Feedback mechanisms Across the Lifecycle for Customer-driven Optimization of iNnovative product-service design

Acronym: FALCON
Project No: 636868

FoF-05-2014
Duration: 2015/01/01-2017/12/31

PROJECT DELIVERABLE 3.2:

The FALCON Ontology

Content : This deliverable explains the FALCON semantic model as an ontology presenting the domain of interest in the context of the FALCON project. The document presents the basic entities of the project and models relevant structures covering the multi domain knowledge of Product-Services managed.

Versioning and contribution history

Version	Description	Contributors
0.1	Preparation of deliverable outline	Sangje Cho (EPFL)
0.2	Preparation of the first draft	Sangje Cho (EPFL)
0.3	Second draft	Sangje Cho (EPFL)
1.0	Final Version complied	Sangje Cho (EPFL)
Final	Final check, formatting and change the dissemination status from Confidential to Public as stated in DoA	Indah Lengkong (BIBA) and Karl Hribernik (BIBA)

Reviewer

Name	Affiliation
Marco Franke	BIBA

Table of contents

List of Figures

List of Tables

Executive Summary

This is the document for FALCON Project Deliverable D3.2 – The FALCON ontology. The deliverable consists of the results of T 3.2 - Semantic model design and implementation that are an upper ontology as an upper template for all FALCON business cases as well as future business cases, and specific ontologies describing the domain interests in each business scenario with all the layers of FALCON ontology including all its entities and semantic structure.

The FALCON ontology has been designed based on the Basic Formal Ontology (BFO) and the other reference ontologies. To represent the domain knowledge of the Product-Service System, it consists of various layers: PSS layer represents the structure of PSS in the context of FALCON project, Data source layer has as entities, Material layer, and Attribute layer. One of the fundamentals to meet the requirements of business partners is answering the list of questions of business stories described in Deliverable 5.2/6.2/7.2/8.2. In addition, it will be adapted to meet upcoming business story requirement.

The FALCON ontology will be implemented in RDF format because it is a standard model for data interchange on the Web and is a W3C recommendation designed to standardize the definition and use of metadata-descriptions of Web-based resources. The FALCON ontology plays the role to define the structure and contact of the Triple Store, to be used to define semantic search parameters for social media, and to be used to query PUI.

In addition, this document also provide a through state of the art analysis of semantic technologies and relevant information concerning the actual implementation.

1 Introduction

1.1 Objective of WP 3

WP 3 deals with the design and implementation of FALCON semantic model and the mechanisms for intelligent filtering. The WP will provide a unified and all-spanning semantic model covering the multi domain knowledge of product-services managed through the entire lifecycle (BOL, MOL, EOL). The WP will also deal cross-sectoral knowledge exploration and filtering through mechanisms.
To summarize, main objectives of WP3 are:

- to **identify** the domain of interest, covering all relevant products, data sources, information flow resources, design and manufacturing processes, user-interface access points and dynamics of the entire system
- to **design** and implementation of the semantic model
- to **provide** a linked data integration framework that will extract, export, and harmonize data from various sources,
- to **enable** semantic enrichment (e.g. annotations, tagging) of data originating from disparate research or existing systems (PLM/CAx),
- to **provide** an intelligent engine which will allow active exploration of the linked data sets and implicit knowledge discovery
- to **support** data security and privacy.

1.2 Objective of Task 3.2

Task 3.2 develops the FALCON semantic model (i.e. Ontology) designed to serve as a common reference model for the annotation and description of the BOL, MOL and EOL phases. Re-use of existing ontologies will be envisaged towards the design of the project's ontology. The ontology will describe the basic entities of the project and model relevant structures of planning, design, engineering, manufacturing and exploitation processes. It will be able to meet the requirements of different stakeholders who need to have access to different aspects of product-related information and knowledge. Within the task, the designed ontology will be appropriately realized and implemented using standard-based languages within the sDM.

1.3 Contents of this Document

The main purpose of this document is to present the implementation of the FALCON semantic model consisting of an upper ontology and specific ontologies in the context of the FALCON project. An upper ontology is a generalization of the business scenarios serving as an upper template for all FALCON business cases as well as future business cases, and specific ontologies describe the domain interests in each business scenario. The deliverable also acts as an amendment of Deliverable D3.1 as it provides a revised version of the FALCON semantic model, including updates to the classes and their relations in the way of comparison with existing ontologies.

As the main result of T 3.2, the FALCON ontology includes the domain of interest extracted using User Story Mapping method which facilitate to meet the requirement of stake holder in the domain field. And implicit and explicit knowledge as ontology plays the role to add value for people who try to understand domain knowledge of Healthcare scenario. Meanwhile this ontology constitutes the formal representation of the FALCON semantic model and the knowledge that this model encapsulates as the part of the FALCON VOP. Therefore, codification of the knowledge will allow to exchange information regarding the Product-Service context and to be desirable to use it, in order to increase added values of FALCON platform. In addition, FALCON ontology enables integration of PEID data and Social media data. The data integration enables FALCON VOP to have semantic interoperability. Meanwhile, the FALCON ontology has the ability to assume existence of rules expressing logics. The FALCON ontology plays the role to define the structure and contact of the Triple Store, to be used to define semantic search parameters for social media, and to be used to query PUI.

The document is structured as follows. In Chapter 2, the FALCON Semantic Model is presented with a complete list of the associated concepts, object properties and data type properties. In Chapter 3, we provide a thorough state of the art analysis of semantic technologies that comprise all the required knowledge to make the right technical choices for implementing the FALCON Semantic Model as an ontology. Subsequently, Chapter 4 highlights relevant information concerning the actual implementation while Appendix A and B illustrate the serialized versions of the current version of the FALCON Ontology. Finally, Chapter 5 concludes the document by highlighting the main results achieved and the connections with future activities.

2 THE FALCON SEMANTIC MODEL

This section presents the FALCON semantic model with foundations constituted on which the FALCON Ontology is built. The FALCON semantic model consists of an upper ontology and five specific ontologies aligned to its specializations, one for each business scenario. Both the upper ontology and the specific ontologies are elaborated in the following sub-sections, presenting all their classes, object and data properties together with annotations.

In order to design the taxonomies of describing Product Service System (PSS), the agile software development called User Story Mapping (USM) has been applied to (i) define the application domain boundaries and (ii) capture elements definition. Deliverable D3.1 provided definition of the FALCON domain and sort of reference glossary. For generally applicable solutions, generalization of the business scenario is serving as an upper ontology for not only all the FALCON business scenarios, but also further business scenarios which will be the application of the FALCON platform. By composing a top-level overview, abstract concepts facilitate to perform system architecture planning and optimization. After the FALCON-wide reference semantic vocabulary, the list of classes is updated, compared with existing ontology such as BFO, Linked Design ontology and Diversity general category/PSS.

This report provides an amendment of the deliverable D3.1 as it provides a revised version of the FALCON ontology in the way of comparison with existing ontologies such as Basic Formal Ontology (BFO), Linked Design ontology and Diversity general category/PSS which have been refereed in the development of the FALCON ontology.

2.1 Upper Ontology

All the products, processes, data sources and resources are captured by USM and updated through comparison with existing PSS ontologies. In order to make concrete definitions of the business scenario domain, the FALCON–wide reference semantic vocabulary has been created through a number of iterations. Generalization of all the FALCON business scenarios has been identified from similarities between domains of each business scenario that will lead to a future design of the new ontology template which is mainly responsible for the elaboration of the FALCON semantic model.

FALCON ontology has various kinds of layers for which the main aspects are PSS, Data Source, Material Entities and Attribute, in terms of domain representation.

First of all, the PSS layer has elements describing the PSS domain. Further, Goedkoop et al. (1999) define the key elements of PSS as follows:

- **Product**: a tangible commodity manufactured to be sold which are continuants
- **Service**: an activity done for others with an economic value and often done on a commercial basis which are occurrent
- **Product system**: a set of material products needed to jointly fulfil a user`s needs
- **Product Service System** : a marketable set of products and services capable of jointly fulfilling a user`s need

Considering these elements, FALCON semantic framework has additional elements defined as follows:

- **Software element** : Sequences of abstract problem statements that describe computations to be performed by a machine (Council & Heineman, 2001)
- **Operational element** : Service elements in terms of maintenance, development, usage recommendation and so on.
- **Part** : Composing element of a Product in the bottom level
- **Sub-assembly** : Composing element of a Product consisting of parts
- **PEID**: Product Embedded Information Device

Figure 1 depicts the structure of a general PSS in the matter of PSS layer of FALCON ontology.

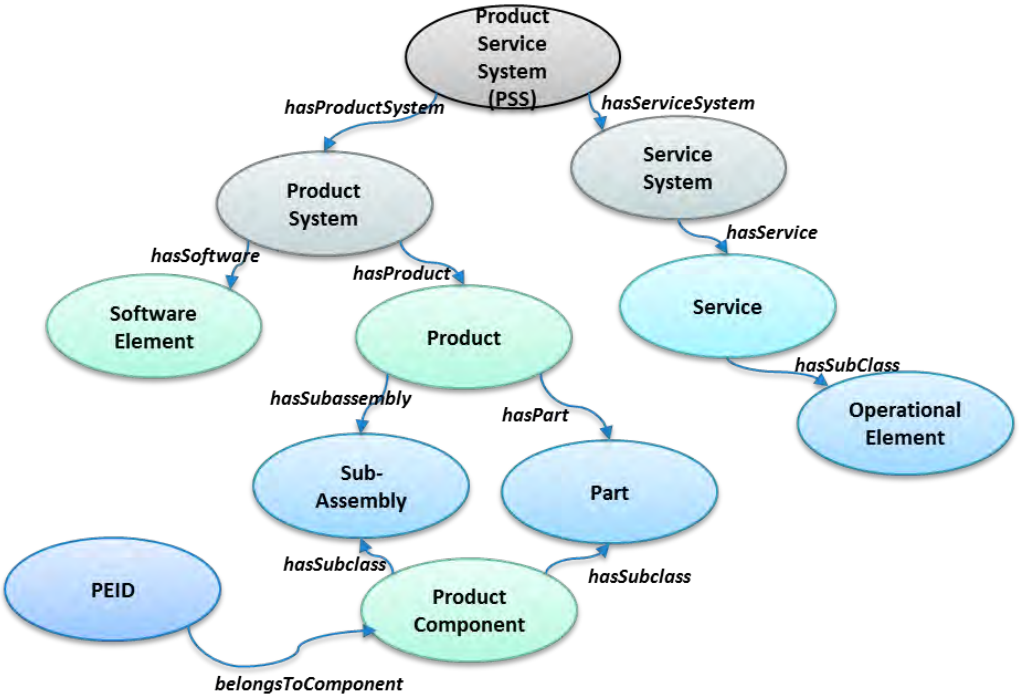

Figure 1. PSS layer of FALCON semantic model

Second, the Data source layer has two kinds of advanced technologies during MOL as follows, which are associated with the FALCON project :

- **User Feedback** : Feedback from help desk or social media using a labelling system to add semantic structure which are User complaints and User recommendations
- **PEID data** : Data generated by PEID which are log files or log history
- **Software Log** : Data generated by Software elements which are log files or log history

Third, the material entity layer has elements describing materials which should be used for service executions.

- **Actor** : All roles that human can take in a sense of requirements or responsibility. It is a list of types of interactions that an individual can have with FALCON platform
- **Resource** : The lists of resources which should be used for service executions in terms of application, document, equipment, and material.

Last, the Attribute layer has elements as follows:

- **PSS type** : Classes of PSS in terms of Product Oriented PSS, Use Oriented PSS, and Result oriented PSS (Cook, 2004)
- **PLC** : Processes describing Product Life Cycle such as BOL, MOL, and EOL
- **Need/Requirement** : Summary of requirements of customers or function
- **Benefit** : Added values which are sorts of Performance Improvement, Cost reduction, Risk reduction and Eco friendliness (Dong at al., 2011) service instances have

FALCON semantic framework has the functionality structure (See Fig. 2). From Data source, Need/Requirement will be extracted in a sense of customer recommendation/complaint and functional requirement which will be classifications of user behaviours/product states or customer complaint/recommendation. A service instance has relations with Resource and Benefit to represent benefits of a service instance and resources used for service executions. In addition, a relation between Service system and Actor represents what actor takes into account of a specific service instance.

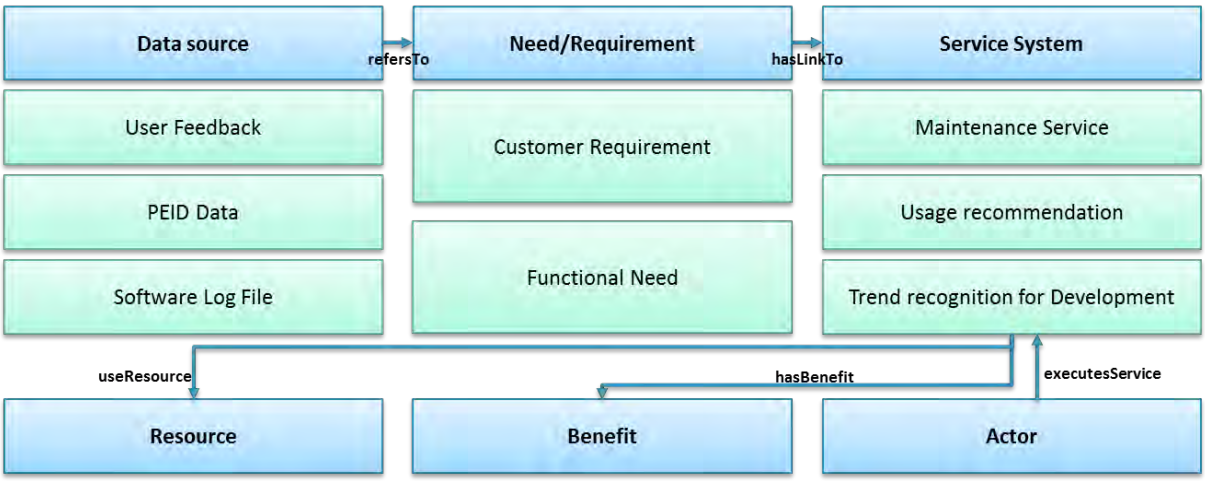

Figure 2. FALCON functionality structure

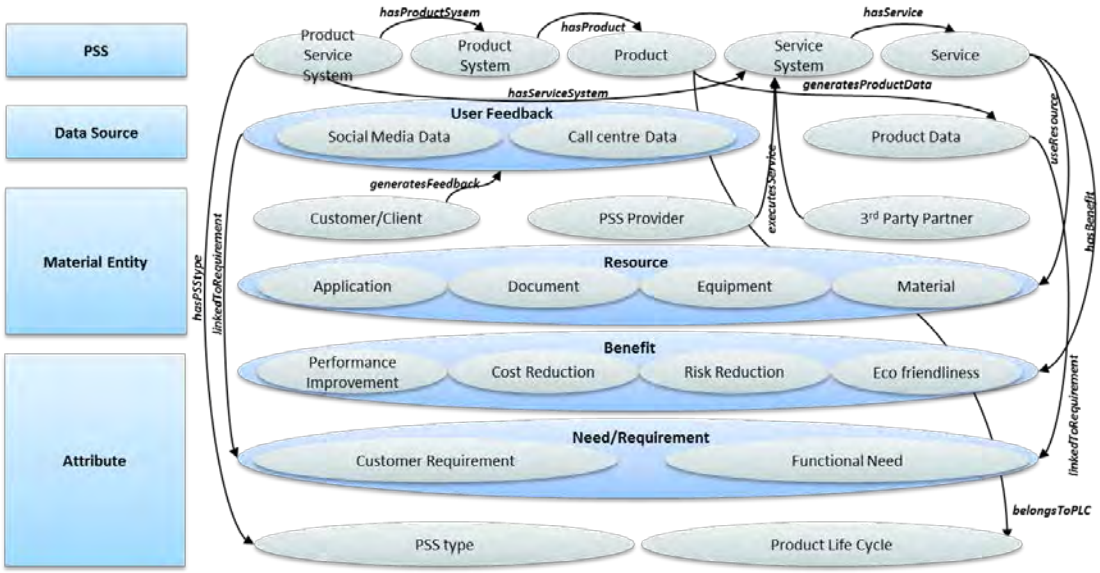

Figure 3. FALCON semantic Framework

To summarize FALCON semantic framework (See Fig. 3), first in PSS layer, a PSS has Product systems and Service systems which have products or services as subclasses. Second, in Data source layer, User feedback (i.e. social media data and call centre data) is generated by customers and product data is generated by PEID comprised of products or software components. On the other hand, Resources are used for the consumption of services. In addition, in Material Entity layer, Services has benefits in terms of performance improvement, cost reduction, risk reduction and eco friendliness. Moreover, from the Data sources, customer requirements or Functional needs which are the classifications of the data source will be extracted. Last, PSS has a PSS type and Product has a Product-Lifecycle.

Entities in the FALCON semantic framework are arranged based on the Basic Formal Ontology (BFO) (See Fig. 4) which is a formal ontology framework developed by Barry Smith and his associates (Smith at al., 2014). In BFO, there are two varieties which are continuants comprehending continuant entities such as three-dimensional enduring objects and occurrent comprehending processes conceived as extended through (or as spanning) time. BFO thus incorporates both three-dimensionalist and four-dimensionalist perspectives on reality within a single framework.

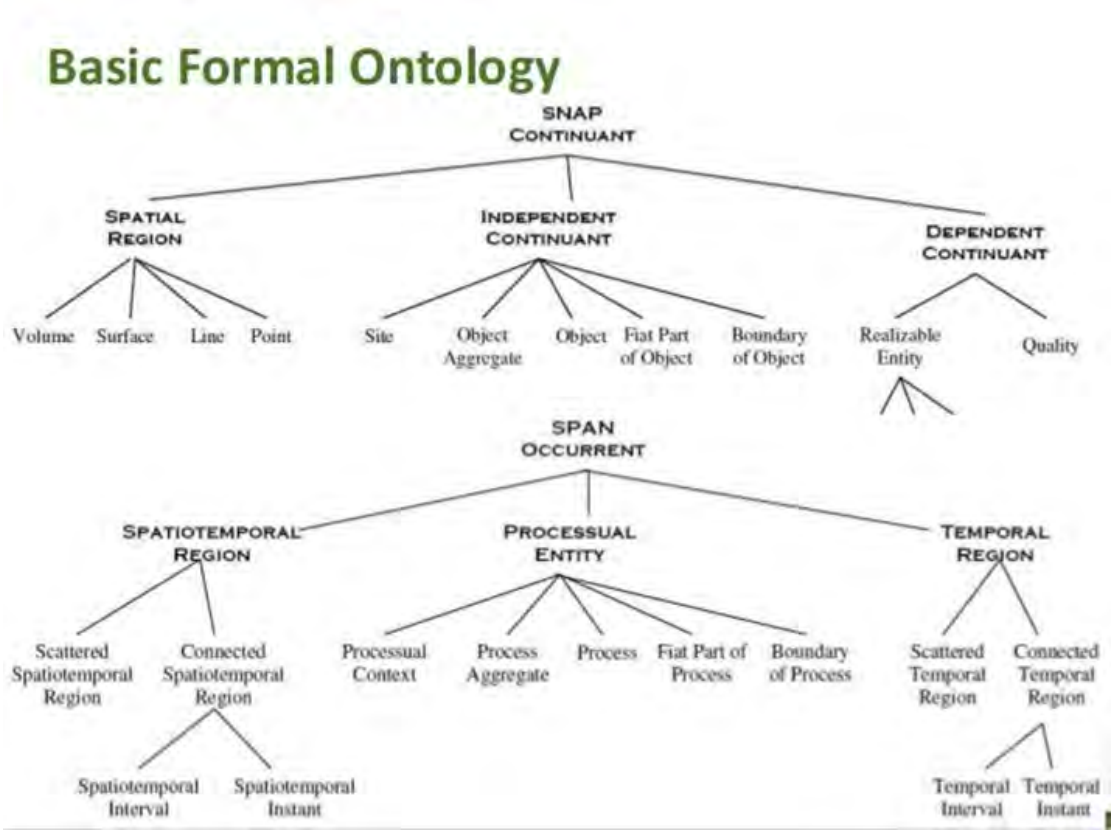

Figure 4. Basic Formal Ontology

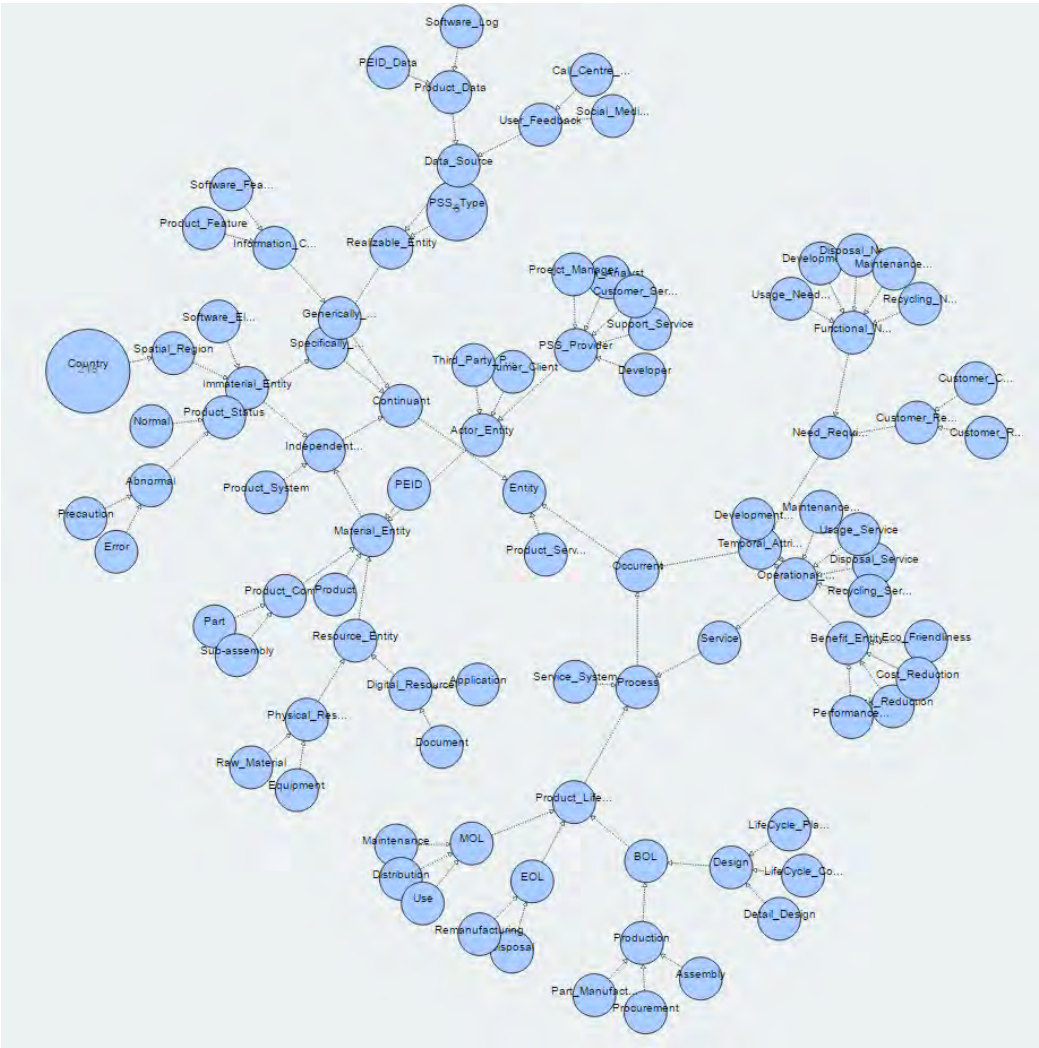

Figure 5. FALCON semantic Framework arranged based upon BFO

FALCON semantic framework will be in the form of an ontology network where each business scenario specific ontology will be connected to an upper ontology through generalization. The graph representation of the FALCON upper ontology is presented in Fig 5.

2.1.1 Classes

Table 1 lists the upper ontology classes, their sub-classes and their description.

Table 1: Upper Ontology classes

High Level Class	*Sub-class*	*Sub-class*	*Sub-class*	*Sub-class*	*Sub-class*	*Description*
Product Service System	-	-	-	-	-	Marketable sets of products and services capable of jointly fulfilling a user's need
Continuant	Generically Dependent Continuant	Information Content Entity	Product Feature	-	-	Groups all the product features and information
			Software Feature	-	-	Groups all the Software features and information
	Independent Countinuant	Immaterial Entity	Software Element	-	-	Sequences of abstract problem statements that describe computations to be performed by a machine
			Spatial Region	Country	-	Groups all the list of countries
		Material Entity	Actor Entity	Consumer Client	-	All roles that human can take in a sense of requirements or responsibility. It is a list of types of interactions that an individual can have with FALCON platform
				PSS Provider	Customer Service	
					Data Analyst	
					Developer	
					Project Manager	
					Support Service	

				Third Party Partner	-	
			PEID	-	-	Groups all the Product Embedded Information Devices
			Product	-	-	Tangible commodities manufactured to be sold which are continuants
			Product Component	Part	-	Composing elements of a Product in the bottom level
				Sub-assembly	-	Composing elements of a Product consisting of parts
			Resource Entity	Digital Resource	Application	The lists of resources which should be used for service executions in terms of application, document, equipment, and material.
					Document	
				Physical Resource	Equipment	
					Raw Material	
		Product System	-	-	-	A set of material products needed to jointly fulfil a user`s needs
	Specifically Dependent Continuant	Product Status	Abnormal	Error	-	Groups all the statuses of Product
				Precaution	-	
			Normal	-	-	
		Realizable Entity	Data Source	Product Data	PEID Data	Data generated by a PEID sensor
					Software Log	Data generated by a software element

				User Feedback	Call Centre Data	User feedback from the call centre or the help desk
					Social Media Data	User feedback from the social media
			PSS Type	-	-	Classes of PSS in terms of Product Oriented PSS, Use Oriented PSS, and Result oriented PSS
Occurrent	Process	Product Life Cycle	BOL	Design	Detail Design	Processes describing Product Life Cycle such as BOL, MOL, and EOL
					LifeCycle Conceptual Design	
					LifeCycle Planning	
				Production	Assembly	
					Part Manufacturing	
					Procurement	
			EOL	Disposal	-	
				Remanufacturing	-	
			MOL	Distribution	-	
				Maintenance	-	
				Use	-	
		Service	Operational Element	Development Service	-	Groups all the service elements for the development
				Disposal Service	-	Groups all the service elements for the disposal
				Maintenance Service	-	Groups all the service elements for the maintenance

				Recycling Service	-	Groups all the service elements for the recycling
				Usage Service	-	Groups all the service elements for usage
		Service System	-	-	-	A set of operational elements needed to jointly fulfil a user`s needs
	Generically_Dependent_Occurrent	Benefit Entity	Cost Reduction	-	-	Benefit concept represents a service which can include the value proposition of each service which includes performance improvement, cost reduction, risk reduction, eco friendliness, as some key indicators
			Eco Friendliness	-	-	
			Performance Improvement	-	-	
			Risk Reduction	-	-	
		Need Requirement Entity	Customer Requirement Entity	Customer Complaint Entity	-	Need/Requirement have a role which could be results of analysis and triggers of service instances based on the analysis of data source in terms of customer recommendation, complaint, and functional requirement
				Customer Recommendation Entity	-	
			Functional Need Entity	Development Need Entity	-	
				Disposal Need Entity	-	
				Maintenance Need Entity	-	
				Recycling Need Entity	-	
				Usage Need Entity	-	

2.1.2 Object Properties

Table 2 lists the object properties of the upper ontology. For each property, the domain and range are defined.

Table 2: Upper Ontology Object Properties

Relation	*Domain*	*Range*
belongsToComponent	PEID	Product_Component
belongsToPLC	Product	Product_Life_Cycle
excutesDevelopmentService	Developer	Development_Service
excutesDisposalService	Customer_Service	Disposal_Service
excutesMaintenanceSerivce	Support_Service	Maintenance_Service
excutesRecyclingService	Customer_Service	Recycling_Service
excutesUsageService	Customer_Service	Usage_Service
generatesCallCentreData	Consumer_Client	Call_Centre_Data
generatesPEIDData	PEID	PEID_Data
generatesSocialMediaData	Consumer_Client	Social_Media_Data
generatesSoftwareLog	Software_Element	Software_Log
hasBenefit	Service	Benefit_Entity
hasLinkToDevelopment	Development_Need_Entity	Development_Service
hasLinkToDisposal	Disposal_Need_Entity	Disposal_Service
hasLinkToMaintenance	Maintenance_Need_Entity	Maintenance_Service
hasLinkToRecycling	Recycling_Need_Entity	Recycling_Service
hasLinkToUsage	Usage_Need_Entity	Usage_Service
hasPart	Product	Part
hasProduct	Product_System	Product
hasProductFeature	Product	Product_Feature
hasProductStatus	Product	Product_Status
hasProductSystem	Product_Service_System	Product_System
hasPSSType	Product_Service_System	PSS_Type
hasService	Service_System	Service

hasServiceSystem	Product_Service_System	Service_System
hasSoftware	Product_System	Software_Element
hasSoftwareFeature	Software_Element	Software_Feature
hasSubassembly	Product	Sub-assembly
isLocatedIn	Product	Country
isPlacedIn	Actor_Entity	Country
mentionsAboutDevelopment	Customer_Requirement_Entity	Development_Service
mentionsAboutDisposal	Customer_Requirement_Entity	Disposal_Service
mentionsAboutMaintnenace	Customer_Requirement_Entity	Maintenance_Service
mentionsAboutRecycling	Customer_Requirement_Entity	Recycling_Service
mentionsAboutUsage	Customer_Requirement_Entity	Usage_Service
providesComplaintResult	Data_Analyst	Customer_Complaint_Entity
providesDevelopmentResult	Data_Analyst	Development_Need_Entity
providesDisposalResult	Data_Analyst	Disposal_Need_Entity
providesMaintenanceResult	Data_Analyst	Maintenance_Need_Entity
providesRecommendationResult	Data_Analyst	Customer_Recommendation_Entity
ProvidesRecyclingResult	Data_Analyst	Recycling_Need_Entity
providesUsageResult	Data_Analyst	Usage_Need_Entity
refersToComplaint	User_Feedback	Customer_Complaint_Entity
refersToDevelopment	Data_Source	Development_Need_Entity
refersToDisposal	Data_Source	Disposal_Need_Entity
refersToMaintenance	Data_Source	Maintenance_Need_Entity
refersToPart	Abnormal	Part
refersToRecommendation	User_Feedback	Customer_Recommendation_Entity
refersToRecycling	Data_Source	Recycling_Need_Entity
refersToSubassembly	Abnormal	Sub-assembly
refersToUsage	Data_Source	Usage_Need_Entity
usesApplication	Service	Application
usesDocument	Service	Document
usesEquipment	Service	Equipment
usesRawMaterial	Service	Raw_Material

2.1.3 Datatype Properties

Table 3 lists the datatype properties of the upper ontology. For each property, the domain and range are defined.

Table 3: Upper Ontology datatype properties

Domain	*Property*	*Type*	*Range*
Consumer_Client	ClientAddress	Datatype	xsd:string
	ClientContact	Datatype	xsd:string
	ClientEmail	Datatype	xsd:string
	ClientID	Datatype	xsd:int
	ClientName	Datatype	xsd:string
	ClientWarranty	Datatype	xsd:int
Customer_Service	CustomerSerivceAccessAuthority	Datatype	xsd:string
	CustomerSerivceAddress	Datatype	xsd:string
	CustomerSerivceAffiliation	Datatype	xsd:string
	CustomerSerivceContact	Datatype	xsd:string
	CustomerSerivceEmail	Datatype	xsd:string
	CustomerSerivceID	Datatype	xsd:string
	CustomerSerivceName	Datatype	xsd:string
Data_Analyst	DataAnalystAccessAuthority	Datatype	xsd:string
	DataAnalystAddress	Datatype	xsd:string
	DataAnalystAffiliation	Datatype	xsd:string
	DataAnalystContact	Datatype	xsd:string
	DataAnalystEmail	Datatype	xsd:string
	DataAnalystID	Datatype	xsd:int
	DataAnalystName	Datatype	xsd:string
Developer	DeveloperAccessAuthority	Datatype	xsd:string
	DeveloperAddress	Datatype	xsd:string
	DeveloperAffiliation	Datatype	xsd:string
	DeveloperContact	Datatype	xsd:string
	DeveloperEmail	Datatype	xsd:string
	DeveloperID	Datatype	xsd:int

	DeveloperName	Datatype	xsd:string
Project_Manager	ManagerAccessAuthority	Datatype	xsd:string
	ManagerAddress	Datatype	xsd:string
	ManagerAffiliation	Datatype	xsd:string
	ManagerContact	Datatype	xsd:string
	ManagerEmail	Datatype	xsd:string
	ManagerID	Datatype	xsd:int
	ManagerName	Datatype	xsd:string
Support_Service	SupportServiceAccessAuthority	Datatype	xsd:string
	SupportServiceAddress	Datatype	xsd:string
	SupportServiceAffiliation	Datatype	xsd:string
	SupportServiceContact	Datatype	xsd:string
	SupportServiceEmail	Datatype	xsd:string
	SupportServiceID	Datatype	xsd:int
	SupportServiceName	Datatype	xsd:string
Third_Party_Partner	ThridPartyAccessAuthority	Datatype	xsd:string
	ThridPartyAddress	Datatype	xsd:string
	ThridPartyAffiliation	Datatype	xsd:string
	ThridPartyContact	Datatype	xsd:string
	ThridPartyEmail	Datatype	xsd:string
	ThridPartyID	Datatype	xsd:int
	ThridPartyName	Datatype	xsd:string
Cost_Reduction	CostReductionAddedValue	Datatype	xsd:string
	CostReductionCurrentState	Datatype	xsd:string
	CostReductionKPI	Datatype	xsd:string
Eco_Friendliness	EcoFriendlinessAddedValue	Datatype	xsd:string
	EcoFriendlinessCurrentState	Datatype	xsd:string
	EcoFriendlinessKPI	Datatype	xsd:string
Performance_Improvement	PerformanceImprovementAddedValue	Datatype	xsd:string
	PerformanceImprovementCurrentState	Datatype	xsd:string
	PerformanceImprovementKPI	Datatype	xsd:string
Risk_Reduction	RiskReductionAddedValue	Datatype	xsd:string

	RiskReductionCurrentState	Datatype	xsd:string
	RiskReductionKPI	Datatype	xsd:string
Country	CountryCode	Datatype	xsd:string
	IncomeGroup	Datatype	xsd:string
	Region	Datatype	xsd:string
Call_Centre_Data	CallCentreLogTime	Datatype	xsd:dateTime
	CallCentreProblem	Datatype	xsd:string
PEID_Data	PEIDLogTime	Datatype	xsd:dateTime
	PEIDMeasuredValue	Datatype	xsd:string
Social_Media_Data	SocialMediaComment	Datatype	xsd:string
	SocialMediaLogTime	Datatype	xsd:dateTime
	SocialMediaSource	Datatype	xsd:string
	SocialMediaVote	Datatype	xsd:integer
	SocialMediaWishList	Datatype	xsd:string
Software_Log	SWLog	Datatype	xsd:string
	SWLogTime	Datatype	xsd:dateTime
Customer_Complaint_Entity	ComplaintAssessment	Datatype	xsd:string
	ComplaintComplexity	Datatype	xsd:string
	ComplaintCriticality	Datatype	xsd:string
	ComplaintReiteration	Datatype	xsd:string
Development_Need_Entity	DevelopmentNeedAssessment	Datatype	xsd:string
	DevelopmentNeedComplexity	Datatype	xsd:string
	DevelopmentNeedCriticality	Datatype	xsd:string
	DevelopmentNeedReiteration	Datatype	xsd:string
Disposal_Need_Entity	DisposalNeedAssessment	Datatype	xsd:string
	DisposalNeedComplexity	Datatype	xsd:string
	DisposalNeedCriticality	Datatype	xsd:string
	DisposalNeedReiteration	Datatype	xsd:string
Maintenance_Need_Entity	MaintenanceNeedAssessment	Datatype	xsd:string
	MaintenanceNeedComplexity	Datatype	xsd:string
	MaintenanceNeedCriticality	Datatype	xsd:string
	MaintenanceNeedReiteration	Datatype	xsd:string

Customer_Recommendation_Entity	RecommendationAssessment	Datatype	xsd:string
	RecommendationComplexity	Datatype	xsd:string
	RecommendationCriticality	Datatype	xsd:string
	RecommendationReiteration	Datatype	xsd:string
Recycling_Need_Entity	RecyclingNeedAssessment	Datatype	xsd:string
	RecyclingNeedComplexity	Datatype	xsd:string
	RecyclingNeedCriticality	Datatype	xsd:string
	RecyclingNeedReiteration	Datatype	xsd:string
Usage_Need_Entity	UsageNeedEntityAssessment	Datatype	xsd:string
	UsageNeedEntityComplexity	Datatype	xsd:string
	UsageNeedEntityCriticality	Datatype	xsd:string
	UsageNeedEntityReiteration	Datatype	xsd:string
PEID	PEIDID	Datatype	xsd:string
	PEIDName	Datatype	xsd:string
	PEIDtype	Datatype	xsd:string
	ProductComponentProperty	Datatype	xsd:string
	PartProperty	Datatype	xsd:string
Part	PartID	Datatype	xsd:int
	PartName	Datatype	xsd:string
	PartType	Datatype	xsd:string
Sub-assembly	Sub-assemblyID	Datatype	xsd:int
	Sub-assemblyName	Datatype	xsd:string
	Sub-assemblyType	Datatype	xsd:string
Product	ProductID	Datatype	xsd:int
	ProductName	Datatype	xsd:string
	ProductSerialNumber	Datatype	xsd:string
Error	CurrentState	Datatype	xsd:string
	DownTime	Datatype	xsd:dateTime
	ErrorLogtime	Datatype	xsd:dateTime
	ErrorSolution	Datatype	xsd:string
Normal	Performance	Datatype	xsd:string

	Quality	Datatype	xsd:string
Precaution	PrecautionCriticality	Datatype	xsd:string
	PrecautionType	Datatype	xsd:string
	ResidualLifeTime	Datatype	xsd:dateTime
Product_System	ProductSystemID	Datatype	xsd:int
	ProductSystemName	Datatype	xsd:string
Product_Service_System	PSSID	Datatype	xsd:int
	PSSName	Datatype	xsd:string
Application	ApplicationAvailability	Datatype	xsd:string
	ApplicationProductivity	Datatype	xsd:string
	ApplicationQuality	Datatype	xsd:string
Document	DocumentAvailability	Datatype	xsd:string
	DocumentProductivity	Datatype	xsd:string
	DocumentQuality	Datatype	xsd:string
Equipment	EquipmentAvailability	Datatype	xsd:string
	EquipmentProductivity	Datatype	xsd:string
	EquipmentQuality	Datatype	xsd:string
Raw_Material	RawMaterialAvailability	Datatype	xsd:string
	RawMaterialProductivity	Datatype	xsd:string
	RawMaterialQuality	Datatype	xsd:string
Development_Service	DevelopmentServiceID	Datatype	xsd:int
	DevelopmentServiceName	Datatype	xsd:string
	DevelopmentServiceStatus	Datatype	xsd:string
	DevelopmentServiceType	Datatype	xsd:string
Disposal_Service	DisposalServiceID	Datatype	xsd:int
	DisposalServiceName	Datatype	xsd:string
	DisposalServiceStatus	Datatype	xsd:string
	DisposalServiceType	Datatype	xsd:string
Maintenance_Service	MaintenanceServiceID	Datatype	xsd:int
	MaintenanceServiceName	Datatype	xsd:string
	MaintenanceServiceStatus	Datatype	xsd:string
	MaintenanceServiceType	Datatype	xsd:string

Recycling_Service	RecyclingServceType	Datatype	xsd:int
	RecyclingServiceID	Datatype	xsd:string
	RecyclingServiceName	Datatype	xsd:string
	RecyclingServiceStatus	Datatype	xsd:string
Usage_Service	UsageServiceID	Datatype	xsd:int
	UsageServiceName	Datatype	xsd:string
	UsageServiceStatus	Datatype	xsd:string
	UsageServiceType	Datatype	xsd:string
Software_Element	SoftwareID	Datatype	xsd:int
	SoftwareName	Datatype	xsd:string

2.2 White Goods Ontology

As part of the FALCON semantic framework, ontology for white goods business scenario has been developed to fulfil the functional requirements of the Arçelik use case. In this context, the main role of the FALCON VOP for white goods business case is to support the definition of products and services by gathering customers' feedback and data. Further details are described in the FALCON deliverables D5.1 and D5.2. The graph representation of White goods ontology is presented in Figure 6. In the following subsections: (i) Section 2.2.1 provides the list of classes, sub-classes and their description; (ii) Section 2.2.2 provides the list of object properties defining the relations between the classes; (iii) Section 2.2.3 provides the list of datatype properties; and (iv) Section 2.2.4 introducts White Goods Ontology mapping business stories.

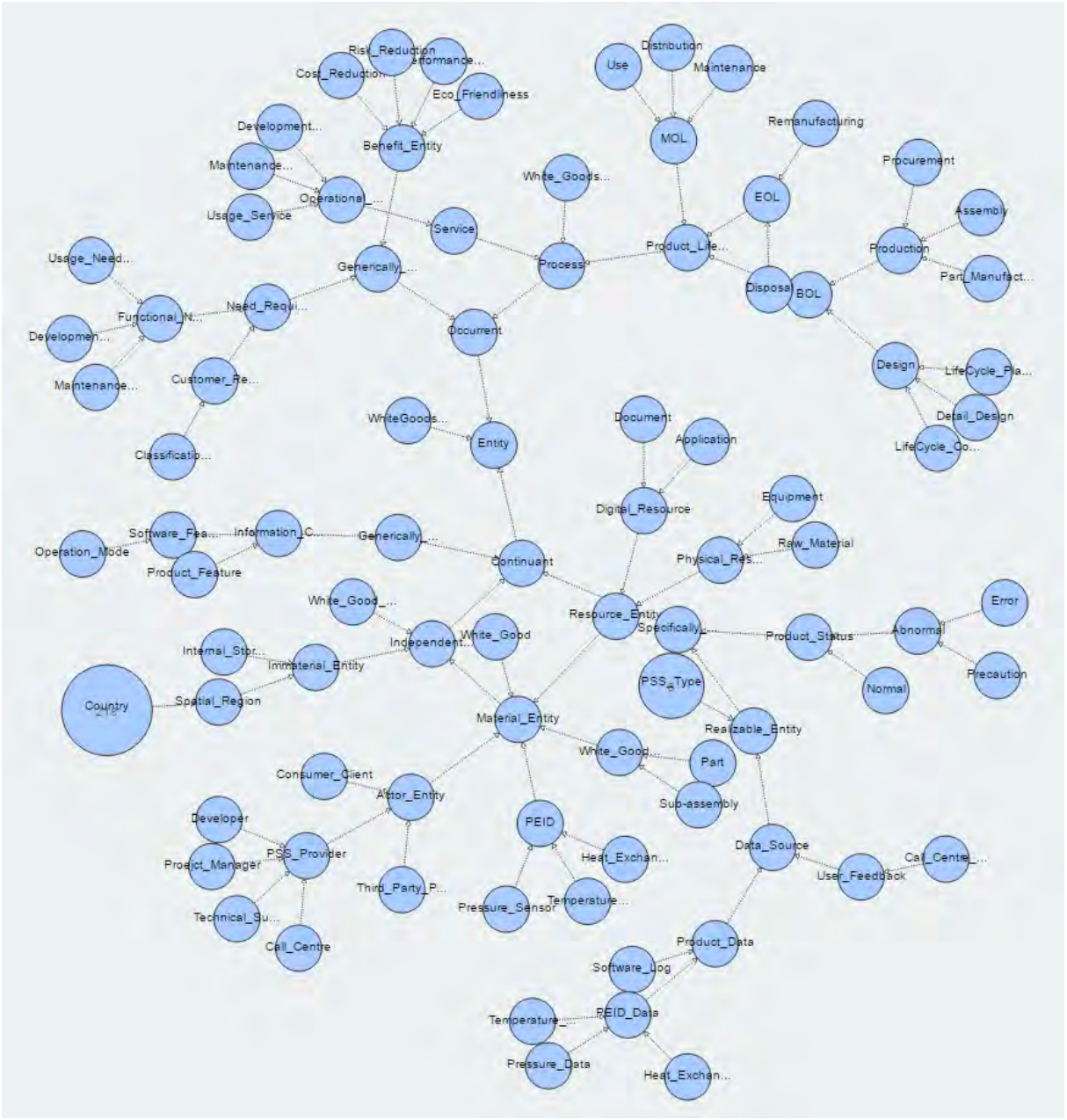

Figure 6. Graphic representation of White goods Products Ontology

2.2.1 Classes

Table 4 lists the upper ontology classes, their sub-classes and their description.

Table 4: White Goods Ontology classes

High Level Class	*Sub-class*	*Sub-class*	*Sub-class*	*Sub-class*	*Sub-class*	*Sub-class*	*Description*
WhiteGoods_PSS	-	-	-	-	-		
Continuant	Generically Dependent Continuant	Information Content Entity	Product Feature	-	-		
			Software Feature	Operation_Mode	-		
	Independent_Continuant	Immatrial Entity	Internal_Stored_Program	-	-		Choices of time, temperature etc that are left to the user are modelled as program of washing
			Spatial Region	Country	-		
		Material Entity	Actor Entity	Consumer Client	-		Generic concept created to group all involved parties
				PSS Provider	Customer Serivce		
					Data Analyst		
					Developer		
					Project Manager		
					Support Serivce		
				Thrid Party Partner	-		
			PEID	Pressure_Sensor	-		
				Heat_Exchanger_Sensor			

				Temperature_Sensor			
			White_Good	-	-		All the groups of White Goods
			White_Good_Component	Part	-		All the parts of the washing machine that might require maintenance or service
				Sub-assembly	-		
			Resource Entity	Digital Resource	Application		
					Document		
				Physical Resource	Equipment		
					Raw Material		
		White_Good_Product_System	-	-	-		
	Specifically Dependent Continuant	Product Status	Abnormal	Error	-		
				Precaution	-		
			Normal	-	-		
		Realizable Entity	Data_Source	Product Data	PEID Data	Pressure_Data	All information affecting the triggering of an Event are grouped as resource of information.
						Temperature_Data	
						Heat_Exchanger_Data	
					Software Log		
				User Feedback	Call Centre Data		
			PSS Type	-	-		
Occurrent	Process	Product Life Cycle	BOL	Design	Detail Design		
					LifeCycle Conceptual Design		
					LifeCycle Planning		
				Production	Assembly		
					Part Manufacturing		

					Procurement		
			EOL	Disposal	-		
				Remanufacturing	-		
			MOL	Distribution	-		
				Maintenance	-		
				Use	-		
		Service	Operational Element	Development Service	-		
				Disposal Service	-		
				Maintenance Service	-		
				Recycling Service	-		
				Usage Service	-		
		White_Goods_Service_System	-	-	-		
	Generically_Dependent_Occurrent	Benefit Entity	Cost Reduction	-	-		
			Eco Friendliness	-	-		
			Performance Improvement	-	-		
			Risk Reduction	-	-		
		Need Requirement Entity	Customer Requirement Entity	ClassificationOfProblem	-		
			Funtional Need Entity	Development Need Entity	-		
				Maintenance Need Entity	-		
				Usage Need Entity	-		

2.2.2 Object Properties

Table 5 lists the object properties of the upper ontology. For each property, the domain and range are defined.

Table 5: White Goods Ontology object properties

Relation	*Domain*	*Range*
hasLinkToOperationMode	Operation_Mode	Software_Log
relavantToSubassembly	ClassificationOfProblem	Sub-assembly
hasLinkToCallCentreData	ClassificationOfProblem	Call_Centre_Data
relavantToPart	ClassificationOfProblem	Part
belongsToComponent	PEID	White_Good_Component
belongsToPLC	White_Good	Product_Life_Cycle
excutesDevelopmentService	Developer	Development_Service
excutesMaintenanceSerivce	Support_Service	Maintenance_Service
excutesUsageService	Call_Centre	Usage_Service
generatesCallCentreData	Consumer_Client	Call_Centre_Data
generatesPEIDData	PEID	PEID_Data
generatesSoftwareLog	Internal_Stored_Program	Software_Log
hasBenefit	Service	Benefit_Entity
hasLinkToDevelopment	Development_Need_Entity	Development_Service
hasLinkToMaintenance	Maintenance_Need_Entity	Maintenance_Service
hasLinkToUsage	Usage_Need_Entity	Usage_Service
hasPart	White_Good	Part
hasProduct	White_Good_Product_System	White_Good
hasProductFeature	White_Good	Product_Feature
hasProductStatus	White_Good	Product_Status

hasProductSystem	WhiteGoods_PSS	White_Good_Product_System
hasPSSType	WhiteGoods_PSS	PSS_Type
hasService	White_Goods_Service_System	Service
hasServiceSystem	WhiteGoods_PSS	White_Goods_Service_System
hasSoftware	White_Good_Product_System	Internal_Stored_Program
hasSoftwareFeature	Internal_Stored_Program	Software_Feature
hasSubassembly	White_Good	Sub-assembly
isLocatedIn	White_Good	Country
isPlacedIn	Actor_Entity	Country
mentionsAboutDevelopment	Customer_Requirement_Entity	Development_Service
mentionsAboutMaintnenace	Customer_Requirement_Entity	Maintenance_Service
mentionsAboutUsage	Customer_Requirement_Entity	Usage_Service
refersToDevelopment	Data_Source	Development_Need_Entity
refersToMaintenance	Data_Source	Maintenance_Need_Entity
refersToPart	Abnormal	Part
refersToSubassembly	Abnormal	Sub-assembly
refersToUsage	Data_Source	Usage_Need_Entity
usesApplication	Service	Application
usesDocument	Service	Document
usesEquipment	Service	Equipment
usesRawMaterial	Service	Raw_Material

2.2.3 Datatype Properties

Table 6 lists the datatype properties of the upper ontology. For each property, the domain and range are defined.

Table 6: White Goods Ontology datatype properties

Domain	*Property*	*Type*	*Range*
ClassificationOfProblem	ProbleCode	Datatype	xsd:int
	Solution	Datatype	xsd:string
	TypeOfProblem	Datatype	xsd:string
Internal_Stored_Program	ProgramID	Datatype	xsd:int
	ProgramName	Datatype	xsd:string
Operation_Mode	OperationModeID	Datatype	xsd:int
	OperationModeName	Datatype	xsd:string
Consumer_Client	ClientAddress	Datatype	xsd:string
	ClientContact	Datatype	xsd:string
	ClientEmail	Datatype	xsd:string
	ClientID	Datatype	xsd:int
	ClientName	Datatype	xsd:string
	ClientWarranty	Datatype	xsd:int
Call_Centre	CustomerSerivceAccessAuthority	Datatype	xsd:string
	CustomerSerivceAddress	Datatype	xsd:string
	CustomerSerivceAffiliation	Datatype	xsd:string
	CustomerSerivceContact	Datatype	xsd:string
	CustomerSerivceEmail	Datatype	xsd:string
	CustomerSerivceID	Datatype	xsd:string
	CustomerSerivceName	Datatype	xsd:string
Developer	DeveloperAccessAuthority	Datatype	xsd:string
	DeveloperAddress	Datatype	xsd:string
	DeveloperAffiliation	Datatype	xsd:string
	DeveloperContact	Datatype	xsd:string
	DeveloperEmail	Datatype	xsd:string
	DeveloperID	Datatype	xsd:int

	DeveloperName	Datatype	xsd:string
Technical_Support_Service	ManagerAccessAuthority	Datatype	xsd:string
	ManagerAddress	Datatype	xsd:string
	ManagerAffiliation	Datatype	xsd:string
	ManagerContact	Datatype	xsd:string
	ManagerEmail	Datatype	xsd:string
	ManagerID	Datatype	xsd:int
	ManagerName	Datatype	xsd:string
Support_Service	SupportServiceAccessAuthority	Datatype	xsd:string
	SupportServiceAddress	Datatype	xsd:string
	SupportServiceAffiliation	Datatype	xsd:string
	SupportServiceContact	Datatype	xsd:string
	SupportServiceEmail	Datatype	xsd:string
	SupportServiceID	Datatype	xsd:int
	SupportServiceName	Datatype	xsd:string
Third_Party_Partner	ThridPartyAccessAuthority	Datatype	xsd:string
	ThridPartyAddress	Datatype	xsd:string
	ThridPartyAffiliation	Datatype	xsd:string
	ThridPartyContact	Datatype	xsd:string
	ThridPartyEmail	Datatype	xsd:string
	ThridPartyID	Datatype	xsd:int
	ThridPartyName	Datatype	xsd:string
Cost_Reduction	CostReductionAddedValue	Datatype	xsd:string
	CostReductionCurrentState	Datatype	xsd:string
	CostReductionKPI	Datatype	xsd:string
Eco_Friendliness	EcoFriendlinessAddedValue	Datatype	xsd:string
	EcoFriendlinessCurrentState	Datatype	xsd:string
	EcoFriendlinessKPI	Datatype	xsd:string
Performance_Improvement	PerformanceImprovementAddedValue	Datatype	xsd:string
	PerformanceImprovementCurrentState	Datatype	xsd:string
	PerformanceImprovementKPI	Datatype	xsd:string
Risk_Reduction	RiskReductionAddedValue	Datatype	xsd:string

	RiskReductionCurrentState	Datatype	xsd:string
	RiskReductionKPI	Datatype	xsd:string
Country	CountryCode	Datatype	xsd:string
	IncomeGroup	Datatype	xsd:string
	Region	Datatype	xsd:string
Call_Centre_Data	CallCentreLogTime	Datatype	xsd:dateTime
	CallCentreProblem	Datatype	xsd:string
PEID_Data	PEIDLogTime	Datatype	xsd:dateTime
	PEIDMeasuredValue	Datatype	xsd:string
Software_Log	SWLog	Datatype	xsd:string
	SWLogTime	Datatype	xsd:dateTime
Development_Need_Entity	DevelopmentNeedAssessment	Datatype	xsd:string
	DevelopmentNeedComplexity	Datatype	xsd:string
	DevelopmentNeedCriticality	Datatype	xsd:string
	DevelopmentNeedReiteration	Datatype	xsd:string
Maintenance_Need_Entity	MaintenanceNeedAssessment	Datatype	xsd:string
	MaintenanceNeedComplexity	Datatype	xsd:string
	MaintenanceNeedCriticality	Datatype	xsd:string
	MaintenanceNeedReiteration	Datatype	xsd:string
Usage_Need_Entity	UsageNeedEntityAssessment	Datatype	xsd:string
	UsageNeedEntityComplexity	Datatype	xsd:string
	UsageNeedEntityCriticality	Datatype	xsd:string
	UsageNeedEntityReiteration	Datatype	xsd:string
PEID	PEIDID	Datatype	xsd:string
	PEIDName	Datatype	xsd:string
	PEIDtype	Datatype	xsd:string
Part	PartID	Datatype	xsd:int
	PartName	Datatype	xsd:string
	PartType	Datatype	xsd:string
Sub-assembly	Sub-assemblyID	Datatype	xsd:int
	Sub-assemblyName	Datatype	xsd:string
	Sub-assemblyType	Datatype	xsd:string

White_Good	ProductID	Datatype	xsd:int
	ProductName	Datatype	xsd:string
	ProductSerialNumber	Datatype	xsd:string
Error	CurrentState	Datatype	xsd:string
	DownTime	Datatype	xsd:dateTime
	ErrorLogtime	Datatype	xsd:dateTime
	ErrorSolution	Datatype	xsd:string
Normal	Performance	Datatype	xsd:string
	Quality	Datatype	xsd:string
Precaution	PrecautionCriticality	Datatype	xsd:string
	PrecautionType	Datatype	xsd:string
	ResidualLifeTime	Datatype	xsd:dateTime
White_Good_Product_System	ProductSystemID	Datatype	xsd:int
	ProductSystemName	Datatype	xsd:string
WhiteGoods_PSS	PSSID	Datatype	xsd:int
	PSSName	Datatype	xsd:string
Application	ApplicationAvailability	Datatype	xsd:string
	ApplicationProductivity	Datatype	xsd:string
	ApplicationQuality	Datatype	xsd:string
Document	DocumentAvailability	Datatype	xsd:string
	DocumentProductivity	Datatype	xsd:string
	DocumentQuality	Datatype	xsd:string
Equipment	EquipmentAvailability	Datatype	xsd:string
	EquipmentProductivity	Datatype	xsd:string
	EquipmentQuality	Datatype	xsd:string
Raw_Material	RawMaterialAvailability	Datatype	xsd:string
	RawMaterialProductivity	Datatype	xsd:string
	RawMaterialQuality	Datatype	xsd:string
Development_Service	DevelopmentServiceID	Datatype	xsd:int
	DevelopmentServiceName	Datatype	xsd:string
	DevelopmentServiceStatus	Datatype	xsd:string
	DevelopmentServiceType	Datatype	xsd:string

Maintenance_Service	MaintenanceServiceID	Datatype	xsd:int
	MaintenanceServiceName	Datatype	xsd:string
	MaintenanceServiceStatus	Datatype	xsd:string
	MaintenanceServiceType	Datatype	xsd:string
Usage_Service	UsageServiceID	Datatype	xsd:int
	UsageServiceName	Datatype	xsd:string
	UsageServiceStatus	Datatype	xsd:string
	UsageServiceType	Datatype	xsd:string
Software_Element	SoftwareID	Datatype	xsd:int
	SoftwareName	Datatype	xsd:string

2.2.4 Ontology mapping business story

The current Deliverable 5.2 describing business stories to indentify and analize the the potential product-service featrues of Acrcelik White Goods scenario, and each business story has the list of questions. White Goods ontology can answer some of them. Therefore, this chapter introduces the mapping between White Goods ontology and the list of questions in business stories. The list of questions White Goods ontology maps are; (Q1**)** To get maintenance alerts from from sensor PUI; and (Q2**)** To analyze pressure sensor PUI to enable predictive maintenance.

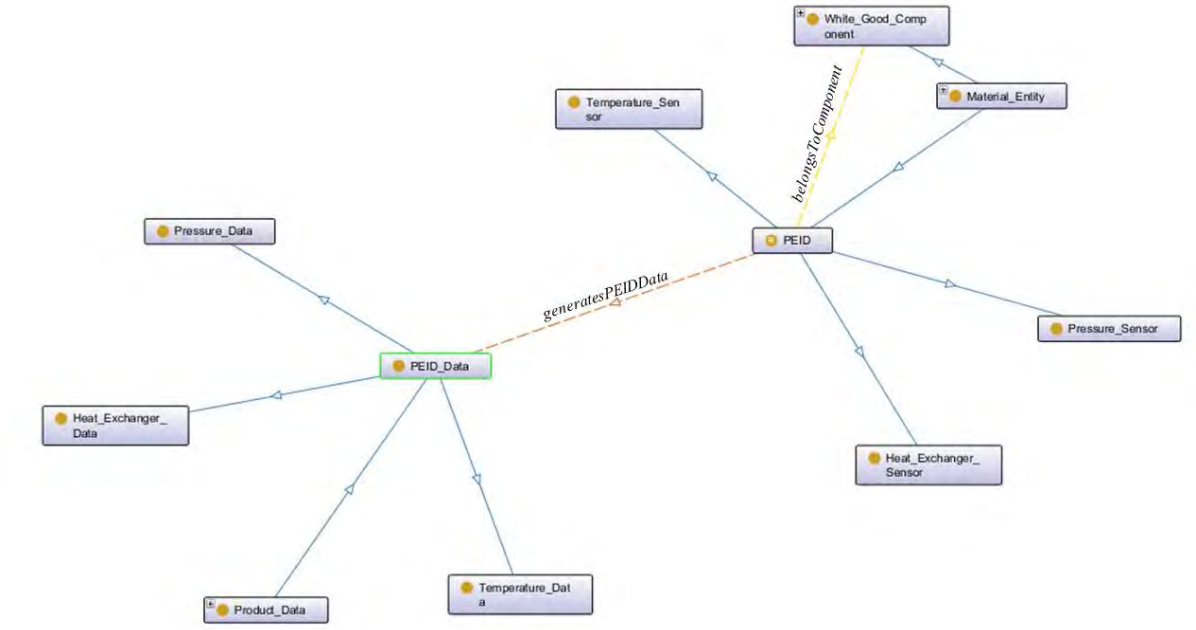

Figure 7. White Goods ontology mapping White Goods Business stories

To respond to two questions, White Goods ontology has class PEID and PEID_Data. PEID has sub class Pressure_Sensor, Heat_Exchanger_Sensor, and Temperature_Sensor. It describes that PEID *belongsTo* White_Goods Component, and PEID *generates* PEID_Data. In addition, PEID_Data has measuredValue as a datatype property. Therefore, when PEID data comes from the specific sensor, it facilitates to recognize what PEID generates PEID data and what component has this PEID. If technical support service opens PUI query manager, he/she can get measured values depending to his/her concept selection. After then, he/she will be able to get maintenance alerts or prediction values using the KCCM module or prediction advice widgets.

2.3 Brown Goods Ontology

As part of the FALCON semantic framework, ontology for Brown goods business scenario has been developed to fulfil the functional requirements of Arçelik use case. In this context, the main role of the FALCON VOP for Brown goods business case is to support the definition of products and services by gathering customers' feedback and data. Further details are described in the FALCON deliverables D5.1 and D5.2. The graph representation of Brown goods ontology is presented in Figure 7. In the following subsections: (i) Section 2.3.1 provides the list of classes, sub-classes and their description; (ii) Section 2.3.2 provides the list of object properties defining the relations between the classes; and (iii) Section 2.3.3 provides the list of datatype properties; and (iv) Section 2.3.4 introduces Brown Goods Ontology mapping business stories.

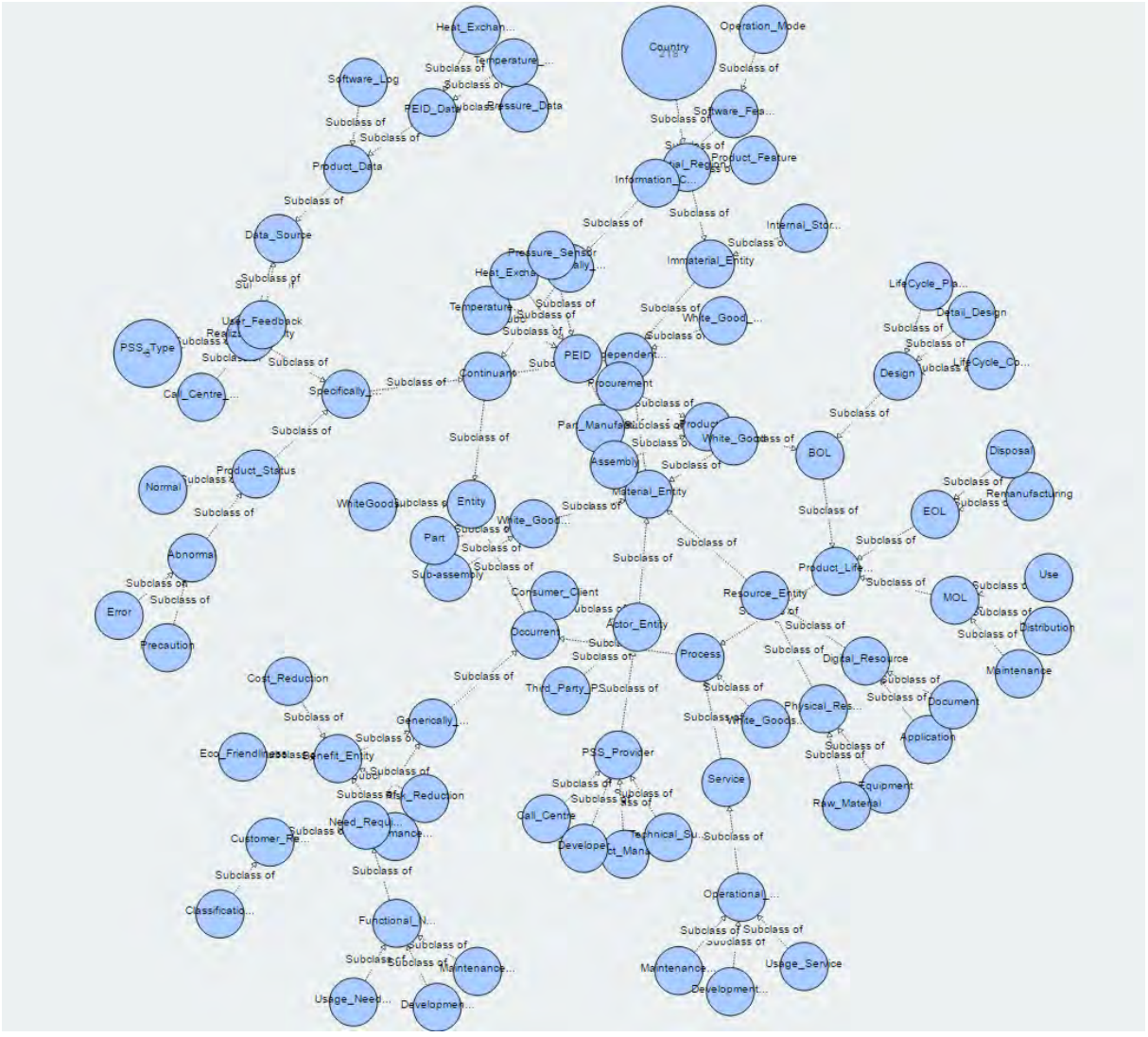

Figure 8. Graphic representation of Brown goods Products Ontology

2.3.1 Classes

Table 7 lists the upper ontology classes, their sub-classes and their description.

Table 7: Brown Goods Ontology classes

High Level Class	*Sub-class*	*Sub-class*	*Sub-class*	*Sub-class*	*Sub-class*	*Description*
Product Service System	-	-	-	-	-	
Continuant	Generically Dependent Continuant	Information Content Entity	Product Feature	TV_Series	-	
			Software Feature	Command_Actions	-	All the commands available command actions
				Application_Category	-	All the categories of Applications
	Independent Countinuant	Immaterial Entity	Software Element	Appstore		All the Appstores that are available for smart TVs
				TV_Application	-	All the applications that are available through the application store.
			Spatial Region	City		
				Country	-	
		Material Entity	Actor Entity	Consumer Client	-	Is the owner and/or user of Arçeliksmart

						TV. Although he is not the user of the FALCON VOP, his behavior is an important source of information
				PSS Provider	Call_Centre	Call Centre operators are responsible for collecting the data about smart TV malfunctioning or assistance that customers required. Through key-words labelling of each call centre interaction with customer statistics of smart TV usage will be produced as a valuable data source
					Product&Project_Management_Office	FALCON users receive information through the VOP in two different modes, the first being responsesto initiated queries about specific application usage and second being automatic alerts when statistics in certain application usage exceedspredefined thresholds.

					Software_Development_Department	FALCON users from this group will have insight into activities and conclusions of Product & Project Management Department in order to prevent issues reappearing in new releases
					Test&Verification_Department	FALCON users from this group will be able to access Customer usage data analysis results in order to get better insight into critical and weak points of previous application releases in order to make the testing procedure more efficient.
				Third Party Partner	-	
			Smart_TV	-	-	The Smart TV
			TV_Component	Part	-	
				Sub-assembly	-	
			Resource Entity	Digital Resource	Application	
					Document	
				Physical Resource	Equipment	
					Raw Material	
		BrownGoods_Product_System	-	-	-	

	Specifically_Dependent_Continuant	Realizable_Entity	Data_Source	Product_Data	TV Usage Data	Usage of applications is followed through a number of downloads and interaction. It is defined in a form of log files for each customer.
				User Feedback	Call Centre Data	Calls directed to the call centre will be combined with multiple-choice questionnaires that will create meta-data for this unstructured data source. Meta-data are modelled through this concept.

			PSS Type	-	-	
Occurrent	Process	Product Life Cycle	BOL	Design	Detail Design	
					LifeCycle Conceptual Design	
					LifeCycle Planning	
				Production	Assembly	
					Part Manufacturing	
					Procurement	
			EOL	Disposal	-	
				Remanufacturing	-	
			MOL	Distribution	-	
				Maintenance	-	
				Use	-	
		Service	Operational Element	Development Service	-	
				Disposal Service	-	
				Maintenance Service	-	
				Recycling Service	-	
				Usage Service	-	
		Service System	-	-	-	
	Generically_Dependent_Occurrent	Benefit Entity	Cost Reduction	-	-	
			Eco Friendliness	-	-	
			Performance Improvement	-	-	
			Risk Reduction	-	-	

		Need Requirement Entity	ClassificationOfApplicationReason	-	-	All the Classifications that are the reason of Userfeedback
	Temporal_Entity	Application_Usage				All the records of Application usage

2.3.2 Object Properties

Table 8 lists the object properties of the upper ontology. For each property, the domain and range are defined.

Table 8: Brown Goods Ontology object properties

Relation	*Domain*	*Range*
belongsToPLC	Smart_TV	Product_Life_Cycle
excutesDevelopmentService	Software_Development_Department	Development_Service
generatesCallCentreData	Consumer_Client	Call_Centre_Data
generatesSoftwareLog	Software_Element	Software_Log
hasApplicationReason	Call_Centre_Data	ClassificationOfApplicationReason
hasAppStore	Smart_TV	Appstore
hasBenefit	Service	Benefit_Entity
hasCategory	TV_Application	Application_Category
hasCity	Country	City
hasCommandAction	TV_Application	Command_Actions
hasPart	Smart_TV	Part

hasProduct	BrownGoods_Product_System	Smart_TV
hasProductFeature	Smart_TV	Product_Feature
hasProductSystem	BrownGoodsPSS	BrownGoods_Product_System
hasPSSType	BrownGoodsPSS	PSS_Type
hasReason	Software_Log	ClassificationOfApplicationReason
hasService	BrownGoods_Serivce_System	Service
hasServiceSystem	BrownGoodsPSS	BrownGoods_Serivce_System
hasSoftware	BrownGoods_Product_System	Software_Element
hasSoftwareFeature	Software_Element	Software_Feature
hasSubassembly	Smart_TV	Sub-assembly
hasTVApplication	Smart_TV	TV_Application
hasTVSeries	Smart_TV	TV_Series
hasUsage	TV_Application	Application_Usage
includeTVApplication	Appstore	TV_Application
isLocatedIn	Smart_TV	Country
isLocatedInCity	Smart_TV	City
isLocatedInCountry	Smart_TV	Country
isPlacedIn	Actor_Entity	Country
isPlacedInCity	Actor_Entity	City
isPlacedInCountry	Actor_Entity	Country
isRelatedToCity	Application_Usage	City
isRelatedToCountry	Application_Usage	Country
recordsUsage	recordsUsage	Application_Usage
refersToCity	Call_Centre_Data	City
refersToCountry	Call_Centre_Data	Country

refersToTVApplication	Data_Source	TV_Application
usesApplication	Service	Application
usesDocument	Service	Document
usesEquipment	Service	Equipment
usesRawMaterial	Service	Raw_Material

2.3.3 Datatype Properties

Table 9 lists the datatype properties of the upper ontology. For each property, the domain and range are defined.

Table 9: Brown Goods Ontology datatype properties

Domain	*Property*	*Type*	*Range*
Application	ApplicationAvailability	Datatype	xsd:string
	ApplicationProductivity	Datatype	xsd:string
	ApplicationQuality	Datatype	xsd:string
Application_Category	CategoryID	Datatype	xsd:int
	CategoryName	Datatype	xsd:string
Application_Usage	UsageStart	Datatype	xsd:dateTime
	UsageEnd	Datatype	xsd:dateTime
	UsageLengthSec	Datatype	xsd:int
Appstore	AppstoreID	Datatype	xsd:int
	AppstoreName	Datatype	xsd:string
BrownGoods_Product_System	ProductSystemID	Datatype	xsd:int
	ProductSystemName	Datatype	xsd:string
BrownGoodsPSS	PSSID	Datatype	xsd:int
	PSSName	Datatype	xsd:string
Call_Centre_Data	CallCentreLogTime	Datatype	xsd:dateTime
	CallCentreProblem	Datatype	xsd:string
City	CityID	Datatype	xsd:int
	CityName	Datatype	xsd:string

Command_Actions	CommandID	Datatype	xsd:int
	CommandName	Datatype	xsd:string
Consumer_Client	ClientAddress	Datatype	xsd:string
	ClientContact	Datatype	xsd:string
	ClientEmail	Datatype	xsd:string
	ClientID	Datatype	xsd:int
	ClientName	Datatype	xsd:string
	ClientWarranty	Datatype	xsd:int
Cost_Reduction	CostReductionAddedValue	Datatype	xsd:string
	CostReductionCurrentState	Datatype	xsd:string
	CostReductionKPI	Datatype	xsd:string
Country	CountryCode	Datatype	xsd:string
	IncomeGroup	Datatype	xsd:string
	Region	Datatype	xsd:string
CustomerServiceProperty	CustomerSerivceAccessAuthority	Datatype	xsd:string
	CustomerSerivceAddress	Datatype	xsd:string
	CustomerSerivceAffiliation	Datatype	xsd:string
	CustomerSerivceContact	Datatype	xsd:string
	CustomerSerivceEmail	Datatype	xsd:string
	CustomerSerivceID	Datatype	xsd:int
	CustomerSerivceName	Datatype	xsd:string
Development_Service	DevelopmentServiceID	Datatype	xsd:string
	DevelopmentServiceName	Datatype	xsd:string
	DevelopmentServiceStatus	Datatype	xsd:string
	DevelopmentServiceType	Datatype	xsd:string
Document	DocumentAvailability	Datatype	xsd:string
	DocumentProductivity	Datatype	xsd:string
	DocumentQuality	Datatype	xsd:string
Eco_Friendliness	EcoFriendlinessAddedValue	Datatype	xsd:string
	EcoFriendlinessCurrentState	Datatype	xsd:string
	EcoFriendlinessKPI	Datatype	xsd:string
Equipment	EquipmentAvailability	Datatype	xsd:string

	EquipmentProductivity	Datatype	xsd:string
	EquipmentQuality	Datatype	xsd:string
Part	PartID	Datatype	xsd:int
	PartName	Datatype	xsd:string
	PartType	Datatype	xsd:string
Performance_Improvement	PerformanceImprovementAddedValue	Datatype	xsd:string
	PerformanceImprovementCurrentState	Datatype	xsd:string
	PerformanceImprovementKPI	Datatype	xsd:string
Product&Project_Management_Office	ManagerAccessAuthority	Datatype	xsd:string
	ManagerAddress	Datatype	xsd:string
	ManagerAffiliation	Datatype	xsd:string
	ManagerContact	Datatype	xsd:string
	ManagerEmail	Datatype	xsd:string
	ManagerID	Datatype	xsd:int
	ManagerName	Datatype	xsd:string
Raw_Material	RawMaterialAvailability	Datatype	xsd:string
	RawMaterialProductivity	Datatype	xsd:string
	RawMaterialQuality	Datatype	xsd:string
Risk_Reduction	RiskReductionAddedValue	Datatype	xsd:string
	RiskReductionCurrentState	Datatype	xsd:string
	RiskReductionKPI	Datatype	xsd:string
Smart_TV	Brand	Datatype	xsd:string
	ProductCode	Datatype	xsd:int
	ProductID	Datatype	xsd:string
	ProductName	Datatype	xsd:string
	ProductSerialNumber	Datatype	xsd:string
	TVName	Datatype	xsd:string
Software_Development_Department	DeveloperAccessAuthority	Datatype	xsd:string
	DeveloperAddress	Datatype	xsd:string
	DeveloperAffiliation	Datatype	xsd:string
	DeveloperContact	Datatype	xsd:string
	DeveloperEmail	Datatype	xsd:string

	DeveloperID	Datatype	xsd:int
	DeveloperName	Datatype	xsd:string
Software_Element	SoftwareID	Datatype	xsd:int
	SoftwareName	Datatype	xsd:string
Software_Log	SWLog	Datatype	xsd:string
	SWLogTime	Datatype	xsd:dateTime
Sub-assembly	Sub-assemblyID	Datatype	xsd:int
	Sub-assemblyName	Datatype	xsd:string
	Sub-assemblyType	Datatype	xsd:string
Test&Verification_Department	SupportServiceAccessAuthority	Datatype	xsd:string
	SupportServiceAddress	Datatype	xsd:string
	SupportServiceAffiliation	Datatype	xsd:string
	SupportServiceContact	Datatype	xsd:string
	SupportServiceEmail	Datatype	xsd:string
	SupportServiceID	Datatype	xsd:int
	SupportServiceName	Datatype	xsd:string
Third_Party_Partner	ThridPartyAccessAuthority	Datatype	xsd:string
	ThridPartyAddress	Datatype	xsd:string
	ThridPartyAffiliation	Datatype	xsd:string
	ThridPartyContact	Datatype	xsd:string
	ThridPartyEmail	Datatype	xsd:string
	ThridPartyID	Datatype	xsd:int
	ThridPartyName	Datatype	xsd:string
TV_Series	TVCode	Datatype	xsd:string
	TVSeriesID	Datatype	xsd:int
	TVSeriesName	Datatype	xsd:string

2.3.4 Ontology mapping business story

The current Deliverable 5.2 describing business stories to identify and analyze the the potential product-service features of Arçelik Brown Goods scenario, and each business story has the list of questions. Brown Goods ontology can answer some of them. Therefore, this chapter introduces the mapping between Brown Goods ontology and the list of questions in business stories. The list of questions Brown Goods ontology maps are; (Q1_1) To analyse the usage frequency of the connected application "children`s TV" app; and (Q1_2) To analyse the usage length/duration the connected application "chidren`s TV" app.

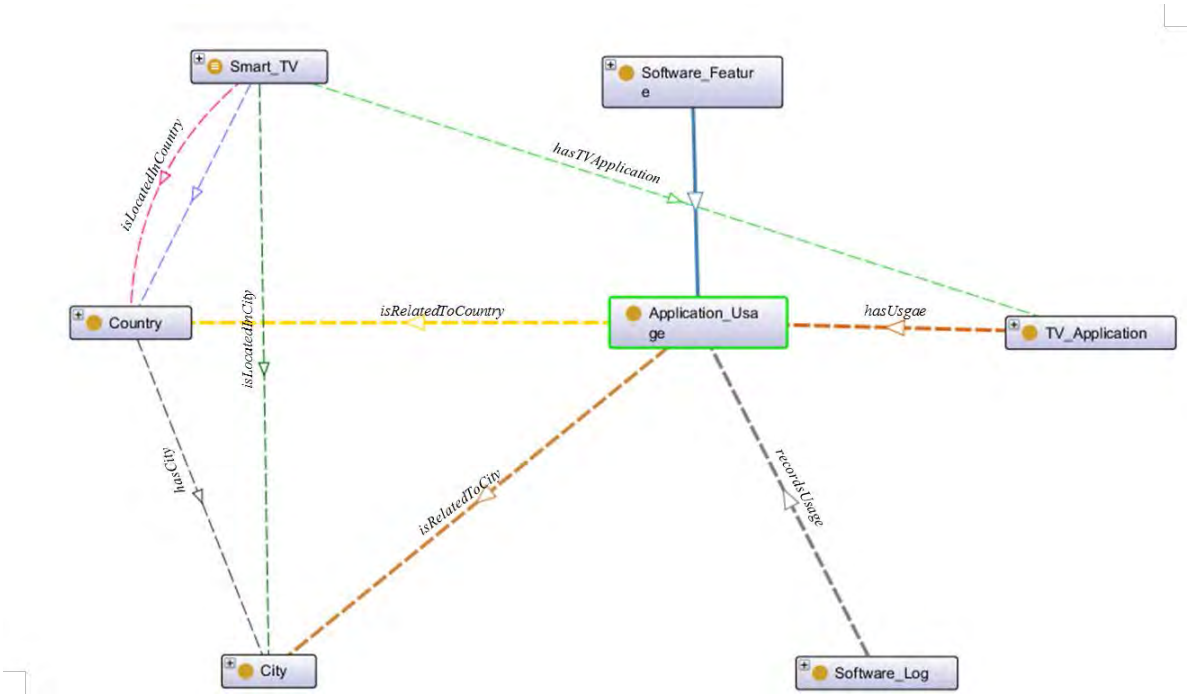

Figure 9. Brown Goods ontology mapping Brown Goods Business stories

To respond to two questions, Brown Goods ontology has class Smart_TV, TV_Application, Country, City,and Software_Log. It describes that Application_Usage *isRelatedTo* Country and City, and TV_Application *has* Application_Usage. In addition, Application_Usage has UsageStart, UsageEnd, and UsageLength as datatypeproperties. Therefore, when Application_Usage instances are recorded, they will have relations with TV_Application, and Regions. If a Software Designer opens PUI Query manager, he/she can get start/end point of usage, and usage length. After then, he/she will be able to visualize this information using Data Visualization module.

2.4 Healthcare Products Ontology

As part of the FALCON semantic framework, ontology for Healthcare business scenario has been developed to fulfil the functional requirements of PHILIPS use case. The main role of the FALCON VOP for Healthcare business case is to support the definition of products and services by gathering customers' feedback and data. Further details are described in the FALCON deliverables D6.1 and D6.2. The graph representation of Healthcare ontology is presented in Figure 8. In the following subsections: (i) Section 2.4.1 provides the list of classes, sub-classes and their description; (ii) Section 2.4.2 provides the list of object properties defining the relations between the classes; and (iii) Section 2.4.3 provides the list of datatype properties, and (iv) Section 2.4.4 introduces Healthcare products Ontology mapping business stories.

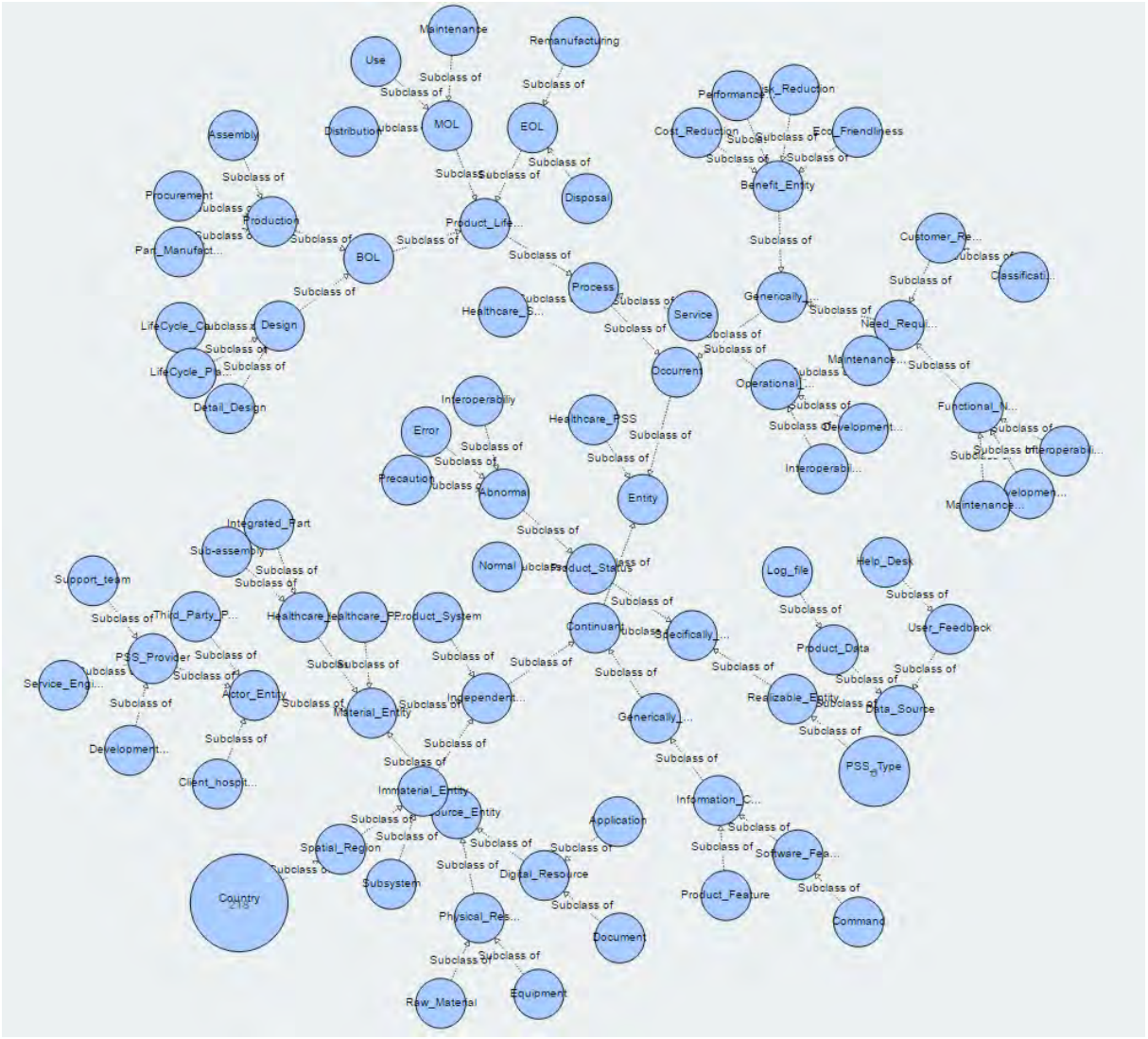

Figure 10. Graphic representation of Healthcare Products Ontology

2.4.1 Classes

Table 10 lists the upper ontology classes, their sub-classes and their description.

Table 10: Healthcare Products Ontology classes

High Level Class	*Sub-class*	*Sub-class*	*Sub-class*	*Sub-class*	*Sub-class*	*Description*
Healthcare_PSS	-	-	-	-	-	
Continuant	Generically Dependent Continuant	Information Content Entity	Product Feature	-	-	
			Software Feature	Command	-	
	Independent_Continuant	Immatrial Entity	Subsystem	-	-	
			Spatial Region	Country	-	
		Material Entity	Actor Entity	Client_hospital_customer	-	Clients are owners of Phillips health care products. Although they are not direct users of FALCON platform, their behaviour is one relevant data source and FALCONs' functionalities' outputs are targeted to modifying clients behaviour
				PSS Provider	Support_team	Based on log files and information coming from help desk, the support team will have all the relevant information about previous and

						potential issues before getting back to the client and thus leading to increased product-service value
					Development_team	The development team will gain insight into statistically the most frequent issues with product functionalities or installation problems, based on which they will adjust the future generations of products or software updates for the existing ones.
					Service_Engineer	Service engineers are team who travel to clients facility to perform scheduled maintenance and service activities.
				Thrid Party Partner	-	
			Healthcare_Product	-	-	Phillips healthcare equipment: ‘medical image acquisition modality, also known as ‘scanner.
			Healthcare_Product_ Component	Integrated_Part	-	Group of parts that are considered as relevant when

						analyzing interoperability or interactionrelated issues.
				Sub-assembly	-	Group of Subassembly that are considered as relevant when analyzing interoperability or interactionrelated issues.
			Resource Entity	Digital Resource	Application	
					Document	
				Physical Resource	Equipment	
					Raw Material	
		Product System	-	-	-	
	Specifically Dependent Continuant	Product Status	Abnormal	Error	-	
				Interoperabiliy		
				Precaution	-	
			Normal	-	-	
					Software Log	Log files contain the history of usage modes and operations
				User Feedback	Help_Desk_Data	The help desk uses a labelling system to add semantic structure to phone call data and email free text data

					Social Media Data	
			PSS Type	-	-	
Occurrent	Process	Product Life Cycle	BOL	Design	Detail Design	
					LifeCycle Conceptual Design	
					LifeCycle Planning	
				Production	Assembly	
					Part Manufacturing	
					Procurement	
			EOL	Disposal	-	
				Remanufacturing	-	
			MOL	Distribution	-	
				Maintenance	-	
				Use	-	
		Service	Operational Element	Development Service	-	
				Maintenance Service	-	
				Interoperability_Recommendation_Service	-	
		Service System	-	-	-	
	Generically_Dependent_Occurrent	Benefit Entity	Cost Reduction	-	-	
			Eco Friendliness	-	-	

			Performance Improvement	-	-	
			Risk Reduction	-	-	
		Need Requirement Entity	Customer Requirement Entity	Classification_of_Customer_Feedback	-	
			Funtional Need Entity	Development Need Entity	-	
				Maintenance Need Entity	-	
				Interoperabiliy_Analysis_Entity	-	
	Temporal_Entity					

2.4.2 Object Properties

Table 11 lists the object properties of the upper ontology. For each property, the domain and range are defined.

Table 11: Healthcare Products Ontology object properties

Relation	*Domain*	*Range*
belongsToPLC	Healthcare_Product	Product_Life_Cycle
excutesDevelopmentService	Development_team	Development_Service
excutesMaintenanceSerivce	Support_team	Maintenance_Service
excutesUsageService	Service_Engineer	Interoperability_Recommendation_Service
generatesCallCentreData	Client_hospital_customer	Help_Desk
generatesSoftwareLog	Subsystem	Log_file
hasBenefit	Service	Benefit_Entity
hasCommand	Log_file	Command

hasNextCommand	Log_file	Command
hasLinkToCallCentreData	Classification_of_Customer_Feedback	Help_Desk
hasLinkToDevelopment	Development_Need_Entity	Development_Service
hasLinkToMaintenance	Maintenance_Need_Entity	Maintenance_Service
hasLinkToUsage	Interoperabiliy_Analysis_Entity	Interoperability_Recommendation_Service
hasPart	Healthcare_Product	Integrated_Part
hasProduct	Healthcare_Product_System	Healthcare_Product
hasProductFeature	Healthcare_Product	Product_Feature
hasProductStatus	Healthcare_Product	Product_Status
hasProductSystem	Healthcare_PSS	Healthcare_Product_System
hasPSSType	Healthcare_PSS	PSS_Type
hasService	Healthcare_Service_System	Service
hasServiceSystem	Healthcare_PSS	Healthcare_Service_System
hasSoftware	Healthcare_Product_System	Subsystem
hasSoftwareFeature	Subsystem	Software_Feature
hasSubassembly	Healthcare_Product	Sub-assembly
isLocatedIn	Healthcare_Product	Country
isPlacedIn	Actor_Entity	Country
mentionsAboutDevelopment	Customer_Requirement_Entity	Development_Service
mentionsAboutMaintnenace	Customer_Requirement_Entity	Maintenance_Service
mentionsAboutUsage	Customer_Requirement_Entity	Interoperability_Recommendation_Service
recordsEndUserAction	Log_file	EndUserAction
refersToDevelopment	Data_Source	Development_Need_Entity
refersToMaintenance	Data_Source	Maintenance_Need_Entity
refersToPart	Abnormal	Integrated_Part

refersToSubassembly	Abnormal	Sub-assembly
refersToUsage	Data_Source	Interoperabiliy_Analysis_Entity
relevantToPart	Classification_of_Customer_Feedback	Integrated_Part
relevantToSubassembly	Classification_of_Customer_Feedback	Sub-assembly
usesApplication	Service	Application
usesDocument	Service	Document
usesEquipment	Service	Equipment
usesRawMaterial	Service	Raw_Material

2.4.3 Datatype Properties

Table 12 lists the datatype properties of the upper ontology. For each property, the domain and range are defined.

Table 12: Healthcare Products Ontology datatype properties

Domain	*Property*	*Type*	*Range*
Classification_of_Customer_Feedback	TypeOfProblem	Datatype	xsd:string
	ProblemCode	Datatype	xsd:string
	Solution	Datatype	xsd:string
Client_hospital_customer	ClientAddress	Datatype	xsd:string
	ClientContact	Datatype	xsd:string
	ClientEmail	Datatype	xsd:string
	ClientID	Datatype	xsd:int
	ClientName	Datatype	xsd:string
	ClientWarranty	Datatype	xsd:int
EndUserAction	CurrentAction	Datatype	xsd:string
	PreviousAction	Datatype	xsd:string

	ActionStart	Datatype	xsd:dateTime
	ActionEnd	Datatype	xsd:dateTime
Service_Engineer	CustomerSerivceAccessAuthority	Datatype	xsd:string
	CustomerSerivceAddress	Datatype	xsd:string
	CustomerSerivceAffiliation	Datatype	xsd:string
	CustomerSerivceContact	Datatype	xsd:string
	CustomerSerivceEmail	Datatype	xsd:string
	CustomerSerivceID	Datatype	xsd:string
	CustomerSerivceName	Datatype	xsd:string
Development_team	DeveloperAccessAuthority	Datatype	xsd:string
	DeveloperAddress	Datatype	xsd:string
	DeveloperAffiliation	Datatype	xsd:string
	DeveloperContact	Datatype	xsd:string
	DeveloperEmail	Datatype	xsd:string
	DeveloperID	Datatype	xsd:int
	DeveloperName	Datatype	xsd:string
Support_team	SupportServiceAccessAuthority	Datatype	xsd:string
	SupportServiceAddress	Datatype	xsd:string
	SupportServiceAffiliation	Datatype	xsd:string
	SupportServiceContact	Datatype	xsd:string
	SupportServiceEmail	Datatype	xsd:string
	SupportServiceID	Datatype	xsd:int
	SupportServiceName	Datatype	xsd:string
Third_Party_Partner	ThridPartyAccessAuthority	Datatype	xsd:string
	ThridPartyAddress	Datatype	xsd:string

	ThridPartyAffiliation	Datatype	xsd:string
	ThridPartyContact	Datatype	xsd:string
	ThridPartyEmail	Datatype	xsd:string
	ThridPartyID	Datatype	xsd:int
	ThridPartyName	Datatype	xsd:string
Cost_Reduction	CostReductionAddedValue	Datatype	xsd:string
	CostReductionCurrentState	Datatype	xsd:string
	CostReductionKPI	Datatype	xsd:string
Eco_Friendliness	EcoFriendlinessAddedValue	Datatype	xsd:string
	EcoFriendlinessCurrentState	Datatype	xsd:string
	EcoFriendlinessKPI	Datatype	xsd:string
Performance_Improvement	PerformanceImprovementAddedValue	Datatype	xsd:string
	PerformanceImprovementCurrentState	Datatype	xsd:string
	PerformanceImprovementKPI	Datatype	xsd:string
Risk_Reduction	RiskReductionAddedValue	Datatype	xsd:string
	RiskReductionCurrentState	Datatype	xsd:string
	RiskReductionKPI	Datatype	xsd:string
Country	CountryCode	Datatype	xsd:string
	IncomeGroup	Datatype	xsd:string
	Region	Datatype	xsd:string
Help_Desk	CallCentreLogTime	Datatype	xsd:dateTime
	CallCentreProblem	Datatype	xsd:string
	RepairTextJob	Datatype	xsd:string
	OperationalStartDate	Datatype	xsd:dateTime
	DescriptionPart	Datatype	xsd:string

	CustomerRepairTextJob	Datatype	xsd:string
	CustomerComplintCall	Datatype	xsd:string
	CMHours	Datatype	xsd:integer
	CMHourCosts	Datatype	xsd:integer
	CallOpenDateCall	Datatype	xsd:dateTime
	CallIDCall	Datatype	xsd:integer
Log_file	SWLog	Datatype	xsd:string
	MeasuredValue	Datatype	xsd:float
	SWLogTime	Datatype	xsd:dateTime
Development_Need_Entity	DevelopmentNeedAssessment	Datatype	xsd:string
	DevelopmentNeedComplexity	Datatype	xsd:string
	DevelopmentNeedCriticality	Datatype	xsd:string
	DevelopmentNeedReiteration	Datatype	xsd:string
Maintenance_Need_Entity	MaintenanceNeedAssessment	Datatype	xsd:string
	MaintenanceNeedComplexity	Datatype	xsd:string
	MaintenanceNeedCriticality	Datatype	xsd:string
	MaintenanceNeedReiteration	Datatype	xsd:string
Usage_Need_Entity	UsageNeedEntityAssessment	Datatype	xsd:string
	UsageNeedEntityComplexity	Datatype	xsd:string
	UsageNeedEntityCriticality	Datatype	xsd:string
	UsageNeedEntityReiteration	Datatype	xsd:string
Integrated_Part	PartID	Datatype	xsd:int
	PartName	Datatype	xsd:string
	PartCosts	Datatype	xsd:integer
	PartType	Datatype	xsd:string

Sub-assembly	Sub-assemblyID	Datatype	xsd:int
	Sub-assemblyName	Datatype	xsd:string
	SubassemblyCost	Datatype	xsd:integer
	Sub-assemblyType	Datatype	xsd:string
Product	ProductID	Datatype	xsd:int
	ProductName	Datatype	xsd:string
	ProductSerialNumber	Datatype	xsd:string
Error	CurrentState	Datatype	xsd:string
	DownTime	Datatype	xsd:dateTime
	ErrorLogtime	Datatype	xsd:dateTime
	ErrorSolution	Datatype	xsd:string
Normal	Performance	Datatype	xsd:string
	Quality	Datatype	xsd:string
Precaution	PrecautionCriticality	Datatype	xsd:string
	PrecautionType	Datatype	xsd:string
	ResidualLifeTime	Datatype	xsd:dateTime
Healthcare_Product_System	ProductSystemID	Datatype	xsd:int
	ProductSystemName	Datatype	xsd:string
Healthcare_PSS	PSSID	Datatype	xsd:int
	PSSName	Datatype	xsd:string
Application	ApplicationAvailability	Datatype	xsd:string
	ApplicationProductivity	Datatype	xsd:string
	ApplicationQuality	Datatype	xsd:string
Document	DocumentAvailability	Datatype	xsd:string
	DocumentProductivity	Datatype	xsd:string

	DocumentQuality	Datatype	xsd:string
Equipment	EquipmentAvailability	Datatype	xsd:string
	EquipmentProductivity	Datatype	xsd:string
	EquipmentQuality	Datatype	xsd:string
Raw_Material	RawMaterialAvailability	Datatype	xsd:string
	RawMaterialProductivity	Datatype	xsd:string
	RawMaterialQuality	Datatype	xsd:string
Development_Service	DevelopmentServiceID	Datatype	xsd:int
	DevelopmentServiceName	Datatype	xsd:string
	DevelopmentServiceStatus	Datatype	xsd:string
	DevelopmentServiceType	Datatype	xsd:string
Maintenance_Service	MaintenanceServiceID	Datatype	xsd:int
	MaintenanceServiceName	Datatype	xsd:string
	MaintenanceServiceStatus	Datatype	xsd:string
	MaintenanceServiceType	Datatype	xsd:string
Interoperability_Recommendation_Service	UsageServiceID	Datatype	xsd:int
	UsageServiceName	Datatype	xsd:string
	UsageServiceStatus	Datatype	xsd:string
	UsageServiceType	Datatype	xsd:string
Subsystem	SoftwareID	Datatype	xsd:int
	SoftwareName	Datatype	xsd:string

2.4.4 Ontology mapping business story

The current Deliverable 6.2 describing business stories to identify and analyze the the potential product-service features of Healthcare Products scenario, and each business story has the list of questions. Healthcare Products ontology can answer some of them. Therefore, this chapter introduces the mapping between Healthcare Products ontology and the list of questions in business stories. The list of questions Healthcare Products ontology maps are; (Q1) Find recurring sequences of end-user actions actions specified (queried) by the VOP user; and (Q3/4) Find the most/least frequently used end-user actions.

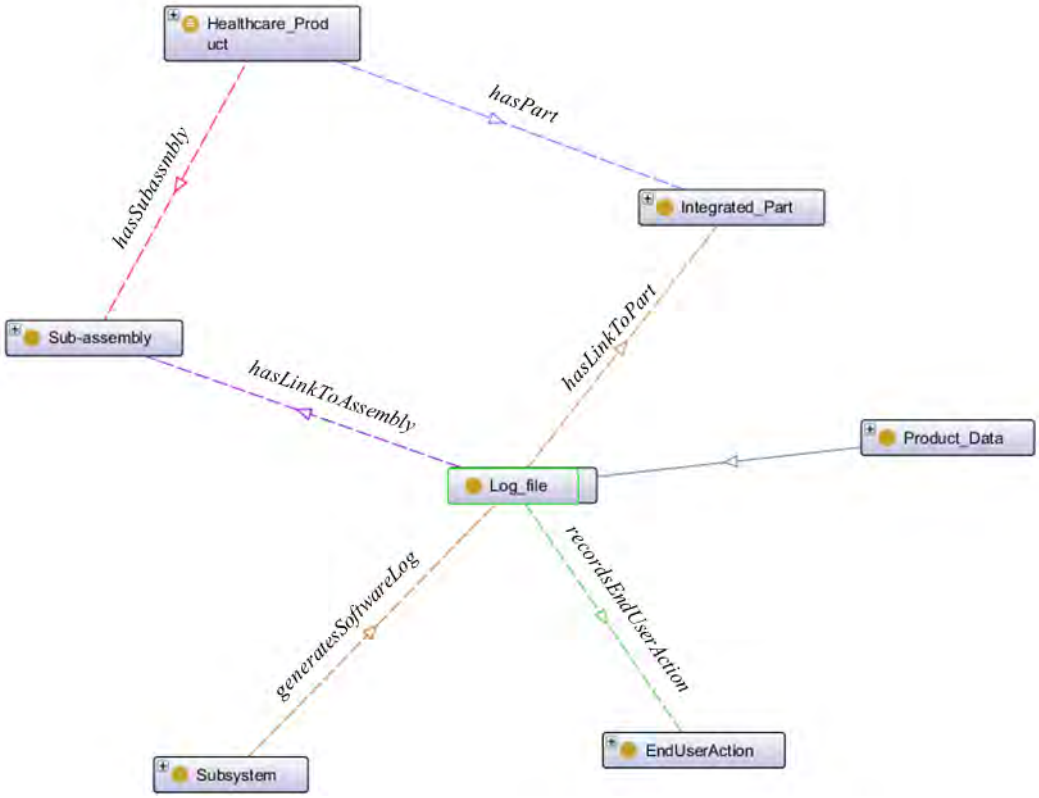

Figure 11. Healthcare Products ontology mapping Healthcare Products Business stories

To respond to three questions, Healthcare Products ontology has class EndUserAction, Log_file, Healthcare_Product and classes representing components of Healthcare. It describes Log_file *records* EndUserAction, and Log_file *haslinkTo* classes of components. Therefore, when Log_file instance including EndUserAction comes, it will have a link with Healthcare product components. In addition, EndUserAction has Start/End point of Acntion, CurrentAction, and Previous Action as datatypeproperties. If a Product Manager opens PUI Quert manager, he/she can get the sequence of actions or frequency of end-user actions. After then, he/she will be able to this information using Data Visualization module.

2.5 Clothing Textiles Ontology

As part of the FALCON semantic framework, ontology for the textile business scenario has been developed to fulfil functional requirements of Dena user-case. The main role of the FALCON VOP for textile business case is to support the definition of products and services through the customers' feedback and data gathering. Further details are described in the FALCON deliverables D7.1 and D7.2. The graph representation of Clothing textiles ontology is presented in Figure 9. In the following subsections: (i) Section 2.5.1 provides the list of classes, sub-classes and their description; (ii) Section 2.5.2 provides the list of object properties defining the relations between the classes; and (iii) Section 2.5.3 provides the list of datatype properties, and (iv) Section 2.5.4 introduces Clothing Textiles Ontology mapping business stories.

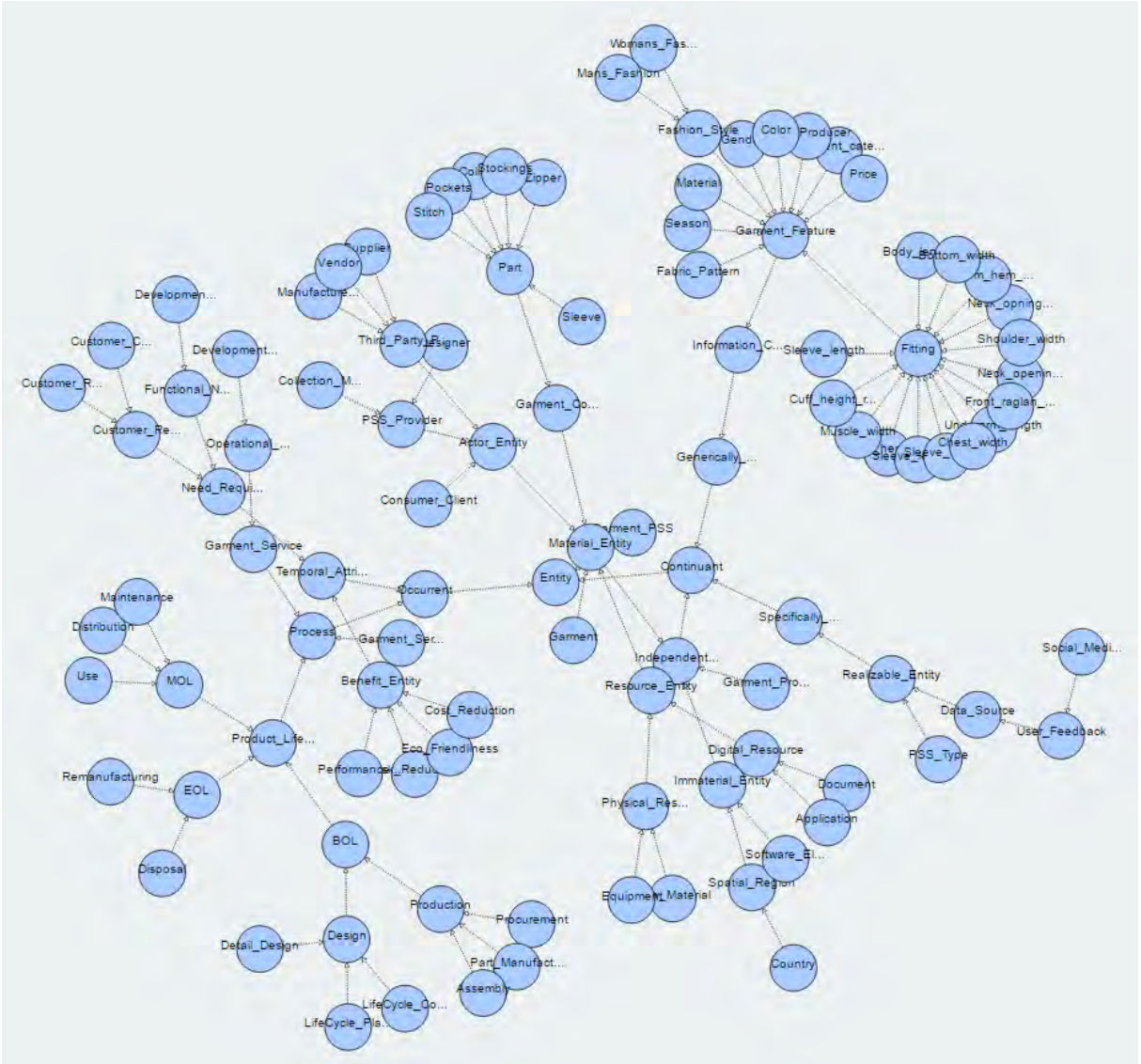

Figure 12. Graphic representation of Clothing textiles Ontology

2.5.1 Classes

Table 13 lists the upper ontology classes, their sub-classes and their description.

Table 13: Clongthing textiles Products Ontology classes

High Level Class	*Sub-class*	*Sub-class*	*Sub-class*	*Sub-class*	*Sub-class*	*Description*
Garment_PSS	-	-	-	-	-	
Continuant	Generically_Dependent_Continuant	Information_Content_Entity	Garment_Feature	BrandProducer	-	as described on the product label or description
				Color	-	LAB od Pantone, calibrated when possible on the basis of light and device
				Fabric_Pattern		Groups of fabric patterns
				Fashion_Style	Mans_Fashion	Groups of fabric categories
					Womans_Fashion	
				Fitting	Body_length	on the basis of tightness parameters on the body
					Bottom_hem_height_rib_1x1_double	

					Bottom_width	
					Chest_width	
					Cuff_height_rib_1x1_double	
					Front_raglan_length	
					Muscle_width	
					Neck_hem_height_rib_1x1_double	
					Neck_opening_width	
					Neck_opning_height	
					Shoulder_width	
					Sleeve_length	
					Sleeve_length_from_centre_collar	
					Sleeve_width_at_10cm_from_the_bottom	
					Underarm_length	
				Garment_category	-	general categories used by international custom and trade agreements

				Gender	-	male/female/unisex
				Material	-	general categories used by international custom and trade agreements
				Price	-	when available accompanied by currency definition (Euro, Usd, etc…)
				Season	-	the 2 main ones of clothing production (F/W or S/S)
	Independent_Continuant	Garment_Product_System	-	-	-	
			Spatial_Region	Country	-	
		Material_Entity	Actor_Entity	Consumer_Client	-	
				PSS_Provider	Collection_Manager	Generic concept created to group all involved parties.
					Designer	
				Third_Party_Partner	Manufacturers_Machine_Programmer	

					Supplier	
					Vendor	
			Garment	-	-	
			Garment_Component	Collar	-	All parts that are treated as design questions. List of sub-concepts defined here is not definitive and can be extended as required.
				Pockets	-	
				Sleeve	-	
				Stitch	-	
				Stockings	-	
				Zipper	-	
			Resource_Entity	Digital_Resource	Application	
					Document	
				Physical_Resource	Equipment	
					Raw_Material	
	Specifically_Dependent_Continuant	Realizable_Entity	Data_Source	User_Feedback	Social_Media_Data	Giving all the labels and keywords that can be assigned to a product or a part, specific social media are monitored for these words occurrences and statistical processing will result in recommendation of currently interesting topics to the Collection Manager.

		PSS_Type	-	-	-	
Occurrent	Process	Garment_Service	Operational_Element	Development_Service	-	
		Garment_Service_System			-	
		Product_Life_Cycle	BOL	Design	Detail_Design	
					LifeCycle_Conceptual_Design	
					LifeCycle_Planning	
				Production	Assembly	
					Part_Manufacturing	
					Procurement	
			EOL	Disposal	-	
				Remanufacturing	-	
			MOL	Distribution	-	
				Maintenance	-	
				Use	-	
	Generically_Dependent_Occurrent	Benefit_Entity	Cost_Reduction	-	-	
			Eco_Friendliness	-	-	
			Performance_Improvement	-	-	

			Risk_Reduction	-	-	
		Need_Requirement_Entity	Customer_Requirement_Entity	Customer_Complaint_Entity	-	
				Customer_Recommendation_Entity	-	
			Functional_Need_Entity	Development_Need_Entity	-	

2.5.2 Object Properties

Table 14 lists the object properties of the upper ontology. For each property, the domain and range are defined.

Table 14: Clongthing textiles Products Ontology object properties

Relation	*Domain*	*Range*
belongsToPLC	Garment	Product_Life_Cycle
excutesDevelopmentService	Designer	Development_Service
generatesSocialMediaData	Consumer_Client	Social_Media_Data
hasBenefit	Garment_Service	Benefit_Entity
hasBodyLength	Garment	Body_length
hasBottomHemHeight	Garment	Bottom_hem_height_rib_1x1_double
hasBottomWidth	Garment	Bottom_width
hasBrandProducer	Garment	BrandProducer
hasChestWidth	Garment	Chest_width
hasColor	Garment	Color
hasCuffHeight	Garment	Cuff_height_rib_1x1_double

hasFabricPattern	Garment	Fabric_Pattern
hasFrontRaglanLength	Garment	Front_raglan_length
hasGarmentCategory	Garment	Garment_category
hasGender	Garment	Gender
hasLinkToDevelopment	Development_Need_Entity	Development_Service
hasMansFashion	Garment	Mans_Fashion
hasMaterial	Garment	Material
hasMuscleWidth	Garment	Muscle_width
hasNeckHemHeight	Garment	Neck_hem_height_rib_1x1_double
hasNeckOpeningWidth	Garment	Neck_opening_width
hasNeckOpningHeight	Garment	Neck_opning_height
hasPart	Garment	Part
hasPrice	Garment	Price
hasProduct	Garment_Product_System	Garment
hasProductSystem	Garment_PSS	Garment_Product_System
hasPSSType	Garment_PSS	PSS_Type
hasSeason	Garment	Season
hasService	Garment_Service_System	Garment_Service
hasServiceSystem	Garment_PSS	Garment_Service_System
hasShoulderWidth	Garment	Shoulder_width
hasSleeveLength	Garment	Sleeve_length
hasSleeveLengthCentreCollar	Garment	Sleeve_length_from_centre_collar
hasSleeveWidth	Garment	Sleeve_width_at_10cm_from_the_bottom
hasUnderarmLength	Garment	Underarm_length
hasWomansFashion	Garment	Womans_Fashion
isLocatedIn	Garment	Country
isPlacedIn	Actor_Entity	Country
mentionsAboutDevelopment	Customer_Requirement_Entity	Development_Service
refersToBodyLength	Social_Media_Data	Body_length
refersToBottomHemHeight	Social_Media_Data	Bottom_hem_height_rib_1x1_double
refersToBottomWidth	Social_Media_Data	Bottom_width
refersToBrandProducer	Social_Media_Data	BrandProducer

refersToChestWidth	Social_Media_Data	Chest_width
refersToColor	Social_Media_Data	Color
refersToComplaint	User_Feedback	Customer_Complaint_Entity
refersToCuffHeight	Social_Media_Data	Cuff_height_rib_1x1_double
refersToDevelopment	Data_Source	Development_Need_Entity
refersToFabricPattern	Social_Media_Data	Fabric_Pattern
refersToFrontRaglanLength	Social_Media_Data	Front_raglan_length
refersToGarmentCategory	Social_Media_Data	Garment_category
refersToGender	Social_Media_Data	Gender
refersToMansFashion	Social_Media_Data	Mans_Fashion
refersToMaterial	Social_Media_Data	Material
refersToMuscleWidth	Social_Media_Data	Muscle_width
refersToNeckHemHeight	Social_Media_Data	Neck_hem_height_rib_1x1_double
refersToNeckOpeningWidth	Social_Media_Data	Neck_opening_width
refersToNeckOpningHeight	Social_Media_Data	Neck_opning_height
refersToPrice	Social_Media_Data	Price
refersToRecommendation	User_Feedback	Customer_Recommendation_Entity
refersToSeanson	Social_Media_Data	Season
refersToShoulderWidth	Social_Media_Data	Shoulder_width
refersToSleeveLength	Social_Media_Data	Sleeve_length
refersToSleeveLengthCentreCollar	Social_Media_Data	Sleeve_length_from_centre_collar
refersToSleeveWidth	Social_Media_Data	Sleeve_width_at_10cm_from_the_bottom
refersToUnderarmLength	Social_Media_Data	Underarm_length
refersToWomansFashion	Social_Media_Data	Womans_Fashion
referToGarment	Social_Media_Data	Garment
usesApplication	Garment_Service	Application
usesDocument	Garment_Service	Document
usesEquipment	Garment_Service	Equipment
usesRawMaterial	Garment_Service	Raw_Material

2.5.3 Datatype Properties

Table 15 lists the datatype properties of the upper ontology. For each property, the domain and range are defined.

Table 15: Clongthing textiles Ontology datatype properties

Domain	*Property*	*Type*	*Range*
Consumer_Client	ClientAddress	Datatype	xsd:string
	ClientContact	Datatype	xsd:string
	ClientEmail	Datatype	xsd:string
	ClientID	Datatype	xsd:int
	ClientName	Datatype	xsd:string
	ClientWarranty	Datatype	xsd:int
Designer	DeveloperAccessAuthority	Datatype	xsd:string
	DeveloperAddress	Datatype	xsd:string
	DeveloperAffiliation	Datatype	xsd:string
	DeveloperContact	Datatype	xsd:string
	DeveloperEmail	Datatype	xsd:string
	DeveloperID	Datatype	xsd:string
	DeveloperName	Datatype	xsd:string
Collection_Manager	ManagerAccessAuthority	Datatype	xsd:string
	ManagerAddress	Datatype	xsd:string
	ManagerAffiliation	Datatype	xsd:string
	ManagerContact	Datatype	xsd:string
	ManagerEmail	Datatype	xsd:string
	ManagerID	Datatype	xsd:int
	ManagerName	Datatype	xsd:string
Third_Party_Partner	ThridPartyAccessAuthority	Datatype	xsd:string
	ThridPartyAddress	Datatype	xsd:string
	ThridPartyAffiliation	Datatype	xsd:string
	ThridPartyContact	Datatype	xsd:string
	ThridPartyEmail	Datatype	xsd:string
	ThridPartyID	Datatype	xsd:int

	ThridPartyName	Datatype	xsd:string
Cost_Reduction	CostReduction	Datatype	xsd:string
	CostReductionAddedValue	Datatype	xsd:string
	CostReductionCurrentState	Datatype	xsd:string
	CostReductionKPI	Datatype	xsd:string
Eco_Friendliness	EcoFriendlinessAddedValue	Datatype	xsd:string
	EcoFriendlinessCurrentState	Datatype	xsd:string
	EcoFriendlinessKPI	Datatype	xsd:string
Performance_Improvement	PerformanceImprovementAddedValue	Datatype	xsd:string
	PerformanceImprovementCurrentState	Datatype	xsd:string
	PerformanceImprovementKPI	Datatype	xsd:string
Risk_Reduction	RiskReductionAddedValue	Datatype	xsd:string
	RiskReductionCurrentState	Datatype	xsd:string
	RiskReductionKPI	Datatype	xsd:string
Color	ColorName	Datatype	xsd:string
	HEX	Datatype	xsd:string
	HSL_Light_Percentage	Datatype	xsd:int
	HSL_Satur_Percentage	Datatype	xsd:int
	HSV_Light_Percentage	Datatype	xsd:int
	HSV_Satur_Percentage	Datatype	xsd:int
	HUE_Degree	Datatype	xsd:int
	RGB_Blue_Percentage	Datatype	xsd:int
	RGB_Green_Percentage	Datatype	xsd:int
	RGB_Red_Percentage	Datatype	xsd:int
Country	CountryCode	Datatype	xsd:string
	IncomeGroup	Datatype	xsd:string
	Region	Datatype	xsd:string
Social_Media_Data	SocialMediaComment	Datatype	xsd:string
	SocialMediaSource	Datatype	xsd:string
	SocialMediaVote	Datatype	xsd:string
	SocialMediaWishList	Datatype	xsd:string
Garment	Boutique	Datatype	xsd:string

	CODICE_ARTICOLO	Datatype	xsd:string
	NOME_ARTICOLO	Datatype	xsd:string
Customer_Complaint_Entity	ComplaintAssessment	Datatype	xsd:string
	ComplaintComplexity	Datatype	xsd:string
	ComplaintCriticality	Datatype	xsd:string
	ComplaintReiteration	Datatype	xsd:string
Development_Need_Entity	DevelopmentNeedAssessment	Datatype	xsd:string
	DevelopmentNeedComplexity	Datatype	xsd:string
	DevelopmentNeedCriticality	Datatype	xsd:string
	DevelopmentNeedReiteration	Datatype	xsd:string
Customer_Recommendation_Entity	RecommendationAssessment	Datatype	xsd:string
	RecommendationComplexity	Datatype	xsd:string
	RecommendationCriticality	Datatype	xsd:string
	RecommendationReiteration	Datatype	xsd:string
Part	PartID	Datatype	xsd:int
	PartName	Datatype	xsd:string
	PartType	Datatype	xsd:string
Garment	ProductID	Datatype	xsd:int
	ProductName	Datatype	xsd:string
	ProductSerialNumber	Datatype	xsd:int
Garment_Product_System	ProductSystemID	Datatype	xsd:int
	ProductSystemName	Datatype	xsd:string
Garment_PSS	PSSID	Datatype	xsd:int
	PSSName	Datatype	xsd:string
Application	ApplicationAvailability	Datatype	xsd:string
	ApplicationProductivity	Datatype	xsd:string
	ApplicationQuality	Datatype	xsd:string
Document	DocumentAvailability	Datatype	xsd:string
	DocumentProductivity	Datatype	xsd:string
	DocumentQuality	Datatype	xsd:string
Equipment	EquipmentAvailability	Datatype	xsd:string
	EquipmentProductivity	Datatype	xsd:string

	EquipmentQuality	Datatype	xsd:string
Raw_Material	RawMaterialAvailability	Datatype	xsd:string
	RawMaterialProductivity	Datatype	xsd:string
	RawMaterialQuality	Datatype	xsd:string
Development_Service	DevelopmentServiceID	Datatype	xsd:int
	DevelopmentServiceName	Datatype	xsd:string
	DevelopmentServiceStatus	Datatype	xsd:string
	DevelopmentServiceType	Datatype	xsd:string

2.5.4 Ontology mapping business story

The current Deliverable 7.2 describing business stories to indentify and analize the the potential product-service featrues of Clothing Textiles scenario, and each business story has the list of questions. Clothing Textiles ontology can answer some of them. Therefore, this chapter introduces the mapping between Clothing Textiles ontology and the list of questions in business stories. The list of questions Clothing Textiles ontology maps are: Q1: Identify texture trends for next season's polo shirts on social media (Facebook, Instagram, Lookbook), Q2: Analyse feedback from a fan forum about ideas for an upcoming collection.

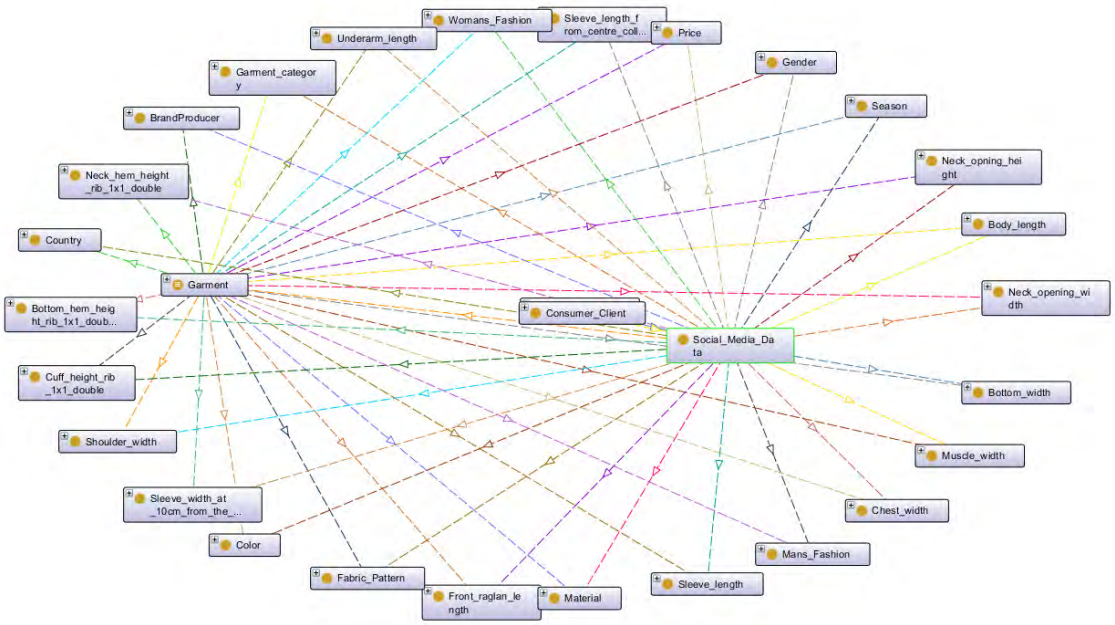

Figure 13. Clothing Textiles ontology mapping Clothing Textiles Business stories

To respond to two questions, Clothing Textiles ontology has class Social_Media_Data, Garment, and Garment_Feature. It describes that Social_Media_Data *refersTo* Garment and Garment_Feature. Thereforen, when Social_Media_Data comes from the Consumer_Client, it facilitates to recognize what Garment_Feature the Social_Media_Data instances refers to. After then, Social_Media_Data instances can be classified based upon Garment_Feature. Further on, extraction classificated data from PUI can be visualized using data visualisation Module.

2.6 High-tech products Ontology

As part of the FALCON semantic framework, ontology for High-tech Products business scenario has been developed to fulfil functional requirements of DATAPIXEL user-case. The main requirement of the FALCON VOP for textile business case is to support the definition of products and services through customers' feedback and data gathering and further details are described in the FALCON deliverables D8.1 and D8.2. The graph representation of High-tech products ontology is presented in Figure 10. Accordingly, Section 2.6.1 provides the list of classes, sub-classes and their description, Section 2.6.2 provides the list of object properties defining the relations between the classes, and Section 2.6.3 provides the list of datatype properties.

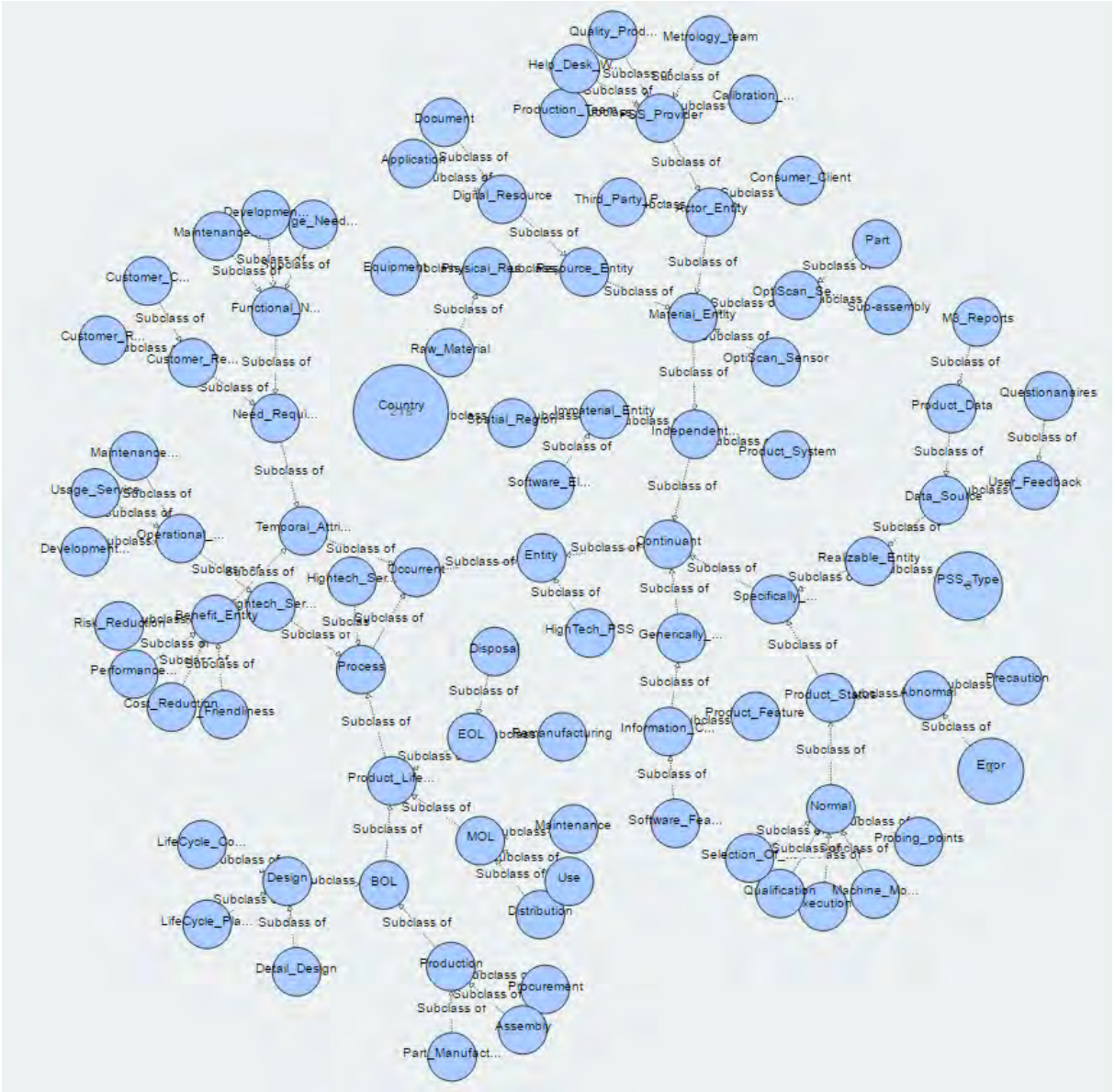

Figure 14. Graphic representation of Hightech Products Ontology

2.6.1 Classes

Table 16 lists the upper ontology classes, their sub-classes and their description.

Table 16: High-tech products Ontology classes

High Level Class	*Sub-class*	*Sub-class*	*Sub-class*	*Sub-class*	*Sub-class*	*Description*
HighTech_PSS	-	-	-	-	-	
Continuant	Generically Dependent Continuant	Information Content Entity	Product Feature	-	-	Groups of OptiScan Sensor
			Software Feature	-	-	Groups of M3 software features
	Independent_Continuant	Immatrial Entity	Software Element	-	-	Groups of M3 software
			Spatial Region	Country	-	
		Material Entity	Actor Entity	Consumer Client	-	
				PSS Provider	Calibration_and_Maintenance_Team	The team is responsible for defining calibration and maintenance schedule according to specific scanning program and installation conditions
					Help_Desk_Worker	The help Desk workers in DATAPIXEL are responsible for collecting information from clients about their issues and inquiries

					Metrology_team	The metrology team at DATPIXEL is responsible for managing projects, updates and upgrades offers based on user experience
					Production_Team	Production workers at the clients facility are the ones actually handling the devices
					Quality_Product_Department	The quality Product Department at the clients facility is monitoring and reporting issues with product/service system
				Thrid Party Partner	-	
			OptiScan_Sensor	-	-	
			OptiScan_Sensor_Component	Part	-	
				Sub-assembly	-	
			Resource Entity	Digital Resource	Application	
					Document	
				Physical Resource	Equipment	
					Raw Material	
		Product System	-	-	-	
	Specifically Dependent Continuant	Product Status	Abnormal	Error	-	
				Precaution	-	
			Normal	Execution	-	

				Machine_Movements	-	
				Probing_points	-	
				Qualification	-	
				Selection_Of_Part	-	
		Realizable_Entity	Product_Data	Product_Data	M3_Reports	M3 reports contain the history of usage of every metrology system as well as conditions in which it operated. Based on these data streams the module for recommending calibration or maintenance & verification is designed.
				User Feedback	Questionanaires	Each call to the help desk will be tagged with caller, topic, equipment in question, source of problem etc. Based on these, Datapixel can make decisions about future improvements of the metrology system. In the same time, user profiles will be created to contribute to better informed responses of

						technical support team to new issues.
			PSS Type	-	-	
Occurrent	Process	Product Life Cycle	BOL	Design	Detail Design	
					LifeCycle Conceptual Design	
					LifeCycle Planning	
				Production	Assembly	
					Part Manufacturing	
					Procurement	
			EOL	Disposal	-	
				Remanufacturing	-	
			MOL	Distribution	-	
				Maintenance	-	
				Use	-	
		Hightech_Service	Operational Element	Development Service	-	
				Disposal Service	-	
				Maintenance Service	-	
				Recycling Service	-	
				Usage Service	-	
		Hightech_Service_System	-	-	-	
		Benefit Entity	Cost Reduction	-	-	

	Generically_Dependent_Occurrent		Eco Friendliness	-	-	
			Performance Improvement	-	-	
			Risk Reduction	-	-	
		Need Requirement Entity	Customer Requirement Entity	Customer Complaint Entity	-	
				Customer Recommendation Entity	-	
			Funtional Need Entity	Development Need Entity	-	
				Maintenance Need Entity	-	
				Usage Need Entity	-	

2.6.2 Object Properties

Table 17 lists the object properties of the upper ontology. For each property, the domain and range are defined.

Table 17: High-tech products Ontology object properties

Relation	*Domain*	*Range*
belongsToComponent	PEID	Product_Component
belongsToPLC	Product	Product_Life_Cycle
excutesDevelopmentService	Developer	Development_Service
excutesDisposalService	Customer_Service	Disposal_Service
excutesMaintenanceSerivce	Support_Service	Maintenance_Service
excutesRecyclingService	Customer_Service	Recycling_Service
excutesUsageService	Customer_Service	Usage_Service

generatesCallCentreData	Consumer_Client	Call_Centre_Data
generatesPEIDData	PEID	PEID_Data
generatesSocialMediaData	Consumer_Client	Social_Media_Data
generatesSoftwareLog	Software_Element	Software_Log
hasBenefit	Service	Benefit_Entity
hasLinkToDevelopment	Development_Need_Entity	Development_Service
hasLinkToDisposal	Disposal_Need_Entity	Disposal_Service
hasLinkToMaintenance	Maintenance_Need_Entity	Maintenance_Service
hasLinkToRecycling	Recycling_Need_Entity	Recycling_Service
hasLinkToUsage	Usage_Need_Entity	Usage_Service
hasPart	Product	Part
hasProduct	Product_System	Product
hasProductFeature	Product	Product_Feature
hasProductStatus	Product	Product_Status
hasProductSystem	Product_Service_System	Product_System
hasPSSType	Product_Service_System	PSS_Type
hasService	Service_System	Service
hasServiceSystem	Product_Service_System	Service_System
hasSoftware	Product_System	Software_Element
hasSoftwareFeature	Software_Element	Software_Feature
hasSubassembly	Product	Sub-assembly
isLocatedIn	Product	Country
isPlacedIn	Actor_Entity	Country
mentionsAboutDevelopment	Customer_Requirement_Entity	Development_Service
mentionsAboutDisposal	Customer_Requirement_Entity	Disposal_Service
mentionsAboutMaintnenace	Customer_Requirement_Entity	Maintenance_Service
mentionsAboutRecycling	Customer_Requirement_Entity	Recycling_Service
mentionsAboutUsage	Customer_Requirement_Entity	Usage_Service
providesComplaintResult	Data_Analyst	Customer_Complaint_Entity
providesDevelopmentResult	Data_Analyst	Development_Need_Entity
providesDisposalResult	Data_Analyst	Disposal_Need_Entity
providesMaintenanceResult	Data_Analyst	Maintenance_Need_Entity

providesRecommendationResult	Data_Analyst	Customer_Recommendation_Entity
ProvidesRecyclingResult	Data_Analyst	Recycling_Need_Entity
providesUsageResult	Data_Analyst	Usage_Need_Entity
refersToComplaint	User_Feedback	Customer_Complaint_Entity
refersToDevelopment	Data_Source	Development_Need_Entity
refersToDisposal	Data_Source	Disposal_Need_Entity
refersToMaintenance	Data_Source	Maintenance_Need_Entity
refersToPart	Abnormal	Part
refersToRecommendation	User_Feedback	Customer_Recommendation_Entity
refersToRecycling	Data_Source	Recycling_Need_Entity
refersToSubassembly	Abnormal	Sub-assembly
refersToUsage	Data_Source	Usage_Need_Entity
usesApplication	Service	Application
usesDocument	Service	Document
usesEquipment	Service	Equipment
usesRawMaterial	Service	Raw_Material

2.6.3 Datatype Properties

Table 18 lists the datatype properties of the upper ontology. For each property, the domain and range are defined.

Table 18: High-tech products Ontology datatype properties

Domain	*Property*	*Type*	*Range*
Consumer_Client	ClientAddress	Datatype	xsd:string
Consumer_Client	ClientContact	Datatype	xsd:string
Consumer_Client	ClientEmail	Datatype	xsd:string
Consumer_Client	ClientID	Datatype	xsd:int
Consumer_Client	ClientName	Datatype	xsd:string
Consumer_Client	ClientWarranty	Datatype	xsd:int
Help_Desk_Worker	CustomerSerivceAccessAuthority	Datatype	xsd:string
Help_Desk_Worker	CustomerSerivceAddress	Datatype	xsd:string
Help_Desk_Worker	CustomerSerivceAffiliation	Datatype	xsd:string

Help_Desk_Worker	CustomerSerivceContact	Datatype	xsd:string
Help_Desk_Worker	CustomerSerivceEmail	Datatype	xsd:string
Help_Desk_Worker	CustomerSerivceID	Datatype	xsd:int
Help_Desk_Worker	CustomerSerivceName	Datatype	xsd:string
Calibration_and_Maintenance_Team	MaintenanceTeamAccessAuthority	Datatype	xsd:string
Calibration_and_Maintenance_Team	MaintenanceTeamAddress	Datatype	xsd:string
Calibration_and_Maintenance_Team	MaintenanceTeamAffiliation	Datatype	xsd:string
Calibration_and_Maintenance_Team	MaintenanceTeamContact	Datatype	xsd:string
Calibration_and_Maintenance_Team	MaintenanceTeamEmail	Datatype	xsd:string
Calibration_and_Maintenance_Team	MaintenanceTeamID	Datatype	xsd:int
Calibration_and_Maintenance_Team	MaintenanceTeamName	Datatype	xsd:string
Metrology_team	MetrologyTeamAccessAuthority	Datatype	xsd:string
Metrology_team	MetrologyTeamAddress	Datatype	xsd:string
Metrology_team	MetrologyTeamAffiliation	Datatype	xsd:string
Metrology_team	MetrologyTeamContact	Datatype	xsd:string
Metrology_team	MetrologyTeamEmail	Datatype	xsd:string
Metrology_team	MetrologyTeamID	Datatype	xsd:int
Metrology_team	MetrologyTeamName	Datatype	xsd:string
Production_Team	ProductionTeamAccessAuthority	Datatype	xsd:string
Production_Team	ProductionTeamAddress	Datatype	xsd:string
Production_Team	ProductionTeamAffiliation	Datatype	xsd:string
Production_Team	ProductionTeamContact	Datatype	xsd:string
Production_Team	ProductionTeamEmail	Datatype	xsd:string
Production_Team	ProductionTeamID	Datatype	xsd:int
Production_Team	ProductionTeamName	Datatype	xsd:string
Quality_Product_Department	QualityProductDepartmentAccessAuthority	Datatype	xsd:string
Quality_Product_Department	QualityProductDepartmentAddress	Datatype	xsd:string
Quality_Product_Department	QualityProductDepartmentAffiliation	Datatype	xsd:string
Quality_Product_Department	QualityProductDepartmentContact	Datatype	xsd:string
Quality_Product_Department	QualityProductDepartmentEmail	Datatype	xsd:string
Quality_Product_Department	QualityProductDepartmentID	Datatype	xsd:int

Quality_Product_Department	QualityProductDepartmentName	Datatype	xsd:string
Third_Party_Partner	ThridPartyAccessAuthority	Datatype	xsd:string
Third_Party_Partner	ThridPartyAddress	Datatype	xsd:string
Third_Party_Partner	ThridPartyAffiliation	Datatype	xsd:string
Third_Party_Partner	ThridPartyContact	Datatype	xsd:string
Third_Party_Partner	ThridPartyEmail	Datatype	xsd:string
Third_Party_Partner	ThridPartyID	Datatype	xsd:int
Third_Party_Partner	ThridPartyName	Datatype	xsd:string
Cost_Reduction	CostReductionAddedValue	Datatype	xsd:string
Cost_Reduction	CostReductionCurrentState	Datatype	xsd:string
Cost_Reduction	CostReductionKPI	Datatype	xsd:string
Eco_Friendliness	EcoFriendlinessAddedValue	Datatype	xsd:string
Eco_Friendliness	EcoFriendlinessCurrentState	Datatype	xsd:string
Eco_Friendliness	EcoFriendlinessKPI	Datatype	xsd:string
Performance_Improvement	PerformanceImprovementAddedValue	Datatype	xsd:string
Performance_Improvement	PerformanceImprovementCurrentState	Datatype	xsd:string
Performance_Improvement	PerformanceImprovementKPI	Datatype	xsd:string
Risk_Reduction	RiskReductionAddedValue	Datatype	xsd:string
Risk_Reduction	RiskReductionCurrentState	Datatype	xsd:string
Risk_Reduction	RiskReductionKPI	Datatype	xsd:string
Country	CountryCode	Datatype	xsd:string
Country	IncomeGroup	Datatype	xsd:string
Country	Region	Datatype	xsd:string
Questionanaires	CallCentreLogTime	Datatype	xsd:dateTime
Questionanaires	CallCentreProblem	Datatype	xsd:string
Questionanaires	EndTime	Datatype	xsd:dateTime
Questionanaires	ProblemReason	Datatype	xsd:string
Questionanaires	Solution	Datatype	xsd:string
Questionanaires	StatusOfSolution	Datatype	xsd:string
M3_Reports	SWLog	Datatype	xsd:string
M3_Reports	SWLogTime	Datatype	xsd:dateTime
Customer_Complaint_Entity	ComplaintAssessment	Datatype	xsd:string

Customer_Complaint_Entity	ComplaintComplexity	Datatype	xsd:string
Customer_Complaint_Entity	ComplaintCriticality	Datatype	xsd:string
Customer_Complaint_Entity	ComplaintReiteration	Datatype	xsd:string
Development_Need_Entity	DevelopmentNeedAssessment	Datatype	xsd:string
Development_Need_Entity	DevelopmentNeedComplexity	Datatype	xsd:string
Development_Need_Entity	DevelopmentNeedCriticality	Datatype	xsd:string
Development_Need_Entity	DevelopmentNeedReiteration	Datatype	xsd:string
Maintenance_Need_Entity	MaintenanceNeedAssessment	Datatype	xsd:string
Maintenance_Need_Entity	MaintenanceNeedComplexity	Datatype	xsd:string
Maintenance_Need_Entity	MaintenanceNeedCriticality	Datatype	xsd:string
Maintenance_Need_Entity	MaintenanceNeedReiteration	Datatype	xsd:string
Customer_Recommendation_Entity	RecommendationAssessment	Datatype	xsd:string
Customer_Recommendation_Entity	RecommendationComplexity	Datatype	xsd:string
Customer_Recommendation_Entity	RecommendationCriticality	Datatype	xsd:string
Customer_Recommendation_Entity	RecommendationReiteration	Datatype	xsd:string
Usage_Need_Entity	UsageNeedEntityAssessment	Datatype	xsd:string
Usage_Need_Entity	UsageNeedEntityComplexity	Datatype	xsd:string
Usage_Need_Entity	UsageNeedEntityCriticality	Datatype	xsd:string
Usage_Need_Entity	UsageNeedEntityReiteration	Datatype	xsd:string
Part	PartID	Datatype	xsd:int
Part	PartName	Datatype	xsd:string
Part	PartType	Datatype	xsd:string
Sub-assembly	Sub-assemblyID	Datatype	xsd:int
Sub-assembly	Sub-assemblyName	Datatype	xsd:string
Sub-assembly	Sub-assemblyType	Datatype	xsd:string
OptiScan_Sensor	Accuracy	Datatype	xsd:string
OptiScan_Sensor	DigitisignSpeed	Datatype	xsd:string
OptiScan_Sensor	FaserSafety	Datatype	xsd:string
OptiScan_Sensor	FieldOfView	Datatype	xsd:string
OptiScan_Sensor	ProductID	Datatype	xsd:int
OptiScan_Sensor	ProductName	Datatype	xsd:string
OptiScan_Sensor	ProductSerialNumber	Datatype	xsd:string

OptiScan_Sensor	Weight	Datatype	xsd:string
OptiScan_Sensor	WorkingDistance	Datatype	xsd:string
Error	CurrentState	Datatype	xsd:string
Error	DownTime	Datatype	xsd:dateTime
Error	ErrorCode	Datatype	xsd:string
Error	ErrorLogtime	Datatype	xsd:dateTime
Error	ErrorSolution	Datatype	xsd:string
Execution	ExecutionOfCommand	Datatype	xsd:string
Execution	NextCommand	Datatype	xsd:string
Machine_Movements	Multipoint	Datatype	xsd:string
Machine_Movements	PositionX	Datatype	xsd:float
Machine_Movements	PositionY	Datatype	xsd:float
Machine_Movements	PositionZ	Datatype	xsd:float
Probing_points	CoordinateX	Datatype	xsd:float
Probing_points	CoordinateY	Datatype	xsd:float
Probing_points	CoordinateZ	Datatype	xsd:float
Probing_points	SelectedScanPointset	Datatype	xsd:string
Qualification	AngleCorrectionX	Datatype	xsd:float
Qualification	AngleCorrectionY	Datatype	xsd:float
Qualification	AngleCorrectionZ	Datatype	xsd:float
Qualification	CenterOfFinalSphereX	Datatype	xsd:float
Qualification	CenterOfFinalSphereY	Datatype	xsd:float
Qualification	CenterOfFinalSphereZ	Datatype	xsd:float
Qualification	CenterOfSphereX	Datatype	xsd:float
Qualification	CenterOfSphereY	Datatype	xsd:float
Qualification	CenterOfSphereZ	Datatype	xsd:float
Qualification	DeviationOfFinalSphereX	Datatype	xsd:float
Qualification	DeviationOfFinalSphereY	Datatype	xsd:float
Qualification	DeviationOfFinalSphereZ	Datatype	xsd:float
Qualification	DeviationOfSphereX	Datatype	xsd:float
Qualification	DeviationOfSphereY	Datatype	xsd:float
Qualification	DeviationOfSphereZ	Datatype	xsd:float

Qualification	InitialPositionOfQualificationA	Datatype	xsd:float
Qualification	InitialPositionOfQualificationB	Datatype	xsd:float
Selection_Of_Part	LastPart	Datatype	xsd:string
Selection_Of_Part	NumberOfSelectedPart	Datatype	xsd:int
Selection_Of_Part	SavedPartNumber	Datatype	xsd:int
Precaution	PrecautionCriticality	Datatype	xsd:string
Precaution	PrecautionType	Datatype	xsd:string
Precaution	ResidualLifeTime	Datatype	xsd:dateTime
Product_System	ProductSystemProperty	Datatype	xsd:string
Product_System	ProductSystemID	Datatype	xsd:int
Product_System	ProductSystemName	Datatype	xsd:string
HighTech_PSS	PSSID	Datatype	xsd:int
HighTech_PSS	PSSName	Datatype	xsd:string
Application	ApplicationAvailability	Datatype	xsd:string
Application	ApplicationProductivity	Datatype	xsd:string
Application	ApplicationQuality	Datatype	xsd:string
Document	DocumentAvailability	Datatype	xsd:string
Document	DocumentProductivity	Datatype	xsd:string
Document	DocumentQuality	Datatype	xsd:string
Equipment	EquipmentAvailability	Datatype	xsd:string
Equipment	EquipmentProductivity	Datatype	xsd:string
Equipment	EquipmentQuality	Datatype	xsd:string
Raw_Material	RawMaterialAvailability	Datatype	xsd:string
Raw_Material	RawMaterialProductivity	Datatype	xsd:string
Raw_Material	RawMaterialQuality	Datatype	xsd:string
Development_Service	DevelopmentServiceID	Datatype	xsd:int
Development_Service	DevelopmentServiceName	Datatype	xsd:string
Development_Service	DevelopmentServiceStatus	Datatype	xsd:string
Development_Service	DevelopmentServiceType	Datatype	xsd:string
Maintenance_Service	MaintenanceServiceID	Datatype	xsd:int
Maintenance_Service	MaintenanceServiceName	Datatype	xsd:string
Maintenance_Service	MaintenanceServiceStatus	Datatype	xsd:string

Maintenance_Service	MaintenanceServiceType	Datatype	xsd:string
Usage_Service	UsageServiceID	Datatype	xsd:int
Usage_Service	UsageServiceName	Datatype	xsd:string
Usage_Service	UsageServiceStatus	Datatype	xsd:string
Usage_Service	UsageServiceType	Datatype	xsd:string
Software_Element	SoftwareID	Datatype	xsd:int
Software_Element	SoftwareName	Datatype	xsd:string

2.6.4 Ontology mapping business story

The current Deliverable 8.2 describing business stories to indentify and analize the the potential product-service featrues of High-tech Products scenario, and each business story has the list of questions. High-tech Products ontology can answer some of them. Therefore, this chapter introduces the mapping between High-tech Products ontology and the list of questions in business stories. The list of questions High-tech Products ontology maps are ; (Q1/5): To get the machine's historical evolution from log files/log file of each client

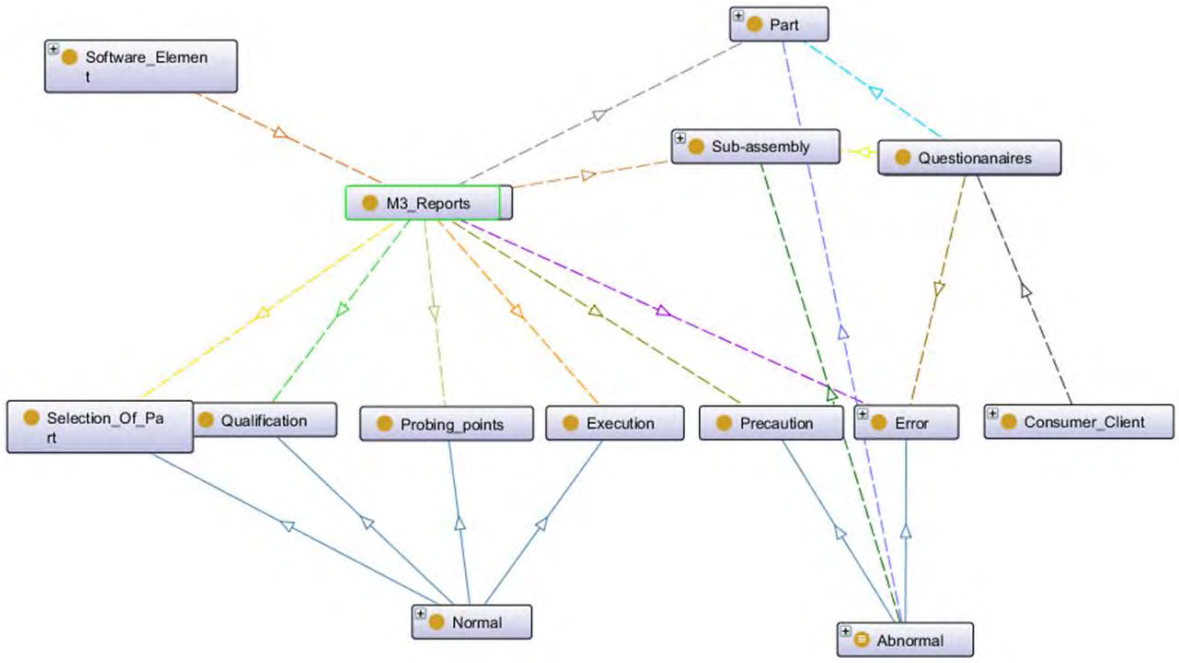

Figure 15. High-tech Products ontology mapping High-tech Products Business stories

To respond to two questions, High-tech Products ontology has class M3_Report, Questionanaires, and classes representing product states and product components. It describes that M3_Report *hasLinkTo* a product state, Customer_Client *generates* Questionanaries, Questionanaries *isRelatedTo* a product state or product components. If a product state is abnormal, an Abnormal instance has a relation with product compoents. If a Helpdesk Support opens PUI Query Manager, he/she can selects a product state. After then, he/she can get the data instances classified into product states or product components. Further on, extraction classificated data from PUI can be visualized using data visualisation Module.

3 STATE OF THE ART

In the following sections, State of the Art of semantic technologies will be presented. First of all, the existing ontology with ontology comparison will be present in Chapter 3.1. Chapter 3.2 will introduce Ontology Engineering, and Chapter 3.3 will present semantic modelling including ontology languages. Finally, Chapter 3.4 will introduce W3C standards of semantic web, linked data and query language.

3.1 Existing Ontologies

In order to capture the standardization concepts of PSS with the domain of Product Lifecycle, other existing ontologies such as Basic Formal Ontology (BFO), LinkedDesign ontology, and Diversity general category/PSS have been referred in the development of FALCON ontology. In this sense, FALCON ontology have been updated to represent generalization of concepts between all the business cases based on comparison works with existing ontologies. Comparison of FALCON generalization concepts with others existing is highlighted in the following Table 19.

Hereafter FALCON upper ontology is elaborated, presenting all their concepts, object and datatype properties together with comments explaining their use. The resulting model will be in the form of an ontology network where each business scenario specific ontology will be connected to an upper ontology through generalized concepts.

Table 19: Ontology comparison

FALCON upper ontology	*BFO/PLC*	*LDO Upper ontology*	*Diversity General Category/PSS*
Product Service System		LDO	PSS
Product Feature	Information Content Entity		Product Type
Software Feature			
Software Element	Immatrial Entity		
Country			
Product System	Independent_Countinuant		
Product	Material Entity	Product	- Product
Part		-Module	- - Product Component
Sub-assembly		-Part	
Actor		Actor	Stakeholder
			- Individual
			--Employee
			--Service Provider
			- - Supplier

-PSS Provider			
--Customer Serivce			
--Data Analyst			
--Developer			
--Proejct Manager			
--Support Serivce			
-Thrid Party Partner			
Consumer Client			Customer
			-Business Customer
			-Consumer
			- Company (Group)
			- - Product Vendor
Resource			
-Digital Resource		Resource	
--Application		-Digital Resource	
--Document		--Application	
-Physical Resource		--Document	
--Equipment		-Physical Resource	

--Raw Material		--Equipment	
Product Status		--Raw Material	
			Infrastructure
			- Hardware
			- Software
			Social Media Environment
-Abnormal	Specifically Dependent Continuant		
-Normal			
-Data Source			
--Product Data			
--User Feedback			Feedback
			Opinion
			Global sentiment
PSS Type			
PLC			PLC

-BOL	Process	PLC	
-MOL		-BOL	
-EOL		-MOL	- - Service Component
Service		-EOL	- Service
			- - Service Component
-Developemtn Service			
-Usage Service			
-Maintenance Service			
-Recycling Service			
-Disposal Service			
Service System			
			Service Type
			Guidelines (e.g. Lean Design Rules
		-Process	
		-ProcessEvent	
Benefit Entity		-Task	
-Cost Reduction	Temporal Attribute Entity		

-Eco Friendliness			
-Performance Improvement			
-Risk Reduction			
Need Requirement Entity			
-Customer Requirement Entity			
-Funtional Need Entity			
	Attribute	Event	
		-TaskEvent	
		Factor	
		Indicator	
			Generic KPI
			- Customer KPI
			- Design KPI

			- Environment KPI
			- Manufacturing KPI

3.2 Ontology Engineering

The ontology engineering is the general term of methodologies and methods for building ontologies. Ontology engineering refers to “The set of activities that concern the ontology development and the ontology lifecycle, the methods and methodologies for building ontologies and the tool suites and languages that support them” (Suárez-Figueroa et al., 2011). The results of ontology engineering provide domain knowledge representation to be reused efficiently, and prevent waste of time and money which are usually caused by non-shared knowledge. In addition, it helps Information Technology (IT) to operate with interoperability and standardization.

3.3 Semantic Modelling

3.3.1 Definition and components

Ontology represents the nature of being, becoming, existence, and so on in the way of philosophy. Among various definitions of ontology, the most well-known definition of ontology is : “ontology is an explicit, formal specification of a shared conceptualization of a domain of interest” (Gruber, 1993). In other words, ontology is the machine understandable meta-model which defines different kinds of concepts and their relations based on the consensus knowledge among not only the members of the domain but also computers.

Ontology represents the following ideas together (Calero at al., 1993):

- Semantic modelling can help defining the data and the relationships between entities.
- An information model provides the ability to abstract different kind of data and provides an understanding of how the data elements are related.
- A semantic model is a type of information model that supports the modelling of entities and their relationships.
- The total set of entities in our semantic model comprises the taxonomy of classes we use in our model to represent the real world.

The main objective of semantic modelling techniques is to define the meaning of data within the context of its correlation, and to model the real world in the abstract level. The benefits of exploiting semantic data models for business applications are mainly as follows:

- Avoiding misunderstanding: by providing a clear, accessible, agreed set of terms, relations as a trusted source and discussions, misunderstandings can easily be resolved.
- Conduct reasoning: by being machine understandable and through the usage of logic statements (rules), ontologies enable automatic reasoning and inference which leads to automatic generation of new and implicit knowledge.
- Leverage resources: by extending and relating an application ontology to external ontological resources, via manual or automatic mapping and merging processes, the need for repetition of entire design process for every application domain is eliminated.

- Improve interoperability: semantic models can serve as a basis for schema matching to support systems' interoperability in close environments where systems, tools and data sources have no common recognition of data type and relationships.

Ontologies provide formal models of domain knowledge exploited in different ways. Therefore, it is important that ontology plays a significant role for many knowledge-intensive applications.

Depending on corresponding languages, different knowledge representation formalisms exist. However, they consist of the following minimal set of components and share them:

- **Classes** represent concepts, which are taken in a broad sense. For instance, in the Product Licecycle domain, concepts are: Life Cycle phase, Product, Activity, Resources, Even, and so on. Classes in ontology are usually organized in taxonomies through which inheritance mechanisms can be applied.

- **Relations** represent a type of association between concepts of the domain. They are formally defined as any subset of a product of *n* sets, that is: $R \subset C_1 \times C_2 \times ... \times C_n$. Ontologies usually contain binary relations. The first argument is known as the domain of the relation, and the second argument is the range.

- **Formal axioms** serve to model sentences that are always true. They are normally used to represent knowledge that cannot be formally defined by the other components. In addition, formal axioms are used to verify the consistency of the ontology itself or the consistency of the knowledge stored in a knowledge base. Formal axioms are very useful to infer new knowledge.

  For instance, Energy Efficiency at Buildings domain could be that it is not possible to build a public building without a fire door (based on legal issues).

- **Instances** are used to represent elements or individuals in an ontology.

As a Design Rationale (DR), ontology can be used as follows (Mizoguchi & Ikeda, 1998):

Level 1: Used as a common vocabulary for communication among distributed agents.

Level 2: Used as a conceptual schema of a relational database. Structural information of concepts and relations among them is used. Conceptualization in a database is nothing other than conceptual schema. Data retrieval from a database is easily done when there is an agreement on its conceptual schema.

Level 3: Used as the backbone information for a user of a certain knowledge base. Levels higher than this plays roles of the ontology, which has something to do with "content".

Level 4: Used for answering competence questions.

Level 5: Standardization 5.1 Standardization of terminology (at the same level of Level 1) 5.2 Standardization of meaning of concepts 5.3 Standardization of components of target objects (domain ontology). 5.4 Standardization of components of tasks (task ontology)

Level 6: Used for transformation of databases considering the differences of the meaning of conceptual schema. This requires not only the structural transformation but also semantic transformation.

Level 7: Used for reusing knowledge of a knowledge base using DR information.

Level 8: Used for reorganizing a knowledge base based on DR information.

3.3.2 Foremost methodologies for building ontologies

The well referred methodologies for building ontologies are METHONTOLOGY, On-To-Knowledge and DILIGENT. These methodologies mainly include guidelines for single ontology construction ranging from ontology requirements specification to ontology implementation and they are mainly targeted to ontology researchers. Comparison of these methodologies are presented as follows:

METHONTOLOGY enables the construction of ontologies at the knowledge level. It includes (a) the identification of the ontology development process; (b) a life cycle based on evolving prototypes; and (c) some techniques to carry out management, development-oriented, and support activities. With respect to the aforementioned dimensions, notions of collaboration are not included. Although dynamic aspects are mentioned, detailed guidelines about how to manage versions are An ontology network is a collection of ontologies related together via a variety of different meta-relationships such as mapping, modularization, version, and dependency relationships. Collaboration refers to consider distributed ontology engineering among heterogeneous and geographically distributed groups of domain experts and ontology practitioners. Dynamism refers to the evolution and versioning of the ontologies not provided. Regarding the reuse of knowledge resources, METHONTOLOGY includes the list of activities to be carried out during ontology reuse and reengineering processes, but it does not provide detailed guidelines for such processes, nor does it consider different levels of granularity during the reuse of ontological resources (e.g., modules or statements). Moreover, METHONTOLOGY does not consider the reuse of ontology design patterns (ODPs) neither the reuse nor reengineering of non-ontological resources.

The **On-To-Knowledge methodology** proposes to build ontologies taking into account how these are going to be used in knowledge management applications. The processes proposed by this methodology are the following: feasibility study, kick-off, refinement, evaluation, and maintenance. Regarding the aspects analysed in this paper, such a methodology does not consider collaboration. Regarding the dynamic evolution of ontologies, it proposes to create a new version after testing possible effects to the application. However, no guidelines about how to manage different versions and when to create them are provided. With respect to the reuse of knowledge resources, in the kick-off process it is mentioned that developers should look for potentially reusable ontologies. However, this methodology does not provide detailed guidelines for identifying such ontologies nor for reusing them. Besides, the methodology does not explicitly mention guidelines for the reuse and re-engineering of non-ontological resources, nor for the reuse of ODPs.

The **DILIGENT** methodology is intended to support domain experts in a distributed setting in order to engineer and evolve ontologies. This methodology is focused on collaborative and distributed ontology engineering. Its ontology development process includes the following five activities: building, local adaptation, analysis, revision, and local update. With respect to the dimensions analysed here, collaboration is the central point in this methodology. Regarding the dynamic dimension, DILIGENT proposes the creation of different versions of the ontology, but it does not provide guidelines on how to manage such versions or when to create different versions, nor how changes can affect. With regard to the reuse of knowledge resources, the methodology does not include guidelines for the reuse and reengineering of existing knowledge resources.

3.3.3 Ontology Languages

- **XML-based Ontology Exchange Language :** The US bioinformatics community designed XOL for the exchange of ontology definitions among a heterogeneous set of software systems in their domain. Researchers created it after studying the representational needs of experts in bioinformatics. They selected Ontolingua and OML as the basis for creating XOL, merging the high expressiveness of OKBC-Lite, a subset of the Open Knowledge Based Connectivity protocol, and the syntax of OML, based on XML. There are no tools that allow the development of ontologies using XOL. However, since XOL files use XML syntax, we can use an XML editor to author XOL files.

- **Simple HTML Ontology Extension** : SHOE, developed at the University of Maryland and used to develop OML, was created as an extension of HTML, incorporating machine-readable semantic knowledge in HTML documents or other Web documents. Recently, the University of Maryland has adapted the SHOE syntax to XML. SHOE makes it possible for agents to gather meaningful inforThe authors analyze the most representative ontology languages created for the Web and compare them using a common framework. mation about Web pages and documents, improving search mechanisms, and knowledge gathering. This process consists of threephases: Define an ontology, annote HTML pages with ontological information to describe themselves and other pages, and have an agent semantically retrieve information by searching all the existing pages and keeping information updated. The Knowledge Annotator annotates ontological information in HTML pages.

- **Ontology Markup Language** : OML, developed at the University of Washington, is partially based on SHOE. In fact, it was first considered an XML serialization of SHOE. Hence, OML and SHOE share many features. Four different levels of OML exist: OML Core is related to logical aspects of the language and is included by the rest of the layers; Simple OML maps directly to RDF(S); Abbreviated OML includes conceptual graphs features; and Standard OML is the most expressive version of OML. We selected Simple OML, because the higher layers don't provide more components than the ones identified in our framework. These higher layers are tightly related to the representation of conceptual graphs. There are no other tools for authoring OML ontologies other than existing generalpurpose XML edition tools.

- **Ontology Interchange Language :** OIL, developed in the OntoKnowledge project (www.ontoknowledge.org/OIL), permits semantic interoperability between Web resources. Its syntax and semantics are based on existing proposals (OKBC, XOL, and RDF(S)), providing modeling primitives commonly used in frame-based approaches to ontological engineering (concepts, taxonomies of concepts, relations, and so on), and formal semantics and reasoning support found in description logic approaches (a subset of first order logic that maintains a high expressive power, together with decidability and an efficient inference mechanism). OIL, built on top of RDF(S) (see Figure 1), has the following layers: Core OIL groups the OIL primitives that have a direct mapping to RDF(S) primitives; Standard OIL is the complete OIL model, using more primitives than the ones defined in RDF(S); Instance OIL adds instances of concepts and roles to the previous model; and Heavy OIL is the layer for future extensions of OIL. OILEd, Protégé2000, and WebODE can be used to author OIL ontologies. OIL's syntax is not only expressed in XML but can also be presented in ASCII. We use ASCII for our examples.

- **DARPA Agent Markup Language+OIL** : DAML+OIL has been developed by a joint committee from the US and the European Union (IST) in the context of DAML, a DARPA project for allowing semantic interoperability in XML. Hence, DAML+OIL shares the same objective as OIL. DAML+OIL is built on RDF(S). Its name implicitly suggests that there is a tight relationship with OIL. It replaces the initial specification, which was called DAML-ONT, and was also based on the OIL language. OILEd, OntoEdit, Protégé2000, and WebODE are tools that can author DAML+OIL ontologies.
- **OWL** : OWL is the result of the work of the W3C Web Ontology Working Group. This language derived from DAML+OIL and, as the previous languages, is intended for publishing and sharing ontologies in the Web. OWL is built upon RDF(S), has a layered structure and is divided into three sublanguages: OWL Lite, OWL DL and OWL Full. OWL is grounded on Description Logics and its semantics are described in two different ways: as an extension of the RDF(S) model theory and as a direct model-theoretic semantics of OWL. Both of them have the same semantic consequences on OWL ontologies.
- **OWL 2** : OWL 2 is an extension and revision of OWL that adds new functionality with respect to OWL; some of the new features are syntactic sugar (e.g., disjoint union of classes) while others offer new expressivity. OWL 2 includes three different profiles (i.e., sublanguages) that offer important advantages in particular application scenarios, each trading off different aspects of OWL's expressive power in return for different computational and/or implementation benefits. These profiles are:
  - **OWL 2 EL**: It is particularly suitable for applications where very large ontologies are needed, and where expressive power can be traded for performance guarantees.
  - **OWL 2 QL**: It is particularly suitable for applications where relatively lightweight ontologies are used to organize large numbers of individuals and where it is useful or necessary to access the data directly via relational queries (e.g., SQL).
  - **OWL 2 RL**: It is particularly suitable for applications where relatively lightweight ontologies are used to organize large numbers of individuals and where it is useful or necessary to operate directly on data in the form of RDF triples. OWL 2 ontologies: the Direct Semantics that assigns meaning directly to ontology structures and the RDF-Based Semantics that assigns meaning directly to RDF graphs.
- **Resource Description Framework and RDF Schema :** RDF, developed by the W3C for describing Web resources, allows the specification of the semantics of data based on XML in a standardized, interoperable manner. It also provides mechanisms to explicitly represent services, processes, and business models, while allowing recognition of no explicit information. The RDF data model is equivalent to the semantic networks formalism. It consists of three object types: resources are described by RDF expressions and are always named by URIs plus optional anchor IDs; properties define specific aspects, characteristics, attributes, or relations used to describe a resource; and statements assign a value for a property in a specific resource (this value might be another RDF statement). The RDF data model does not provide mechanisms for defining the relationships between properties (attributes) and resources—this is the role of RDFS. RDFS offers primitives for defining knowledge models that are closer to frame-based approaches. RDF(S) is widely used as a representation format in many tools and projects, such as Amaya, Protégé, Mozilla, SilRI, and so on.

According to W3C, RDF model has advantages : (1) The RDF model is made up of triples: as such, it can be efficiently implemented and stored; other models requiring variable-length fields would require a more cumbersome implementation, (2) The RDF model is essentially the canonicalization of a (directed) graph, and so as such has all the advantages (and generality) of structuring information using graphs, (3) The basic RDF model can be processed even in absence of more detailed information (an "RDF schema") on the semantics: it already allows basic inferences to take place, since it can be logically seen as a fact basis, (4) The RDF model has the important property of being modular: the union of knowledge (directed graphs) is mapped into the union of the corresponding RDF structures. Since RDF is a standard model for data interchange on the Web and is a W3C recommendation designed to standardize the definition and use of metadata-descriptions of Web-based resources. It is well suited to representing data as well. As knowledge representation, when it comes to semantic interoperability, RDF has significant advantages (Decker at al., 2000) : The object-attribute structure provides natural semantic units because all objects are independent entities. A domain model—defining objects and relationships—can be represented naturally in RDF. To find mappings between two RDF descriptions, techniques from research in knowledge representation are directly applicable. Of course, this does not solve the general interoperability problem of finding semantics-preserving mappings between objects, but using RDF for data interchange raises the level of potential reuse of software components. Therefore, the FALCON ontology will be implemented in RDF format.

4 ONTOLOGY IMPLEMENTATION

4.1 Knowledge Extraction

The FALCON ontology has been designed to apply USM which facilitates the satisfaction of requirements of stakeholders in the domain field. From this methodology, the domain of interest was extracted and provided for the FALCON ontology, referring from various data sources. The FALCON ontology serves as a common reference model for the annotation and description in the context of FALCON project. Moreover, the FALCON ontology describes the basic entities of the FALCON project and model relevant Product-Service structures. This implicit and explicit knowledge adds value for people who try to understand domain knowledge of FALCON project. Furthermore, the FALCON ontology constitutes the formal representation of the FALCON semantic model and the knowledge that this model encapsulates as the part of the FALCON VOP. Therefore, codification of the knowledge will allow to exchange information regarding the Product-Service context and to be desirable to use it, in order to increase added value of the FALCON platform.

Currently the FALCON VOP is demonstrated by five business scenarios that are representatives of Product-Service domain and each one has unique challenges and requirements. The upper ontology provides the generalization of Product-Service domain knowledge and each specific ontology describes domain knowledge of a specific business scenario to meet its requirements. Further on, domain knowledge of the Product-Service system in the FALCON context can be visualized as a graph from the FALCON ontology. The nodes of graph can show entire entities and edges can demonstrate the various relations between entities, to make it easily understandable.

4.2 DATA integration and Semantic Interoperability

The FALCON project is oriented by the main idea to exploit two kinds of advanced enabling technologies during MOL, in order to get feedback from the customer and monitor the product states. The FALCON ontology plays a role which is to integrate various data sources for intelligent filtering. By the relations between the data sources and relevant entities, Feedback from help desk or social media and Data representing the product states will have the labels to add sematic structure from the ontology. Therefore, one of the key benefits of Semantic technologies is the creation of data for identifying data sources which will have semantic meaning into the ontology.

On the other hand, data integration enables FALCON VOP to have semantic interoperability. The semantic interoperability provides abilities to exchange data and information with unambiguous, shared meaning to the platform. This is a requirement of machine computable logic, inferencing, knowledge discovery, and data federation between various modules in the platform.

5 Conclusion

This document presented the implementation details of the FALCON semantic model as an ontology. The FALCON ontology has been developed including the domain of interest for all the business scenarios of the FALCON project. In order to meet the requirements of stakeholders in the domain field, USM method has been applied.

In addition, to adopt BFO framework will provide availability to merge the other Product domain ontology structured by BFO of which domain is not Product Service systems. Moreover, implicit and explicit knowledge as ontology add value for people who try to understand domain knowledge of the FALCON project. Furthermore, this ontology constitutes the formal representation of the FALCON semantic model and the knowledge that this model encapsulates as the part of the FALCON VOP. Therefore, codification of the knowledge will allow information exchange regarding the Product-Service context and the desire to use it, in order to increase added value of the FALCON platform.

6 References

Goedkoop, M. J., Van Halen, C. J., Te Riele, H., & Rommens, P. J. (1999). Product service systems, ecological and economic basics. Report for Dutch Ministries of environment (VROM) and economic affairs (EZ), 36(1), pp. 1-122.

Councill, B., & Heineman, G. T. (2001). Definition of a software component and its elements. Component-based software engineering: putting the pieces together, pp. 5-19.

Arp, R., Smith, B., & Spear, A. D. (2015). Building ontologies with basic formal ontology. Mit Press.

Cook, M. (2004). Understanding the potential opportunities provided by service orientated concepts to improve resource productivity. Professional Engineering Publishing Limited.

Dong, M., Yang, D., & Su, L. (2011). Ontology-based service product configuration system modeling and development. Expert systems with applications, 38(9), pp. 11770-11786.

Smith, B., Almeida, M., Bona, J., Brochhausen, M., Ceusters, W., Courtot, M., ... & Hogan, W. (2014). Basic Formal Ontology 2.0 Draft Specification and User's Guide.

M. C. Suárez-Figueroa, R. García-Castro, B. Villazón Terrazas, and A. Gomez-Pérez, "Essentials In Ontology Engineering: Methodologies, Languages, And Tools," 2011

Gruber, T. R. (1993) A translation approach to portable ontology specifications. Knowledge acquisition, 5(2), pp. 199-220.

Calero, C., Ruiz, F., & Piattini, M. (Eds.). (2006). Ontologies for software engineering and software technology. Springer Science & Business Media.

Mizoguchi, R. & Ikeda, M. (1998) Towards ontology engineering. Journal-Japanese Society for Artificial Intelligence, 13, pp. 9-10.

Suárez-Figueroa, M. C., & Gómez-Pérez, A. (2009). NeOn methodology for building ontology networks: a scenario-based methodology. In Proceedings of the International Conference on Software, Services & Semantic Technologies. Sofia.

B. Motik, P. F. Patel-Schneider, B. Parsia, C. Bock, A. Fokoue, P. Haase, R. Hoekstra, I. Horrocks, A. Ruttenberg, U. Sattler, and others, "OWL 2 web ontology language: Structural specification and functional-style syntax," W3C Recomm., vol. 27, 2009.

Gómez-Pérez, A., & Corcho, O. (2002). Ontology languages for the semantic web. IEEE Intelligent systems, 17(1), 54-60.

Decker, S., Melnik, S., Van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., ... & Horrocks, I. (2000). The semantic web: The roles of XML and RDF. IEEE Internet computing, 4(5), 63-73.