ResourceConservationLevelDefinition defines one to multiple LifePeriod.	
«block» **LifecyclePeriod**	Predefined length (time or performance) of a ResourceConservativeLevelDefinition i.e. RCL, which is a Property of RCL.	• ResCoM: Predefined length [3]
«block» **RCLDefinitionStructure**	Relationship between a ResourceConservativeLevelDefinition and a RCLElement where RCLElement is regarded as a child.	• PLCS: PhysicalElementUsage, Effectivity and EffectivityAssignment [1]
Relationship: lifecycle	An association represents that in one ResourceConservativeLevelDefinition one to multiple RCLDefinitionStructure are defined.	
Relationship: element	An association represents that one RCLDefinitionStructure consists of one to multiple RCLElement.	
«block» **RCLElement**	A basic element of a ResourcConservationLevelDefinition that represents a physical sub part (module or component) of RCP.	• ResCoM: Module [D2.3] • PLCS: PhysicalElement [1]
«block» **ResourceConservativeBreakdown**	A ResourceConservativeBreakdown provides a mean to subdivide a RCP (Resource Conservative Product) into a set of related RCLs (Resource Conservative Levels) to which additional information can be attached. This usually takes the form of a tree.	• ResCoM: RCP [DOW] • PLCS: Breakdown [1]
«block» ***DesignItem***	The identification of a design, a generalization of the design objects. It is a collector of data common to all revisions of the *DesignItem*.	• PLCS: Product [1]
Relationship: *Generalization* Note: a child block inherits all properties that its parent block has, but may have additional properties that make the child block special.	A generalization where *DesignItem* is the parent block and ProductDesign, Material, and Substance are the children blocks. It represents that one *DesignItem* could have three specializations which are ProductDesign, Material, and Substance.	

«block» **ProductDesign**	Specialization of *DesignItem* that collects the definitional information of the versions of a product.	• ResCoM: Product/component indexes [DOW] • PLCS: Part [1]
«block» **Material**	Specialization of *DesignItem* that collects the definitional information of the versions of a non-countable material.	• ResCoM: Material indexes [DOW] • PLCS: Part [1]
«block» **Substance**	Specialization of *DesignItem* that collects the definitional information of the restricted substance in the material.	• ResCoM: Conformity to legislation [D2.3] • PLCS: Part [1]
Relationship: material	An aggregation represents one ProductDesign consists of one to multiple Material.	
Relationship: substance	A composition represents one Material consists of zero or multiple Substance.	
Property	**Description**	**Comment**
«Value Type»		
ActivityMethodType : MANUFACTURING	RCL_0 production	ResCoM concept [D2.3, D2.4, D3.4]
ActivityMethodType : DISTRIBUTION	Distribution of RCP in its first lifecycle	
ActivityMethodType : COLLECTION	Collection of RCP in each of its end of lifecycle	
ActivityMethodType : OPERATIONAL VALUE RECOVERY	RCL_i Production	
ActivityMethodType : REDISTRIBUTION	Distribution of RCP in its 2^{nd}, 3^{rd}… lifecycle	
ActivityMethodType : IN SERVICE	Maintenance during each lifecycle	
TaskMethodType : From Production site to Use phase	A category defined by the ResCoM report of Granta for defining the transport type of a product.	Integration with Granta Enhanced eco-audit
TaskMethodType : Transport from Remanufacturing to Disposal		
TaskMethodType : Transport from Use phase to Remanufacturing		
parts		
ids : Identifier [1..*]	A set of Identifiers for the SerializedAsset/ *Activity*/*ActivityMethod*/Property /ResourceConservationLevelDefinition/RCL Element/ ResourceConservativeBreakdown/ *DesignItem*.	An identifier provides the identifying code for the product data.
records : LifeRecord [0..*]	The identification of a life record that records each RCL a SerializedAsset has gone through and the usage state representing by value within this this RCL.	The hidden block "LifeRecord" which is shown in figure 2.3 ResCoM_InLife model.
references		

levelState : LevelState [0..1]		The hidden block "LevelState" which is shown in figure 2.4 ResCoM_Activity_and_Methods model.
material :Material[1..*]	Specialization of *DesignItem* that collects the definitional information of the versions of a non-countable material.	The hidden block "Material" which is shown in figure 2.2 ResCoM_ProductRCBreakdown_and_Design model.
ofProduct : Product Configuration [1]	An association represents that one ResourceConservativeBreakdown is a breakdown of one ProductConfiguration (product concept).	The hidden block "ProductConfiguration" which is shown in figure 2.2 ResCoM_ProductRCBreakdown_and_Design model.
org : Organization [1]	An Organization is the ResourceConservativeBreakdown belonged to.	PLCS: Organization [1] A hidden block "Organization" which is shown in ReferenceData.
doc : Document [0..*]	A Document is used to identify documentation data of ResourceConservativeBreakdown/ *DesignItem*.	PLCS: Document [1] The hidden block "Document" which is shown in figure 2.2 ResCoM_ProductRCBreakdown_and_Design model.
values		
version : String [1]	A version identifies a version of a SerializedAsset/ ResourceConservativeBreakdown. A version serves as the collector of the data characterizing a realizable object in various application contexts.	A version serves as the collector of the data characterizing a realizable object in various application contexts.
version : String [1..*]	A version identifies a version of *ActivityMethod*/ ResourceConservationLevelDefinition/ RCLElement/*DesignItem*.	
name : String [0..1]	A name provides the identifying name in terms of String for the SerializedAsset/ *Activity*/*ActivityMethod*/Property/ ResourceConservationLevelDefinition/ RCLElement/*DesignItem*.	
description : String [0..1]	A set of text based descriptions of the SerializedAsset/*Activity*/*ActivityMethod*/ LifecyclePlan/Method/ ResourceConservationLevelDefinition/ RCLElement / ResourceConservativeBreakdown/ *DesignItem*.	
startDate : date_time [1] or [0..1]	A date is assigned to a relationship/structure between two objects showing when this relationship/structure begins or a date is assigned to start an activity.	Relationship/structure between: • Start date of the first life and end date of the last life of a SerializedAsset

endDate : date_time [0..1]	A date is assigned to a relationship/structure between two objects showing when this relationship/structure end up, or a date is assigned to end up an activity.	• *Activity* and its sub activities • Start and end dates of an *Activity* • LifecyclePlan and ResourceConservation-LevelDefinition • TaskMethod and *MethoItem* • TaskMethod and Property • RCLElement and ResourceConservation-LevelDefinition • ResourceConservation-LevelDefinition and LifecyclePeriod
optimumNumberOfLifecycles : integer [1..*]	The optimum number of lifecycles of defined for a SerializedAsset.	ResCoM concept [DOW]
category : Category [0..1]	A category defined by Bugaboo for categorizing a product design.	The hidden valueType “Category” which is shown in figure 2.3 ResCoM_InLife model.
value : Real [1] unit : String [1]	A Property/LifecyclePeriod is specified by a value and a unit.	
useCycle : integer [0..1]	A useCycle of a ProductDesign starts when it is released for use to a certain use. It ends when the user of a product changes.	Integration with Granta Enhanced eco-audit
category : Category [0..1]	A category defined by Bugaboo for categorizing a ProductDesign.	The hidden valueType “Category” which is shown in figure 2.2 ResCoM_ProductRCBreakdown_and_Design model.
classification : ProductDesignType [0..1]	A classification is a relationship to the ProductDesign that a subject is a member of.	The hidden valueType “ProductDesignType” which is shown in figure 2.2 ResCoM_ProductRCBreakdown_and_Design model.
classification : MaterialType [0..1]	A classification is a relationship to the Material that a subject is a member of.	The hidden valueType “MaterialType” which is shown in figure 2.2 ResCoM_ProductRCBreakdown_and_Design model.

3. Industrial pilot validation

The ResCoM domain information model described in chapter 2 was verified in the ResCoM collaborative software platform by the 4 industrial pilots: Gorenje washing machine Asko, Loewe TV Bild 5, Tedrive Pinion Steering Gear and Bugaboo stroller Cameleon 3.

3.1 Overview of the ResCoM collaborative software platform[1]

The graphical user interface of the ResCoM collaborative software platform consists of three main parts as shown in figure 3.1: industrial case, product multiple lifecycle management and ResCoM tools applied to the industrial case.

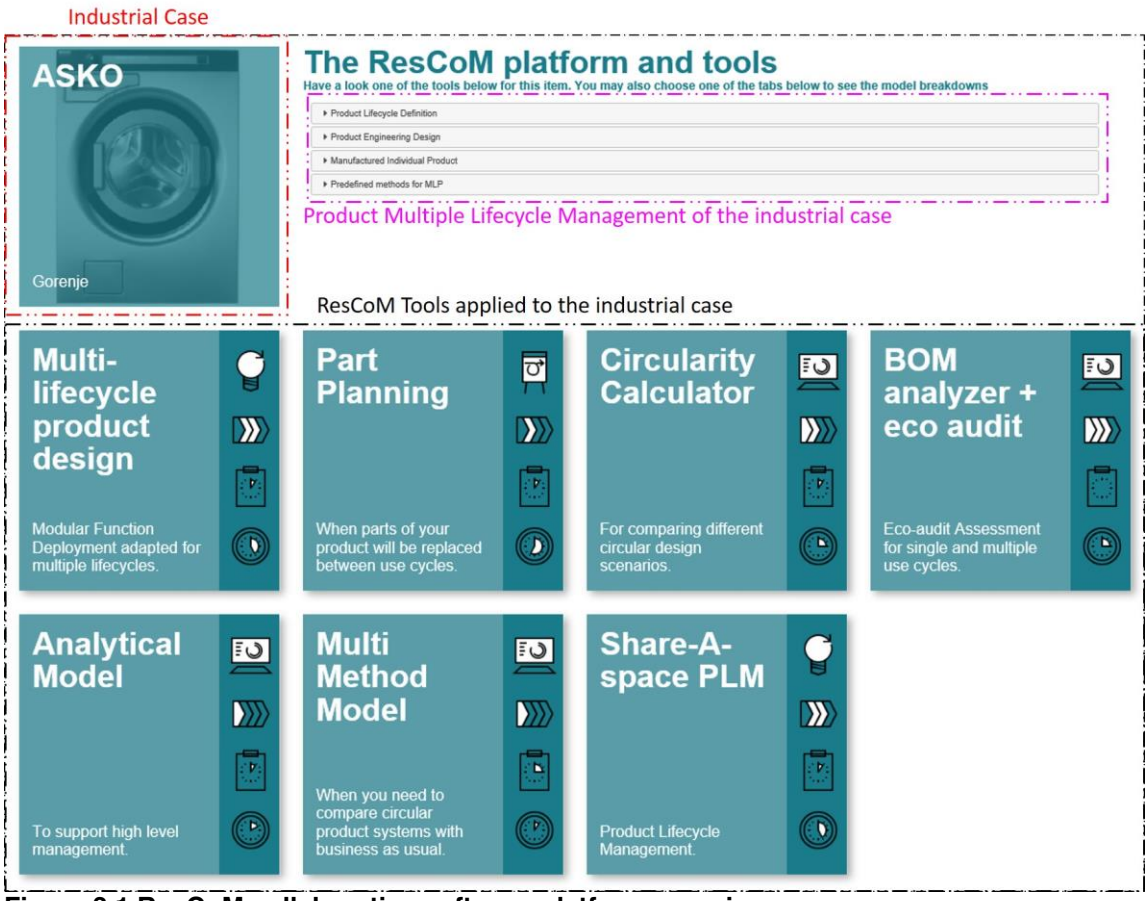


Figure 3.1 ResCoM collaborative software platform overview

3.2 Product multiple lifecycle management

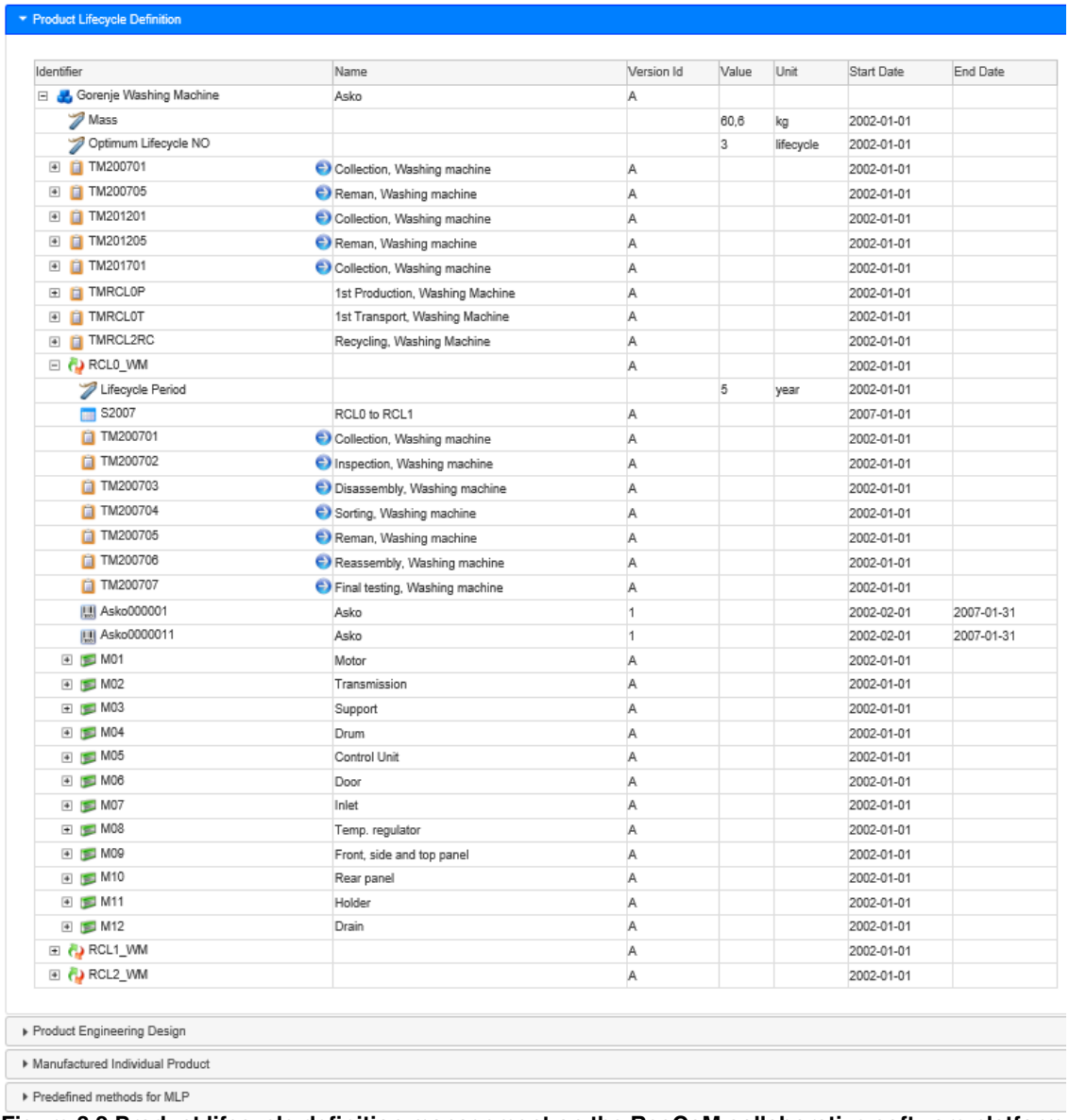
The industrial case provides a BoM, together with the multiple lifecycles information/data for closed-loop production defined by business model, product design and closed loop supply chain in WP2, 3 and 4. The information/data is well structured based on the final standard data model described in chapter 2 and then imported in the ResCoM collaborative software platform.
Four trees with all the well-structured data managed by the platform are shown in the “Product Multiple Lifecycle Management of the industrial case” of figure 3.1. Figures 3.2, 3.3, 3.4 and 3.5 are screenshots of the four trees: product lifecycle definition, product engineering design, manufactured individual product and predefined methods for multiple lifecycle in the ResCoM collaborative software platform.

3.2.1 Product Lifecycle Definition (figure 3.2)

This is mainly based on the ResCoM_ProductRCBreakdown_and_Design model, but also referenced to ResCoM_InLife model and ResCoM_Activities_and_Methods model. In this tree,

[1] The ResCoM collaborative software platform can be accessed through http://rescomd58.ShareAspace.com/ShareAspace/Session/LogOn

one can find the following information regarding the product lifecycle definition defined in the data model:



Identifier	Name	Version Id	Value	Unit	Start Date	End Date
Gorenje Washing Machine	Asko	A				
Mass			60,6	kg	2002-01-01	
Optimum Lifecycle NO			3	lifecycle	2002-01-01	
TM200701	Collection, Washing machine	A			2002-01-01	
TM200705	Reman, Washing machine	A			2002-01-01	
TM201201	Collection, Washing machine	A			2002-01-01	
TM201205	Reman, Washing machine	A			2002-01-01	
TM201701	Collection, Washing machine	A			2002-01-01	
TMRCL0P	1st Production, Washing Machine	A			2002-01-01	
TMRCL0T	1st Transport, Washing Machine	A			2002-01-01	
TMRCL2RC	Recycling, Washing Machine	A			2002-01-01	
RCL0_WM		A			2002-01-01	
Lifecycle Period			5	year	2002-01-01	
S2007	RCL0 to RCL1	A			2007-01-01	
TM200701	Collection, Washing machine	A			2002-01-01	
TM200702	Inspection, Washing machine	A			2002-01-01	
TM200703	Disassembly, Washing machine	A			2002-01-01	
TM200704	Sorting, Washing machine	A			2002-01-01	
TM200705	Reman, Washing machine	A			2002-01-01	
TM200706	Reassembly, Washing machine	A			2002-01-01	
TM200707	Final testing, Washing machine	A			2002-01-01	
Asko000001	Asko	1			2002-02-01	2007-01-31
Asko0000011	Asko	1			2002-02-01	2007-01-31
M01	Motor	A			2002-01-01	
M02	Transmission	A			2002-01-01	
M03	Support	A			2002-01-01	
M04	Drum	A			2002-01-01	
M05	Control Unit	A			2002-01-01	
M06	Door	A			2002-01-01	
M07	Inlet	A			2002-01-01	
M08	Temp. regulator	A			2002-01-01	
M09	Front, side and top panel	A			2002-01-01	
M10	Rear panel	A			2002-01-01	
M11	Holder	A			2002-01-01	
M12	Drain	A			2002-01-01	
RCL1_WM		A			2002-01-01	
RCL2_WM		A			2002-01-01	

Figure 3.2 Product lifecycle definition management on the ResCoM collaborative software platform

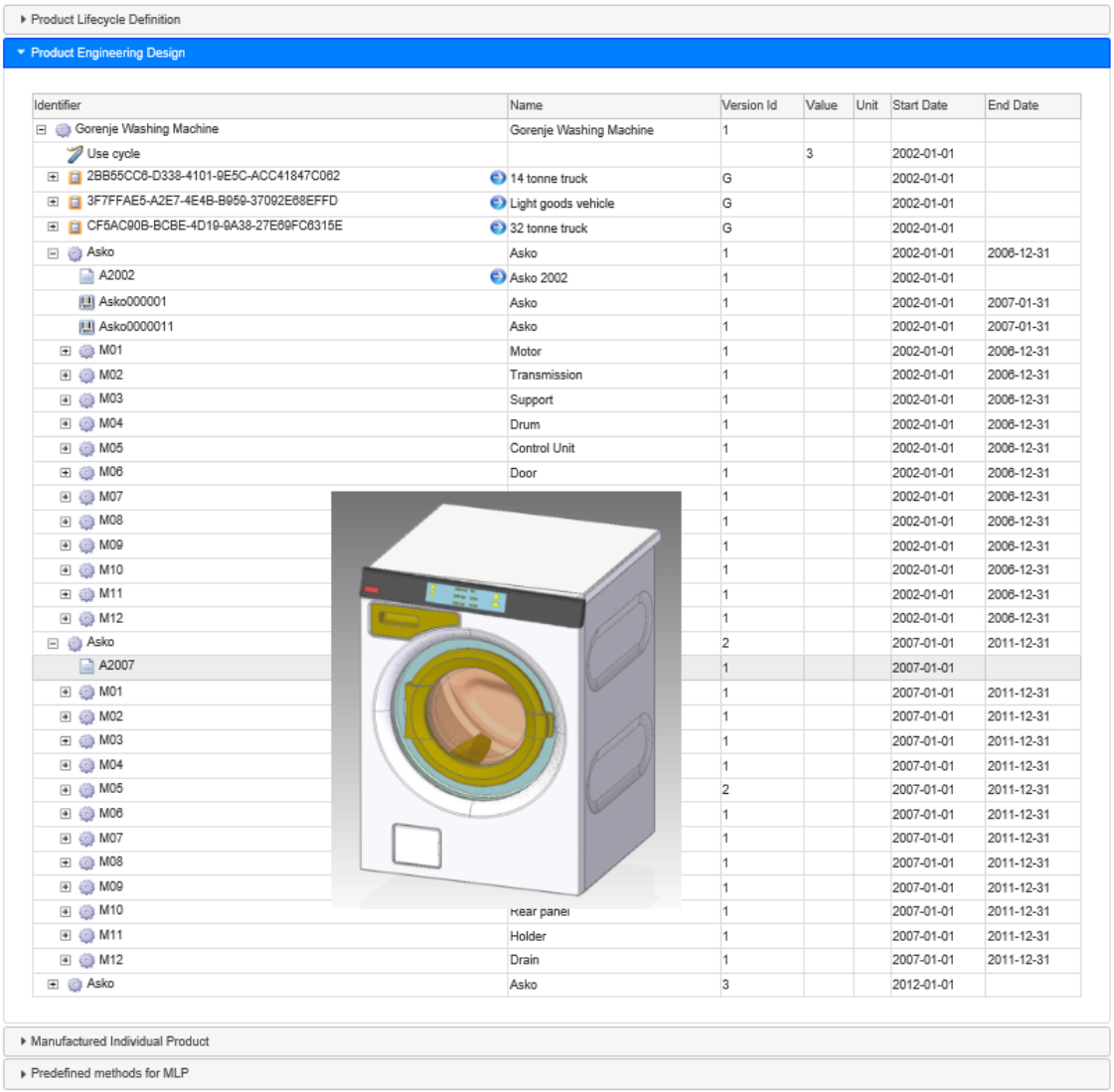
- ResourceConservativeBreakdown[2]
  - Id, version, name, description, and productOptimumLifecycle NO
- ResourceConservativeLevelDefinition
  - Id, version, name, description
  - startDate and endDate of ResourceConservativeBreakdownContext
  - LifecyclePeriod
    - value, unit, startDate and endDate
  - RCLDefinitionStructure
    - startDate and endDate
- RCLElement
  - Id, version, name, description

[2] The objects in the data model will be represented as icons, please see the icon list in the appendix.

    - RCLElementStructure
        - startDate and endDate
- LifecyclePlan (reference to ResCoM_Activities_and_Methods model)
    - Id, version, name, description
    - startDate and endDate of LifecyclePlanAssignment
- Task methods (reference to ResCoM_Activities_and_Methods model)
    - Id, version, name, description
    - startDate and endDate of AppliedTaskMethodAssignment
    - Property
        - Id, name, value, unit, startDate and endDate
- SerializedAsset (reference to ResCoM_InLife model)
    - Id, version, name, description
    - startDate and endDate of BreakdownElementToSerializedAsset

3.2.2 Product Engineering Design (figure 3.3)

This is mainly based on the ResCoM_ProductRCBreakdown_and_Design model, but also referenced to ResCoM_InLife model and ResCoM_Activities_and_Methods model. In this tree, one can find the following information regarding the product engineering design:



Identifier	Name	Version Id	Value	Unit	Start Date	End Date
Gorenje Washing Machine	Gorenje Washing Machine	1				
Use cycle			3		2002-01-01	
2BB55CC8-D338-4101-9E5C-ACC41847C082	14 tonne truck	G			2002-01-01	
3F7FFAE5-A2E7-4E4B-B959-37092E68EFFD	Light goods vehicle	G			2002-01-01	
CF5AC90B-BCBE-4D19-9A38-27E69FC8315E	32 tonne truck	G			2002-01-01	
Asko	Asko	1			2002-01-01	2006-12-31
A2002	Asko 2002	1			2002-01-01	
Asko000001	Asko	1			2002-01-01	2007-01-31
Asko0000011	Asko	1			2002-01-01	2007-01-31
M01	Motor	1			2002-01-01	2006-12-31
M02	Transmission	1			2002-01-01	2006-12-31
M03	Support	1			2002-01-01	2006-12-31
M04	Drum	1			2002-01-01	2006-12-31
M05	Control Unit	1			2002-01-01	2006-12-31
M06	Door	1			2002-01-01	2006-12-31
M07		1			2002-01-01	2006-12-31
M08		1			2002-01-01	2006-12-31
M09		1			2002-01-01	2006-12-31
M10		1			2002-01-01	2006-12-31
M11		1			2002-01-01	2006-12-31
M12		1			2002-01-01	2006-12-31
Asko		2			2007-01-01	2011-12-31
A2007		1			2007-01-01	
M01		1			2007-01-01	2011-12-31
M02		1			2007-01-01	2011-12-31
M03		1			2007-01-01	2011-12-31
M04		1			2007-01-01	2011-12-31
M05		2			2007-01-01	2011-12-31
M06		1			2007-01-01	2011-12-31
M07		1			2007-01-01	2011-12-31
M08		1			2007-01-01	2011-12-31
M09		1			2007-01-01	2011-12-31
M10	Rear panel	1			2007-01-01	2011-12-31
M11	Holder	1			2007-01-01	2011-12-31
M12	Drain	1			2007-01-01	2011-12-31
Asko	Asko	3			2012-01-01	

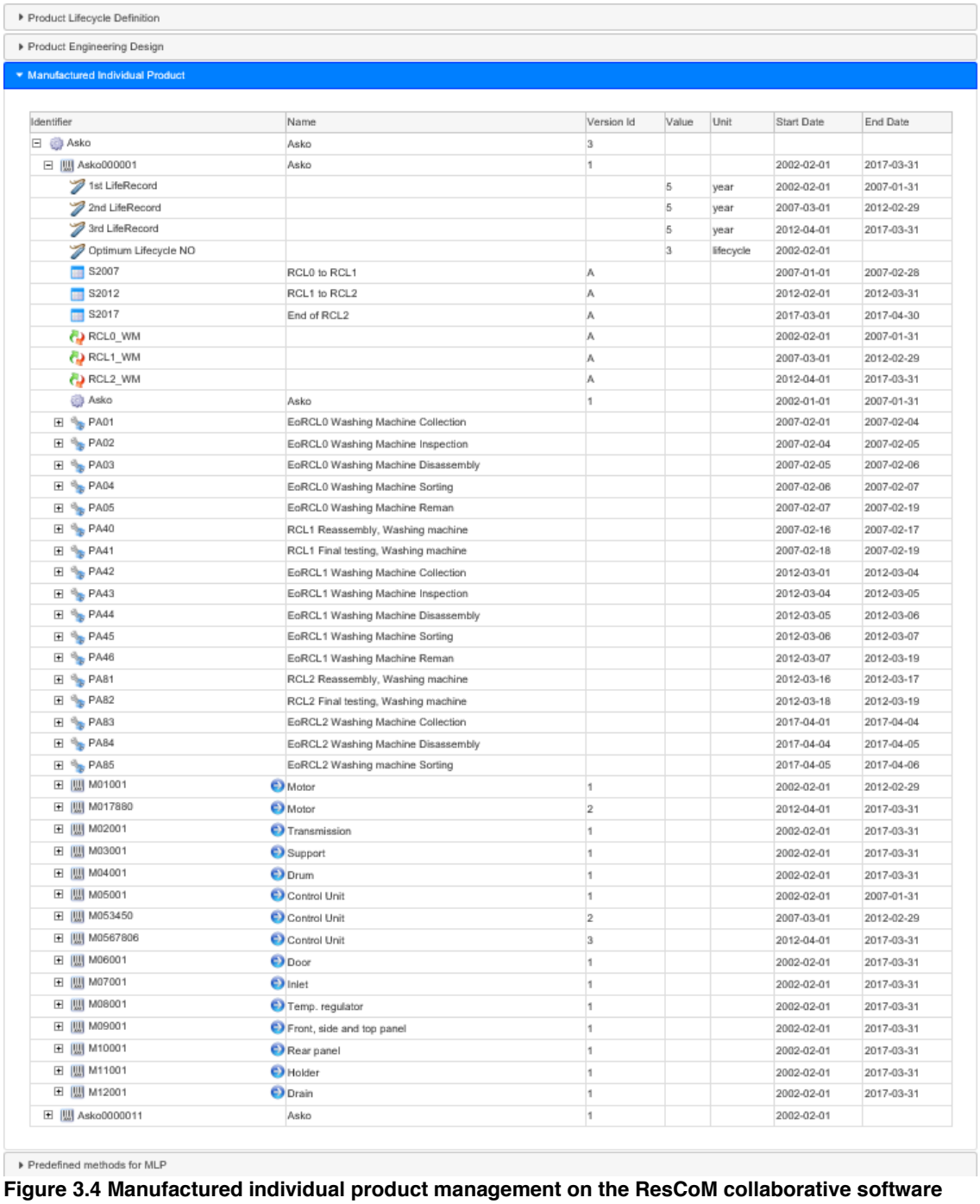
Figure 3.3 Product engineering design management on the ResCoM collaborative software platform

- ProductDesign
  - Id, version, name, description
  - useCycle
  - Document
    - Id, version, name, description, startDate and endDate
  - DesignStructure
    - startDate and endDate
    - Property
      - Id, name, value, unit, startDate and endDate
- Material
  - Id, version, name, description
- Substance
  - Id, version, name, description
- RCLElement
  - Id, version, name, description
  - startDate and endDate of RCLElementToDesign
- SerializedAsset (reference to ResCoM_InLife model)
  - Id, version, name, description
  - startDate and endDate of BasedOn

3.2.3 Manufactured Individual Product (figure 3.4)

This is mainly based on the ResCoM_Activities_and_Methods model, but also referenced to ResCoM_InLife model and ResCoM_ProductRCBreakdown_and_Design model. In this tree, one can find the following information throughout the lifecycles of the product individuals:

- SerializedAsset
  - Id, version, name, description, and productOptimumLifecycle NO
  - startDate of the 1st lifecycle and endDate of the last lifecycle
  - LifeRecord
    - Id, name, value, unit, startDate and endDate
  - SerializedAssetStructure (reference to ResCoM_InLife model)
    - startDate and endDate
  - LevelState
    - Id
- LifecyclePlan
  - Id, version, name, description
  - startDate and endDate of LifecyclePlanAssignment
- PlannedActivity
  - Id, version, name, description, startDate and endDate
- ActivityActual
  - Id, version, name, description, startDate and endDate
- ProductDesign (reference to ResCoM_InLife model)
  - Id, version, name, description
  - startDate and endDate of BasedOn
- RCLDefinition (reference to ResCoM_InLife model)
  - Id, version, name, description
  - startDate and endDate of BreakdownElementToSerializedAsset
- RCLElement (reference to ResCoM_InLife model)
  - Id, version, name, description
  - startDate and endDate of BreakdownElementToSerializedAsset



Identifier	Name	Version Id	Value	Unit	Start Date	End Date
Asko	Asko	3				
Asko000001	Asko	1			2002-02-01	2017-03-31
1st LifeRecord			5	year	2002-02-01	2007-01-31
2nd LifeRecord			5	year	2007-03-01	2012-02-29
3rd LifeRecord			5	year	2012-04-01	2017-03-31
Optimum Lifecycle NO			3	lifecycle	2002-02-01	
S2007	RCL0 to RCL1	A			2007-01-01	2007-02-28
S2012	RCL1 to RCL2	A			2012-02-01	2012-03-31
S2017	End of RCL2	A			2017-03-01	2017-04-30
RCL0_WM		A			2002-02-01	2007-01-31
RCL1_WM		A			2007-03-01	2012-02-29
RCL2_WM		A			2012-04-01	2017-03-31
Asko	Asko	1			2002-01-01	2007-01-31
PA01	EoRCL0 Washing Machine Collection				2007-02-01	2007-02-04
PA02	EoRCL0 Washing Machine Inspection				2007-02-04	2007-02-05
PA03	EoRCL0 Washing Machine Disassembly				2007-02-05	2007-02-06
PA04	EoRCL0 Washing Machine Sorting				2007-02-06	2007-02-07
PA05	EoRCL0 Washing Machine Reman				2007-02-07	2007-02-19
PA40	RCL1 Reassembly, Washing machine				2007-02-16	2007-02-17
PA41	RCL1 Final testing, Washing machine				2007-02-18	2007-02-19
PA42	EoRCL1 Washing Machine Collection				2012-03-01	2012-03-04
PA43	EoRCL1 Washing Machine Inspection				2012-03-04	2012-03-05
PA44	EoRCL1 Washing Machine Disassembly				2012-03-05	2012-03-06
PA45	EoRCL1 Washing Machine Sorting				2012-03-06	2012-03-07
PA46	EoRCL1 Washing Machine Reman				2012-03-07	2012-03-19
PA81	RCL2 Reassembly, Washing machine				2012-03-16	2012-03-17
PA82	RCL2 Final testing, Washing machine				2012-03-18	2012-03-19
PA83	EoRCL2 Washing Machine Collection				2017-04-01	2017-04-04
PA84	EoRCL2 Washing Machine Disassembly				2017-04-04	2017-04-05
PA85	EoRCL2 Washing machine Sorting				2017-04-05	2017-04-06
M01001	Motor	1			2002-02-01	2012-02-29
M017880	Motor	2			2012-04-01	2017-03-31
M02001	Transmission	1			2002-02-01	2017-03-31
M03001	Support	1			2002-02-01	2017-03-31
M04001	Drum	1			2002-02-01	2017-03-31
M05001	Control Unit	1			2002-02-01	2007-01-31
M053450	Control Unit	2			2007-03-01	2012-02-29
M0567806	Control Unit	3			2012-04-01	2017-03-31
M06001	Door	1			2002-02-01	2017-03-31
M07001	Inlet	1			2002-02-01	2017-03-31
M08001	Temp. regulator	1			2002-02-01	2017-03-31
M09001	Front, side and top panel	1			2002-02-01	2017-03-31
M10001	Rear panel	1			2002-02-01	2017-03-31
M11001	Holder	1			2002-02-01	2017-03-31
M12001	Drain	1			2002-02-01	2017-03-31
Asko0000011	Asko	1			2002-02-01	

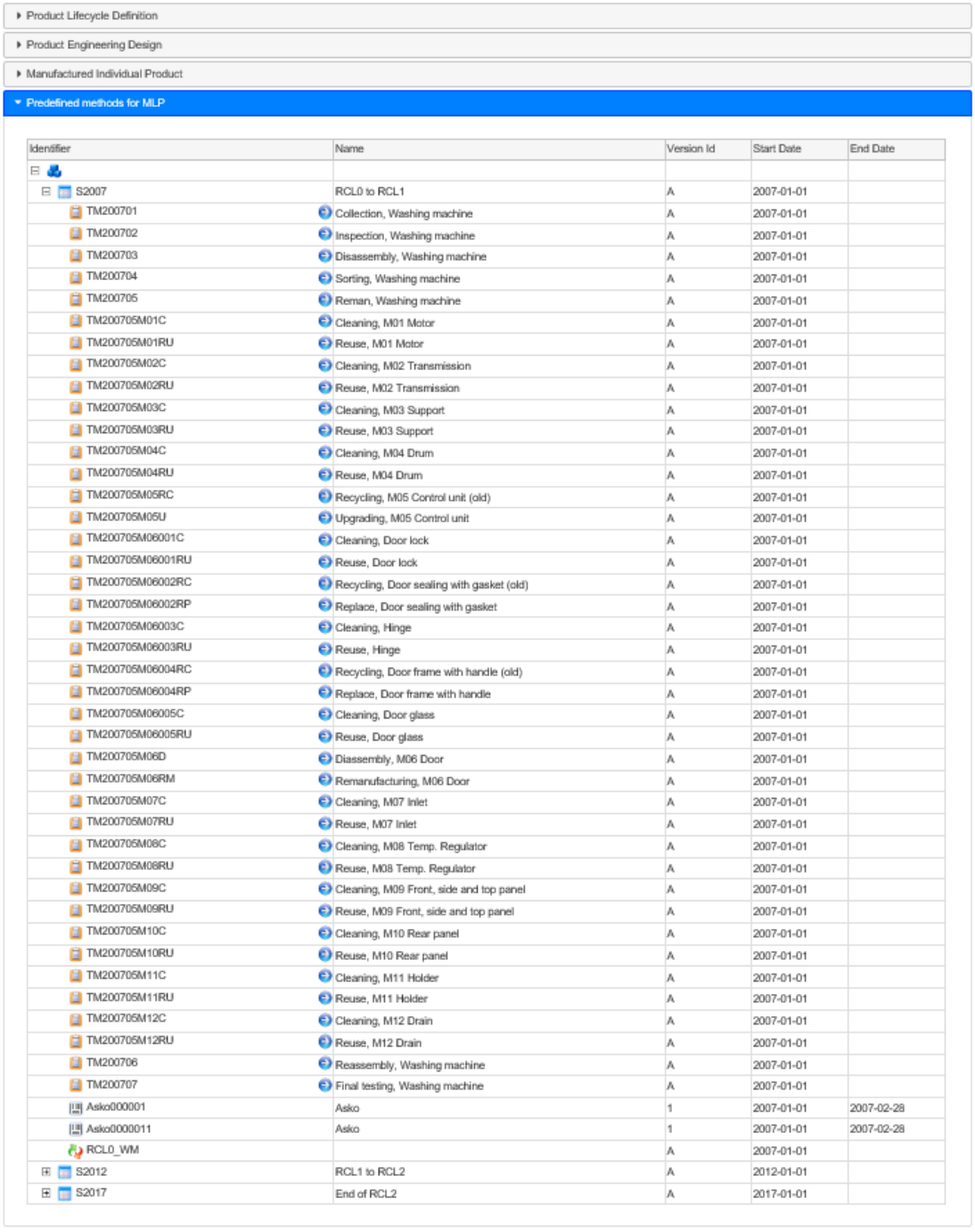
Figure 3.4 Manufactured individual product management on the ResCoM collaborative software platform

3.2.4 Predefined methods for MLP (figure 3.5)

This is mainly based on the ResCoM_Activities_and_Methods model. In this tree, one can find the following information regarding the lifecycle plans of the product design and product individuals:

- LifecyclePlan
  - Id, version, name, description
- TaskMethod
  - Id, version, name, description
- SerializedAsset

  - Id, version, name, description
  - startDate and endDate of *Activity*
- ResourceConservativeLevelDefinition
  - startDate and endDate of LifecyclePlanAssignment



Identifier	Name	Version Id	Start Date	End Date
S2007	RCL0 to RCL1	A	2007-01-01	
TM200701	Collection, Washing machine	A	2007-01-01	
TM200702	Inspection, Washing machine	A	2007-01-01	
TM200703	Disassembly, Washing machine	A	2007-01-01	
TM200704	Sorting, Washing machine	A	2007-01-01	
TM200705	Reman, Washing machine	A	2007-01-01	
TM200705M01C	Cleaning, M01 Motor	A	2007-01-01	
TM200705M01RU	Reuse, M01 Motor	A	2007-01-01	
TM200705M02C	Cleaning, M02 Transmission	A	2007-01-01	
TM200705M02RU	Reuse, M02 Transmission	A	2007-01-01	
TM200705M03C	Cleaning, M03 Support	A	2007-01-01	
TM200705M03RU	Reuse, M03 Support	A	2007-01-01	
TM200705M04C	Cleaning, M04 Drum	A	2007-01-01	
TM200705M04RU	Reuse, M04 Drum	A	2007-01-01	
TM200705M05RC	Recycling, M05 Control unit (old)	A	2007-01-01	
TM200705M05U	Upgrading, M05 Control unit	A	2007-01-01	
TM200705M06001C	Cleaning, Door lock	A	2007-01-01	
TM200705M06001RU	Reuse, Door lock	A	2007-01-01	
TM200705M06002RC	Recycling, Door sealing with gasket (old)	A	2007-01-01	
TM200705M06002RP	Replace, Door sealing with gasket	A	2007-01-01	
TM200705M06003C	Cleaning, Hinge	A	2007-01-01	
TM200705M06003RU	Reuse, Hinge	A	2007-01-01	
TM200705M06004RC	Recycling, Door frame with handle (old)	A	2007-01-01	
TM200705M06004RP	Replace, Door frame with handle	A	2007-01-01	
TM200705M06005C	Cleaning, Door glass	A	2007-01-01	
TM200705M06005RU	Reuse, Door glass	A	2007-01-01	
TM200705M06D	Diassembly, M06 Door	A	2007-01-01	
TM200705M06RM	Remanufacturing, M06 Door	A	2007-01-01	
TM200705M07C	Cleaning, M07 Inlet	A	2007-01-01	
TM200705M07RU	Reuse, M07 Inlet	A	2007-01-01	
TM200705M08C	Cleaning, M08 Temp. Regulator	A	2007-01-01	
TM200705M08RU	Reuse, M08 Temp. Regulator	A	2007-01-01	
TM200705M09C	Cleaning, M09 Front, side and top panel	A	2007-01-01	
TM200705M09RU	Reuse, M09 Front, side and top panel	A	2007-01-01	
TM200705M10C	Cleaning, M10 Rear panel	A	2007-01-01	
TM200705M10RU	Reuse, M10 Rear panel	A	2007-01-01	
TM200705M11C	Cleaning, M11 Holder	A	2007-01-01	
TM200705M11RU	Reuse, M11 Holder	A	2007-01-01	
TM200705M12C	Cleaning, M12 Drain	A	2007-01-01	
TM200705M12RU	Reuse, M12 Drain	A	2007-01-01	
TM200706	Reassembly, Washing machine	A	2007-01-01	
TM200707	Final testing, Washing machine	A	2007-01-01	
Asko000001	Asko	1	2007-01-01	2007-02-28
Asko0000011	Asko	1	2007-01-01	2007-02-28
RCL0_WM		A	2007-01-01	
S2012	RCL1 to RCL2	A	2012-01-01	
S2017	End of RCL2	A	2017-01-01	

Figure 3.5 Predefined methods for MLP management on the ResCoM collaborative software platform

3.3 ResCoM tools integration

According to the requirement on the collaborative software platform, it should support the decision making tools (i.e. ResCoM tools) for making decisions such as component-level

recycling vs. product-level recycling based on the analysis of the product design and the reverse value chain capabilities. The decision making tools applied to the industrial case is shown in the "ResCoM tools applied to the industrial case" of figure 3.1. The layout of the data sharing for Bugaboo is shown in figure 3.6. Note that not all tools are applied to each case, one can choose the tools to use as needed for each case. The ResCoM tools and integration are described in chapter 3.3.1 and chapter 3.3.2.

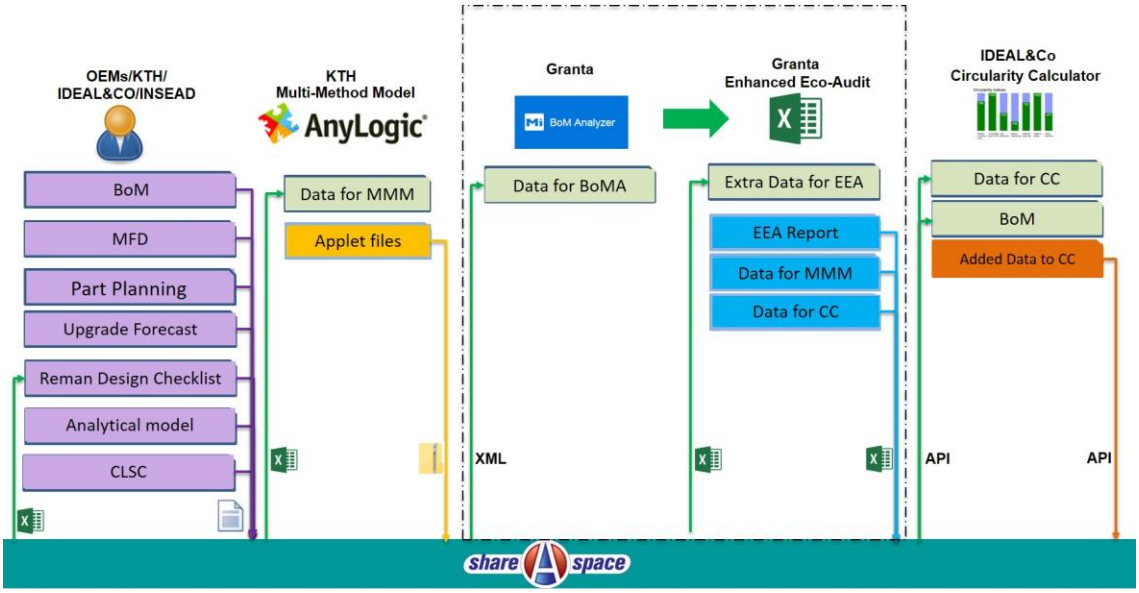


Figure 3.6 Layout of data sharing on the ResCoM collaborative software platform

3.3.1 Tools that ONLY store data on the platform

- **BoMs** from OEM
- **Part planning (IDEAL&CO)** tool aids designers to analyse and plan for carry-over parts across multiple product use-cycles.
- **Modular function deployment for multiple lifecycle (MFD developed by KTH)** tool analyzes the future multiple lifecycles design.
- **Upgrade Forecast (IDEAL&CO)** aids designers in forecasting the trend and specifications of the future product design.
- The analytical model **(INSEAD)** tool allows to quickly check the profitability and environmental impact of refurbishing.

Besides those tools, information documented in different deliverables, e.g. business model and closed loop supply chain, is also captured, structured and stored in the platform.

3.3.2 Tools that BOTH store and utilize data on the platform

Reman design checklist (IDEAL&CO) accesses the future remanufacturability of the product design.

- **Reman Design Checklist**
  - Tool type: Excel
  - Way of integration: copy and paste the data in an Excel file generated from the platform to the Reman Design Checklist
  - Data utilized
    - BoM with names of the components of the product

The **circularity calculator (IDEAL&CO)** provides the possibility to envision different conceptual design solutions and business models.

- **Circularity Calculator**

- Tool type: Online IT-tools
- Way of integration: Import the data by calling ResCoM APIs (developed in D5.8) on the Circularity Calculator
- Data utilized
    - Simplified BoM with parts of the product
    - Mass of each part
    - Quantity of each part in the product
    - Material of part
- Result imported to the platform
    - Analysis result including the revised data

The **enhanced material information management (MI BoM Analyzer and Enhance Eco-Audit tools of Granta)** supports illustrating the benefits, in terms of resource efficiency, CO2 production and cost, of adopting closed loop product design approach at the design stage.

- **MI BoM Analyzer**
    - Tool type: Online IT-tool
    - Way of integration: import the XML file generated from the platform to the MI BoM Analyzer
    - Data utilized
        - Detailed BoM with components and parts of the product
        - Mass of each part
        - Quantity of each part in the product
        - Material of part
        - Process of the material
        - Process mass of the material
        - Reuse feedstock of the part
- **Enhanced Eco-Audit**
    - Tool type: Excel
    - Way of integration: copy and paste the data in an Excel file generated from the platform to the Enhanced Eco-Audit
    - Data utilized
        - Collection rate after use
        - Use cycle
        - Transport
            - Production site to use phase
            - Transport from Use phase to Remanufacturing
            - Transport from Remanufacturing to Disposal
    - Result imported to the platform
        - Enhance Eco-Audit report
        - Material of part assigned by MI database
        - Cost and CO2

Dynamic system modelling (i.e. Multimethod simulation tool: Anylogic used by KTH) tool analyses scenarios of different business model and closed-loop supply chain of the product.

- **Multi Method Model**
    - Tool type: Applet (with the Excel configure file)
    - Way of integration: copy and paste the data in an Excel file generated from the platform to the Excel configure file of the Multi Method Model
    - Data utilized
        - Cost and CO2
        - Return ratio of sold products
    - Result imported to the platform
        - Applet files including the Multi Method Model reports of each scenario

The **enhanced PLM (i.e. Product Multiple Lifecycle Management: ShareAspace of Eurostep)** is part of the platform, i.e. ShareAspace enables the management of the product in the design, manufacturing, in-service and remanufacturing throughout the product lifecycles. It captures, structures and stores the information and data as described in chapter 3.2. At the same time the platform is built on ShareAspace which integrates all the ResCoM tools listed above.

4. Conclusion

In this deliverable, the ResCoM final standard data model is documented, and it is the basis of the ResCoM collaborative software platform development, including developing the solution for managing product with multiple lifecycles based on the ShareAspace, developing the ResCoM Web API, and mappers for integrating ResCoM tools to the platform.
The development of the platform based on the data model has been verified by the industrial pilots in so far as it supports the envisioned ResCoM processes and information. The data model is applicable to an even wider scope than those industrial pilots needed. For example the ResCoM_Requirements model has not been verified by the industrial pilots, since the requirements are supposed to be listed, stored and structured in some requirement management tools and then imported to the collaborative software platform or directly created in the collaborative software platform by the designers.In the industrial pilots the requirements are documented in other deliverables and no list has been clearly made. In the ResCoM_ProductRCBreakdown_and_Design model, it supports modularization, and the data model was tested and validated through an integration with a standalone modularization tool called PALMA. However it is not a “ResCoM tool” to be included in the ResCoM collaborative software platform, but test and validation proved that the ResCoM software platform has the capability to integrate any other standalone modularization tool based on the data model.
The final standard data model is ready for communicating with OASIS and ISO standard. The standardization process through ISO and/or OASIS will be described in D7.7 standardization report.

References

[1] ISO International Organization for Standardization (2012) *ISO 10303-233:2012. Industrial automation systems and integration – Product data representation and exchange – Part 233: Application protocol: Systems engineering.* Geneva.

[2] Dunford, J., Bergtröm, P. (2007). *Standards-based PLM: Re-engineering the Aftermarket with PLCS Part 2 Technologies and Deployments*. Eurostep and John Stark Associates Technology White Paper

[3] Rashid, A., et al., (2013). *Resource Conservative Manufacturing: an essential change in business and technology paradigm for sustainable manufacturing.* Journal of Cleaner Production

This project has received funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement No 603843

Appendix: Icon library

This icon library provides the icons shown in the ResCoM collaborative software platform and the corresponding objects in the ResCoM final standard data model

Icon	Objects
	ResourceConservativeBreakdown
	ResourceConservationLevelDefinition
	RCLElement
	ProductDesign/Material/Substance
	SerializedAsset
	LifecyclePlan
	TaskMethod
	Property/LifecyclePeriod
	Document
	PlannedActivity
	ActivityActual
	LevelState

This project has received funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement No 603843