DIGICOR: Decentralised Agile Coordination Across Supply Networks

WP 5: Middleware and supportive tools and interfaces

D 5.10: Semantic Data Models Implementation

Deliverable Lead: CTU

Main Editor: Václav Jirkovský, CTU

Contributing Partners: SLG, UNIMAN

Date: 2019-03-29

Dissemination: Public

Status: Consortium agreed

This deliverable describes semantic data models developed within the framework of DIGICOR project. The fundamental semantical concepts and relations are covered by DIGICOR Core Ontology. A domain specific ontology aiming at an automotive production scenario is introduced. Data Exchange Ontology is described as a means for integration of various DIGICOR components. The last three sections of this deliverable describe utilization of developed semantic data models for production control and monitoring, tender decomposition, and security purposes.

Document Status	
Deliverable Lead	Václav Jirkovský, CTU
Internal Reviewer 1	Petr Kadera, CTU
Internal Reviewer 2	Filip Jírů, CER
Type	Deliverable
Work Package	WP5: Middleware and supportive tools and interfaces
ID	D5.10: Semantic Data Models Implementation
Due Date	2019-03-31
Delivery Date	2019-03-29
Status	For Review

History

See Annex

Status

This deliverable is subject to final acceptance by the European Commission.

Further Information

www.digicor-project.eu

Disclaimer

The views represented in this document only reflect the views of the authors and not the views of the European Union. The European Union is not liable for any use that may be made of the information contained in this document.

Furthermore, the information is provided “as is” and no guarantee or warranty is given that the information is fit for any particular purpose. The user of the information uses it at its sole risk and liability.

Project Partners:

AIRBUS GROUP

fortiss

almende ORGANIZING NETWORKS

MANCHESTER 1824 The University of Manchester

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

COMAU

CERTICON

CONTROL 2K total solutions provider

HANSE-AEROSPACE e.V.

Singular Logic

ICE Information Catalyst

Executive Summary

This deliverable provides a top-level view on semantic data models that represent the information backbone of the DIGICOR platform. It provides description of the corner stone of the DIGICOR semantic data models – DIGICOR Core Ontology (DCO). Further, it demonstrates on an automotive scenario how a domain ontology can be built on top of the DCO. Next section provides an overview of the Data Exchange Ontology that provides a sematic bridge among various DIGICOR components. Last sections are devoted to description of sematic-enabled features provided by the DIGICOR platform.

Table of Contents

1 Introduction to Deliverable

1.1 Deliverable Purpose and Scope

The goal of this deliverable is to provide an overview of sematic data models developed within the framework of DIGICOR project and their utilisation for several purposes.

1.2 Target Audience

This deliverable is a public document promoting the semantic data model layer of the DIGICOR project to the broad audience.

1.3 Deliverable Context

This deliverable is an output of the WP5 task T5.5.

1.4 Document Structure

This deliverable is broken down into the following sections:

- **Section 1:** Introduction to Deliverable
- **Section 2:** DIGICOR Core Ontology
- **Section 3:** Design and implementation of domain specific ontology
- **Section 4:** Data Exchange Ontology
- **Section 5:** Utilization of Semantics within DIGICOR Tools: Semantic Control and Monitoring Tool
- **Section 6:** Collaboration Ontology for TDMS and DigiGov services
- **Section 7:** Collaboration policies
- **Annexes**:
  - **Annex A**: Document History
  - **Annex B**: Part of presented Data Exchange Ontology together with SWRL rules and generated axioms

1.5 Document Status

This deliverable is a public document.

1.6 Document Dependencies

This document references deliverable D3.5 – Semantic data models specification

1.7 External Annexes and Supporting Documents

No External Annexes or Supporting Documents are referenced by this document.

2 DIGICOR Core Ontology

The DIGICOR platform represents a complex ecosystem which consists of many loosely or tightly coupled interoperable components. Furthermore, the platform is not intended to be limited on only one domain but aims to be the versatile and flexible system which should be applicable on broad spectrum of diverse domains and also provide interoperability across various suppliers. Thus, the backbone knowledge model of DIGICOR platform is built using Semantic Web technologies[1]. In deliverable D3.5 - Sematic Data Models Specification, DIGICOR Core Ontology (DCO) was designed as well as implemented.

DCO was proposed for enabling easy incorporation of various aspects of manufacturing (e.g., tendering or scheduling) as well as for easy deployment of DIGICOR platform in various domains. The main idea is to use DCO as a core knowledge model and a new deployment of DIGICOR to a new domain reside only in the development of a corresponding domain ontology. From the other point of view, the new domain ontology has to be derived from DCO using its general concepts. As prevention of misunderstanding of DCO concepts meaning, DCO exploits DOLCE ontology[2] as its cornerstone, which provides a possibility to clearly define DCO concepts using relations to its more general concepts. Overall schema of DIGICOR Core Ontology architecture is shown in Figure 1.

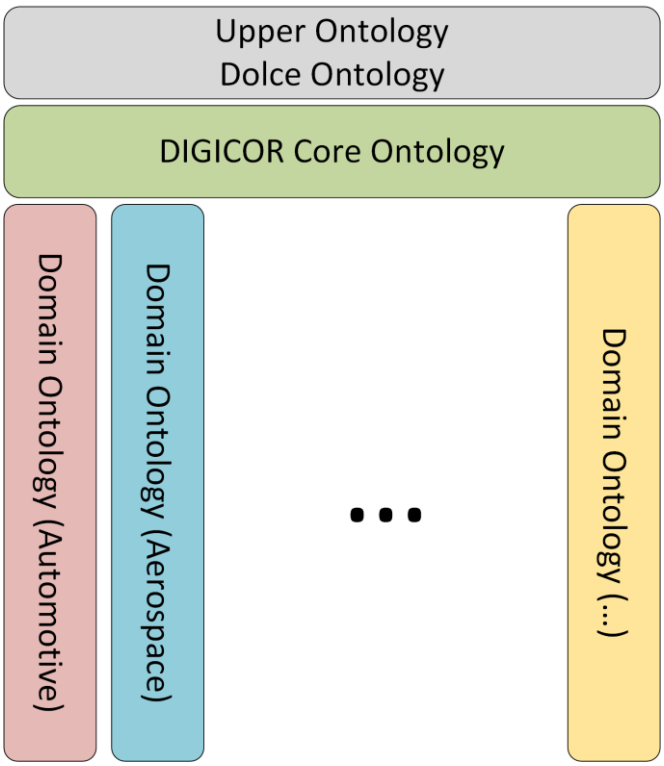

Figure 1: High-level schema of DIGICOR Core Ontology and relation to the DOLCE and domain ontologies

[1] https://www.w3.org/standards/semanticweb/

[2] http://www.loa.istc.cnr.it/dolce/overview.html

There may be distinguished two parts within DCO, i.e., the tender-related part and the scheduling-related par. The tender-related part is based on OASIS Standard - Universal Business Language (UBL)[3] and covers concepts together with their relations such as a tender, tenderer, contracting authority, tender requirement, or contract. The scheduling related part is mainly based on ADACOR ontology (ADaptive holonic COntrol aRchitecture for distributed manufacturing systems) and covers concepts and their relations such as a product, recipe, operation, property, job, schedule, or resources. The detailed description of these DCO parts is in the deliverable D3.5.

The current status of DCO corresponds to actual requirements coming from DIGICOR components. Therefore, it is expected the definition of some concepts or relations may be adapted in the future. For example, the ontology (when referring to the definition in D3.5) was extended by PDDL-related entities and their relations which are described in Section 5 of this document.

[3] http://docs.oasis-open.org/ubl/os-UBL-2.2/UBL-2.2.html

3 Design and implementation of domain specific ontology

The implementation of semantic models within DIGICOR resides primarily in the design of domain ontologies and possible adaptation of existing DCO according to arising requirements. Currently (in the time of the formation of this deliverable), the most elaborated and adequately complex domain ontology within DIGICOR platform is the automotive ontology, and therefore this ontology will be presented in this section.

The motivation for the development of the automotive domain ontology was to provide means and support for the flexible manufacturing within the automotive domain. The flexible manufacturing is understood as capabilities for providing distributed partly-autonomous production system (concerning one or more suppliers) which is robust and is capable to re-plan production in a case of a failure or unexpected outage of some resource based on the defined knowledge base. The important part of such a system should also be flexible production monitoring as well as product classification component which is able to automatically detect the current state of the production and find out possible subsequent operations. Based on these requirements, Semantic Web technologies were identified as suitable for ensuring the mentioned requirement. Besides the development of the automotive domain ontology presented in this section, the utilization of this domain ontology for semantic monitoring by SCM component is introduced in Sec. 5.

As the first step during the automotive domain ontology design and development, concepts, the taxonomic relationships (generalization and specialization), and the meronymy relationships regarding the car assembly were defined and implemented. Two different car variants are described (2WD and 4WD car) together with its components, e.g., doors, windows, seats, transmission, drive shaft, etc. Figure 2 illustrates "*4WD_Car*" components together with relations to its parts.

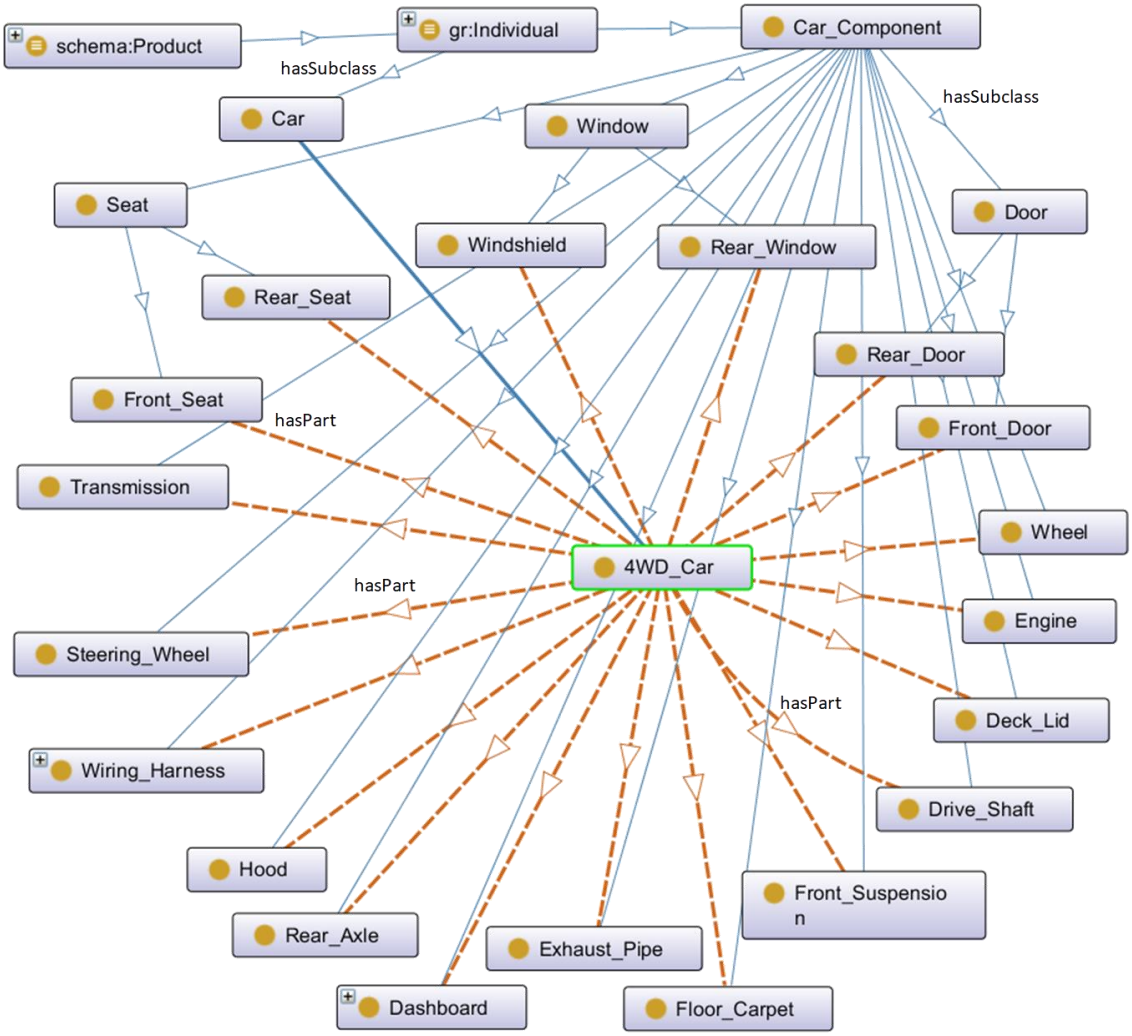

Figure 2: 4WD_Car concept and relations to its parts

Next, operations for the production of 2WD and 4WD cars were defined using DCO. Every operation has a corresponding relation to an appropriate car part as well as a relation to preceding and subsequent operations (if applicable). A part of the operations of the 4WD car assembly is illustrated in Figure 3.

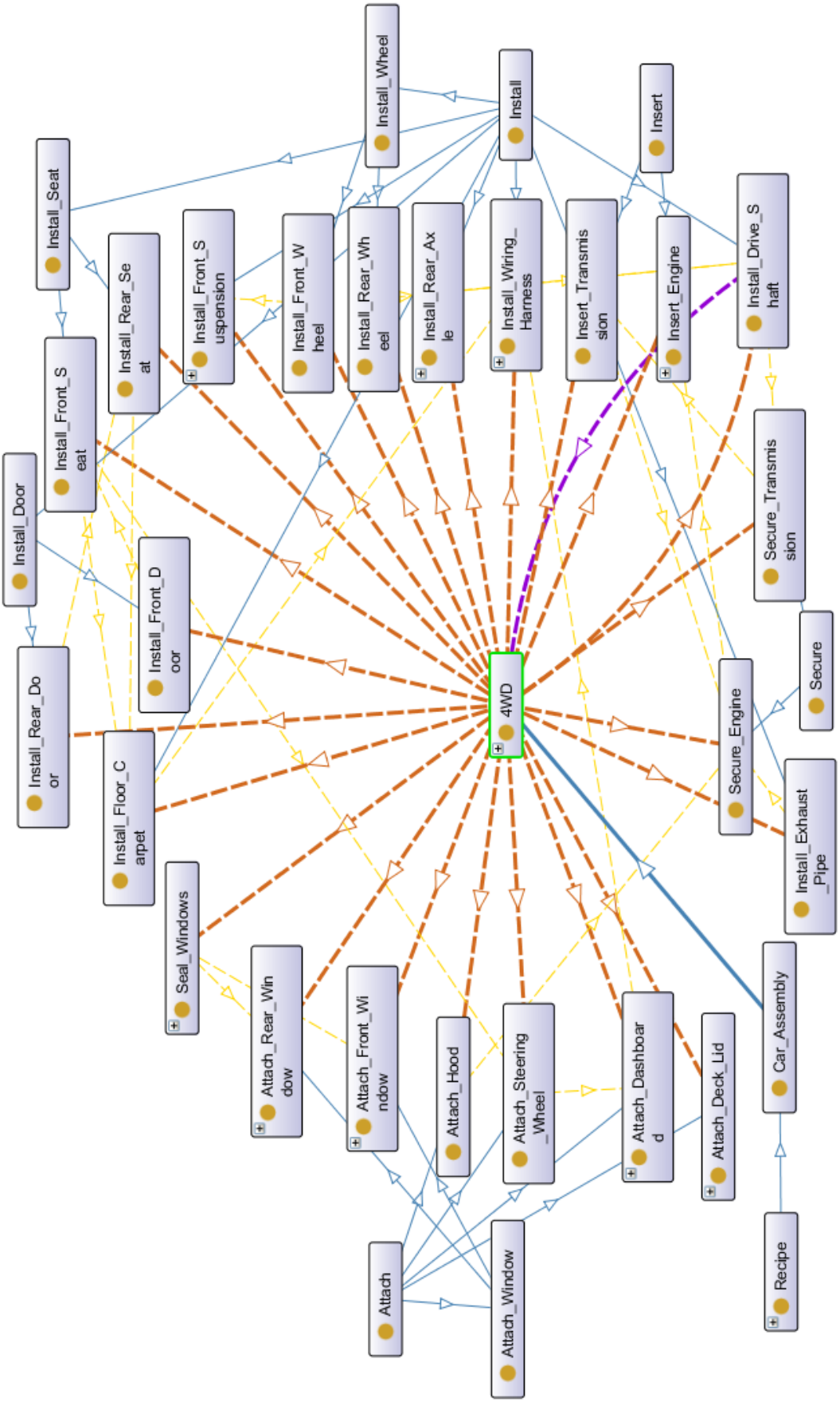

Figure 3: Operations of 4WD car assembly - Relations: hasOperation (orange color), precedingOperation (yellow color), hasSubclass (blue color)

4 Data Exchange Ontology

The complex and versatile environment such as DIGICOR platform has many various components which are focused on solutions of different specialized problems or on providing specific services. These components originate from different designers and developers. Furthermore, this highly heterogeneous nature of DIGICOR platform is especially caused by the possibility of customization or an extension of DIGICOR tool store by a company needs.

There is an effort for semantic unification of data models within the platform by DIGICOR Core Ontology (see Sec. 2). However, exploitation of the upper ontology may not be sufficient for providing effortless and faultless component integration and communication. In many cases, platform components represent isolated parts of the system, which may be influenced by various subjects, and thus the meaning of concepts may develop or change in time. Moreover, there is no possibility to guarantee, that component data models are based on DIGICOR Core Ontology due to a certain level of openness of the platform (e.g., development of a new tool by a third-party company) or due to a specific nature of a platform component function.

Different motivations and requirements for communication among DIGICOR components may be distinguished:

a) Within a contract/tender lifecycle - i.e., tendering, contracting, production planning/scheduling, production control.
b) Between various instance of the same component - e.g., across different data models/domains from different suppliers motivated by needs for flexible manufacturing and sharing of resources.

Data Exchange Ontology (DEO) was proposed and developed for facilitating interoperability among components. The aim of DEO is not unification components' APIs but a semantic description of APIs, i.e., a description of the meaning of API elements, together with their possible relations.

DEO exploits DOLCE Ontology (similarly to DCO) as its basis and serves as a complement to DCO. DEO is designed and implemented in OWL similarly to DCO.

In following paragraphs, a way, how to design and implement an ontological model of a message, which is consumed or produced by a service via API, is described followed by an sample implementation of API message models, which correspond to a communication between Tender Decomposition and Matchmaking Service (TDMS) component and Semantic Control and Monitoring (SCM) component. Furthermore, relations between different APIs will be described.

4.1 Ontological Model for API Description

The basic building block is API itself. Items of an API represent a set of information which is required by "surrounding" services for producing or consuming messages. These items may refer to an existing concept or its (data or object) property from data model of a given

component or may refer to information which was inferred from component's data model by a certain transformation because of demands of surrounding systems.

There are many different ways how to create an ontological model for API description and depends mainly on the intended exploitation of DEO. DEO OWL model is based on five main concepts - "*System*", "*Tool*", "*Service*", "*Message*", and "*Message Model Element*". These concepts together with their relations are shown in Figure 4.

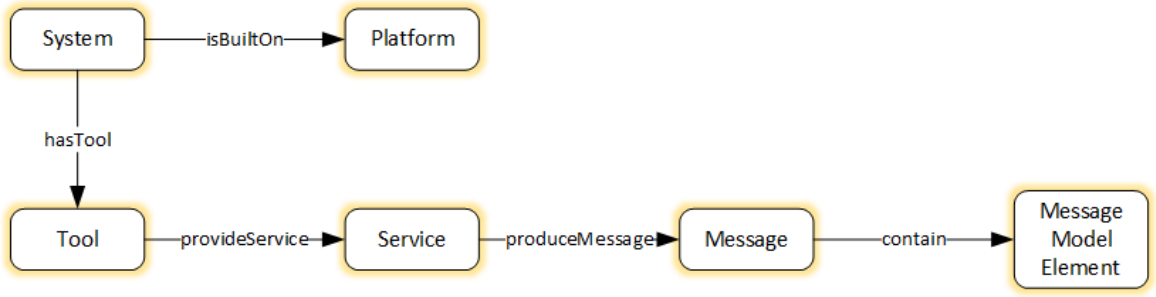

Figure 4: DEO backbone concepts

4.1.1 System Concept

The "*System*" concept describes systems or platforms which communicate with each other. The part of the DEO conceptualization related to "System" concept is naturally focused on DIGICOR platform, and therefore the concept specialization is to "DIGICOR platform" concept and "External" concept. "External" concept represents mainly suppliers and manufacturers, i.e., their systems which communicates with DIGICOR platform.

A general system is composed of various components (sometimes with various names - tools or modules). These parts form a backbone of the overall system architecture and the system ensure platform-relevant operational services for them. The most important properties of the system concept are illustrated in Figure 5.

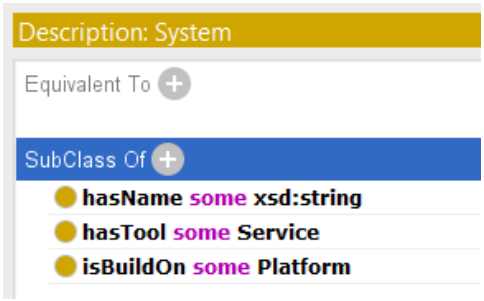

Figure 5: Properties of the system concept

4.1.1 Tool Concept

The "*Tool*" concept represents a tool or a component which has a specific purpose and provide services for fulfilling its goal or function. The characteristic feature of tools is they are very often designed and developed by various architectures and designers. Thus, the first difficulties with heterogeneous data models of APIs during communication may be observed at this level. The most important properties (from data exchange perspective) of the tool concept are illustrated in Figure 6.

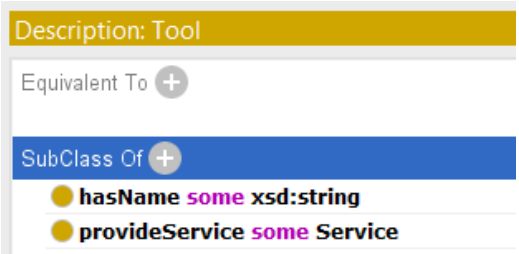

Figure 6: Properties of the tool concept

Except for the "*hasName*" and "*provideService*" properties, the tool concept may contain references to other tools or systems, which may communicate with the tool.

4.1.2 Service Concept

Communication and interaction abilities of tools or components are constituted by services. Interactions with a service are possible by messages - incoming and outcoming. From a very general point of view, an incoming message specifies a demanded action or a request for specific data together with data which are required by a service for fulfilling the demanded request. In opposite, an outcoming message contains a result of a requested action or specification of requested data.

Furthermore, services may be categorized by a different architecture on which they are built-up, e.g., Simple Object Access Protocol[4] (SOAP) or Representational State Transfer[5] (ReST). Next, services may also be naturally categorized by a corresponding system or a corresponding component. Figure 7 depicts specialized concepts of the service concept and their relations in DEO.

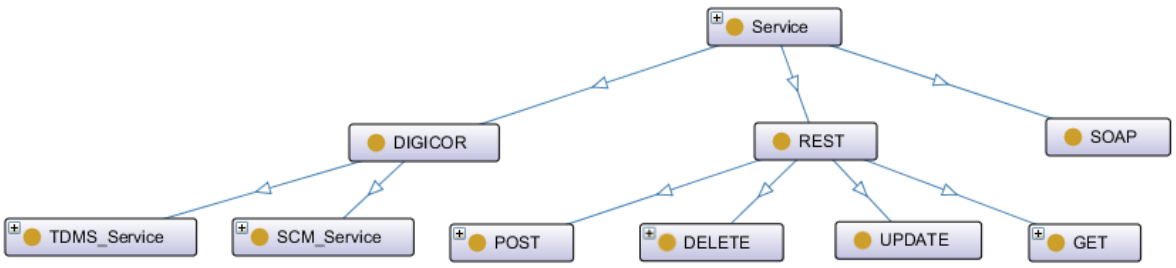

Figure 7: Specialized concepts of the service concept and their relations

One of the benefits resulting from this categorization as well as features of OWL (a service may belong to more than one concept, e.g., semantic controlling and monitoring (SCM) service and get service) is multiple inheritances of properties defined within super-concepts.

4.1.3 Message Concept

The "*Message*" concept serves as an envelope for a message body. Besides a message body represented by "*Message Model Element*" concepts, this concept comprises the id

[4] https://www.w3.org/TR/soap/

[5] https://www.w3schools.in/restful-web-services/rest-methods/

(because of the message traceability) and information about a type of message serialization (Figure 8).

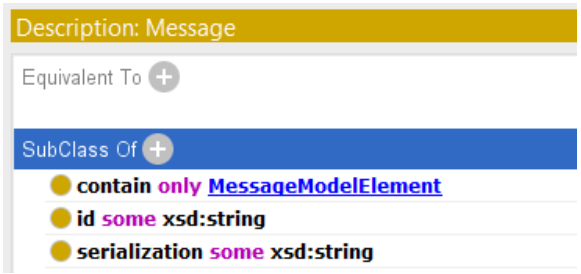

Figure 8: Properties of the message concept

4.1.4 Message Model Element Concept

The "*Message Model Element*" concept is the last of the backbone concepts of DEO. This general concept itself has no specific object or data properties, and all of the properties are defined within its specialized concepts (sub-concepts). Sub-concepts of the "Message Model Element" form together with their relations a message body.

The proper specification of the sub-concepts of the "Message Model Element" is crucial for DEO because these sub-concepts and their relations represent a previously mentioned ontological model of a service API. The problem of a transformation of one ontological model of API to another is the main problem of data exchange.

4.2 Application of DEO: TDMS Component and SCM Component Communication

In this section, the possible description of DIGICOR component (TDMS) services using DEO is introduced. Next, the formulation of a message produced by "*teamById*" service API is presented. Finally, auxiliary approaches for facilitating a new component design or better service API understanding are introduced, i.e., an overview of a given API using the explicit specification of API properties and SPARQL, differences of service APIs using SPARQL, and a transformation of a message from one service into a form of another using SWRL.

4.2.1 DEO Model of Tender Decomposition and Matchmaking Service API

Tender Decomposition and Matchmaking Service (TDMS) component is an important tool which provides valuable information for tendering and contracting process. The component can form a team for forming a requested tender based on the specification of the tender, a product decomposition, and a specification of suppliers' capabilities. A proposed team (or a list of possible teams) serves as the input for subsequent contracting. The inner ontology of TDMS is described in Sec. 6. On the other hand, the conceptualization presented in the following paragraphs represents the model of TDMS from the perspective of data

exchange. The entire TDMS ontology is not needed for the intended purpose of DEO, and even modeling of appropriate services or messages may be impossible.

TDMS as the native DIGICOR tool comply with REST architecture. In other words, it is possible to categorize TDMS services into four disjoint concepts which correspond to REST methods - GET, POST, UPDATE, DELETE. Every service is characterized within a corresponding concept by its endpoint, path, headers, and query parameters. Next, a concept of a service specifies a message which is produced or consumed by the service according to its type, e.g., GET methods typically only produce messages whereas POST methods typically consume messages. The part of the TDMS conceptualization is illustrated in Figure 9. Only GET methods, and "*TeamDescription*" message are depicted for better readability.

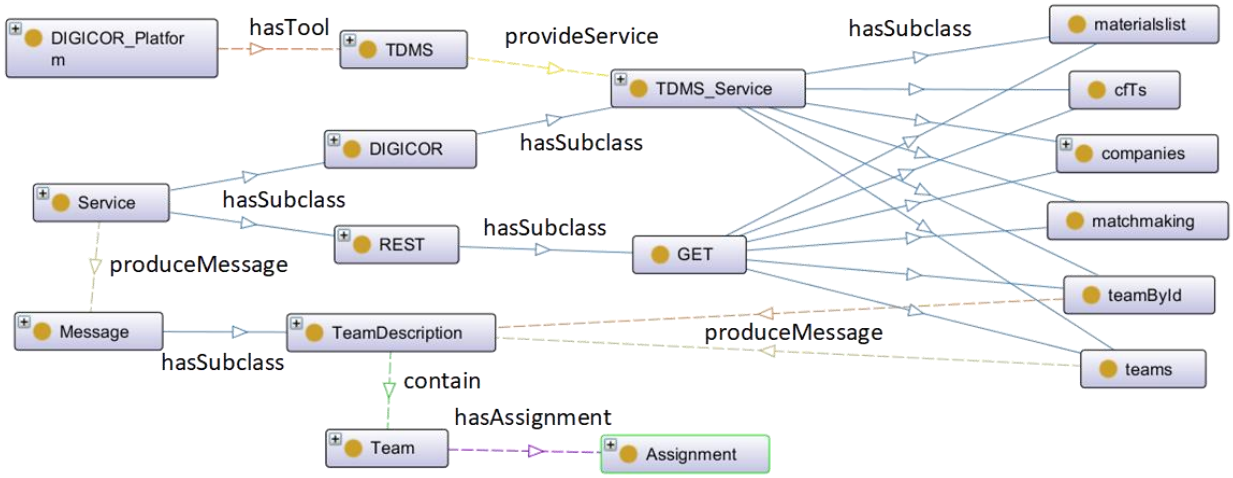

Figure 9: Part of TDMS conceptualization

4.2.2 Team Description Message

In the following paragraphs, the proposed way, how to model an API, is presented using the "*TeamDescription*" message of DEO and "*teamById*" service of TDMS. The "*teamById"* service (GET method) produces a message where a description of a team is specified related to a given tender. This service has one parameter - Team ID. The original structure of the message is illustrated in Figure 10. First, there is stated a header (a team ID and call for tender ID) followed by individual assignments with information about a given task, product, parent product, company (supplier) ID, assignment ID together with a corresponding risk score.

```
{
    "teamid": "aabbccdd",
    "cftid": "12345678",
    "assignments": [
      {
        "assignmentId": "aaabbb",
        "companyId": "111111",
        "productName": "product1",
        "parentProductName": "product2",
        "task": "DESIGN_AND_DEVELOP",
        "riskScore": "0.2"
      },
      {
        "assignmentId": "bbbccc",
        "companyId": "222222",
        "productName": "product2",
        "parentProductName": "product3",
        "task": "PLAN_AND_MANAGE",
        "riskScore": "0.4"
      }
    ]
}
```


Figure 10: Structure of response message from teamById REST service

If this message is converted into the ontological model of DEO, then the message is represented by following concepts - "*TeamDescription*" (subclass of "*Message*" concept), "*Team*" and "*Assignment*" concepts (both as subclasses of "*MessageModelElement*" concept).

The "*TeamDescription*" concept comprises an ID and a specification of message serialization. Next, it contains the reference to a body of the message, i.e., "*Team*". Properties of the concept are illustrated in Figure 11.

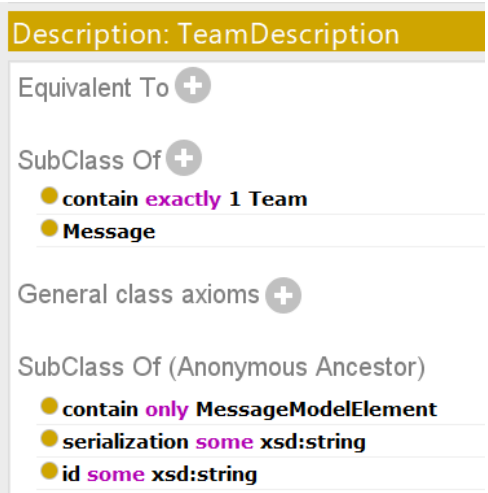

Figure 11: Properties of the TeamDescription concept

The "*Team*" concept properties are shown in Figure 12. They include a reference to a relevant call for tender, a team id, and references to particular assignments, which define team members together with their tasks.

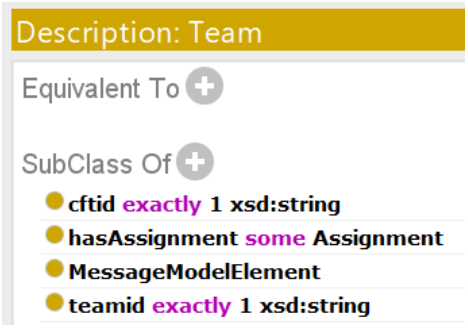

Figure 12: Properties of the Team concept

The last part of the ontological model of the "*teamById"* API is the "*Assignment*" concept. This concept keep information about a team member and its task within a given tender, i.e., ID of the assignment, ID of responsible company, a name of the product together with a name of the parent product, calculated risk score of this participant in the tender, and a specification of the task, which is expected to deliver by this team member. The corresponding properties of the "*Assignment"* concept are illustrated in Figure 13.

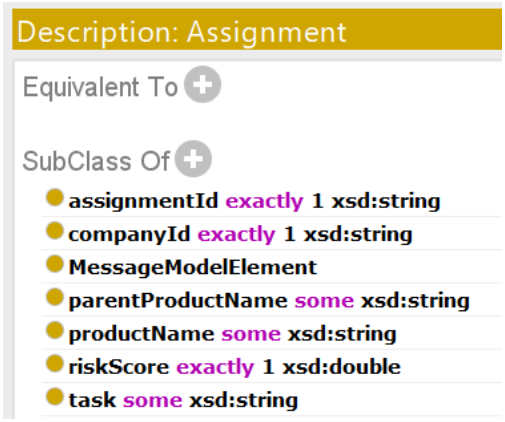

Figure 13: Properties of the Assignment concept

The aforementioned concepts provide means for formulating the team description message. Figure 14 demonstrates how individuals "*Message1*", "*Team1*", "*Assignment1*", "*Assignment2*", and "*Assignment3*" formulate a backbone of the message. The corresponding data properties of the individuals are not shown because of better readability.

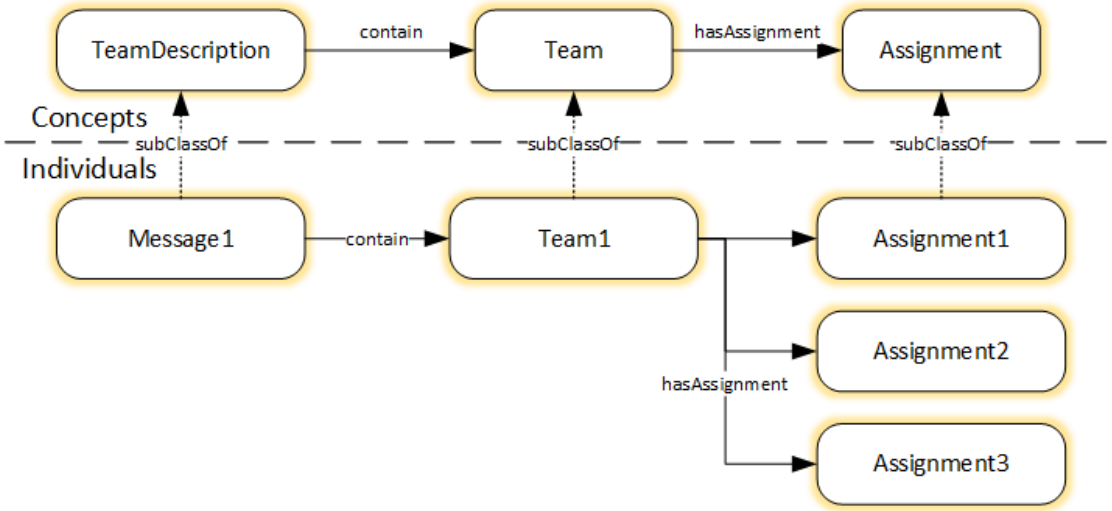

Figure 14: Model of a message produced by teamById TDMS service

4.2.3 Facilitating a Design of a new Component and Service

The design phase and subsequent development of a new component and as well as a new service may be difficult because of an insufficient description of collaborating services and thus complicating understanding of the meaning of particular API items.

A description of APIs and corresponding service messages using DIGICOR Data Exchange Ontology should help to overcome the mentioned obstacles particularly because of the following benefits resulting from utilization of Semantic Web technologies:

- Explicit specification of API and message properties including message components - often the problem for the proper understanding of an API are implicit knowledge and information of a given model within hardcoded algorithms which produce service messages. On the other hand, ontologies provide a way how to store this information in the explicit form and possibly separated from executing algorithms of services or components.
- Exploitation of a shared vocabulary for a definition of API and message properties which decrease the possible chance for a formation of a heterogeneous and defective environment.
- Easy reuse of previously developed component and algorithms because of proper understanding of their function.
- Overview of API message properties is easily achievable for example by exploitation of SPARQL[6]. A more complex example of the SPARQL utilization for finding out differences of two APIs is presented in the following paragraphs.

4.2.4 Relations between different APIs using SPARQL

In this section, the example for management (or controlling) of services interoperability is introduced. Consider two services, the first one - "*teamById*" service of TDMS component, and the second one - "*produce*" service of the SCM component.

[6] https://www.w3.org/TR/rdf-sparql-query/

Semantic Control and Monitoring (SCM) component provides several services and the most important ones are “*monitor*” and “*produce*” services. This component is responsible for controlling a production based on a specification of a demanded (sub-)product and corresponding supplier of the (sub-)product. It should be noted that this information is already stated within the “*TeamDescription*” message of TDMS “*teamById*” service. Then, SCM is able to call an executor tool and corresponding planner related to a given supplier with PDDL[7] goal because these PDDL goals are tied together with demanded (sub-)products specified in ”*ProductionDescription*” message by DIGICOR Core Ontology.

The “*ProductionDescription*” concept (representing the message) includes reference to the message body, i.e., “*DistributedProductionPlan*”. The “*DistributedProductionPlan*” CONCEPT contains order ID (from the previously mentioned contracting component) and specification of overall production goal. Next, it contains reference to individual operations. The “Operation” concept comprises information about a corresponding product name, a given supplier, and demanded task. In general, the “*TeamDescription*” message and the “*ProductionDescription*” message describes similar thing but from a different perspective - the first one, from team formation perspective, and the second one, from the production perspective. The Figure 15 illustrates the ontological model of the “*ProductionDescription*” message. Basically, the backbone of the message seems to be similar to the “*TeamDescription*” message, but they differ significantly in concept properties.

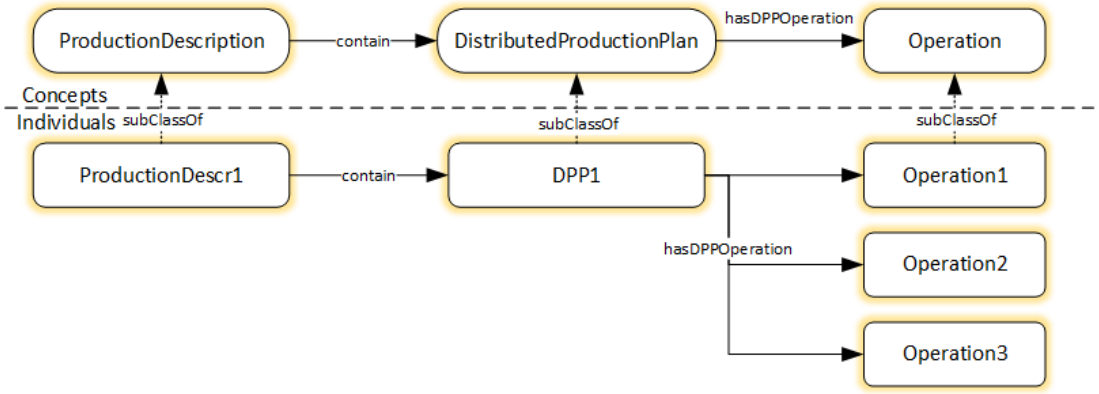

Figure 15: Model of the “ProductionDescription” message

There is no direct connection between these two services because (according to the given workflow and a character of the problem) they are connected by the other additional intermediate component. The “*teamById*” service produces a message with a specification of the new team. Next, this information about the new team serves as an input to a contracting component, which processes information about the team and demanded product. Subsequently, a contracting component may trigger a production using the SCM component, namely the “*produce*” service. The “*produce*” service consumes ”*ProductionDescription*” message. The mentioned contracting component should transform the message from TDMS in the form of SCM message supplemented by other required information.

A SPARQL query may be exploited for the identification of identical and equivalent properties of corresponding APIs. The query, as well as results of the query, are shown in Fig. X. At the bottom, there is shown the list of API1 properties (the distributed production

[7] https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language

plan properties) together with identical and equivalent properties of API2 (the team properties).

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX deo: <http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#>

SELECT distinct (?property as ?propertyOfAPI1) (?property2 as ?samePropertyInAPI2)
                (?equivalentProp as ?equivalentPropertyInAPI2)
WHERE {
            # Define variables of API concepts
            BIND (deo:Team as ?API2) .
            BIND (deo:DistributedProductionPlan as ?API1) .

            # Items of the first API
            ?API1 rdfs:subClassOf ?o .
            ?o rdf:type owl:Restriction ;
            owl:onProperty ?APIPropertyTeam .
            OPTIONAL { ?APIPropertyTeam a owl:ObjectProperty ;
                                    rdfs:range ?range .
                        ?range a owl:Class ;
                                    rdfs:subClassOf ?objectInner .
                        ?objectInner rdf:type owl:Restriction ;
                                    owl:onProperty ?APIPropertyIn .
                        }
            BIND(IF(BOUND(?APIPropertyIn) , ?APIPropertyIn , ?APIPropertyTeam ) as ?property) .

            # Items of the second API
            OPTIONAL { ?API2 rdfs:subClassOf ?o2 .
                         ?o2 rdf:type owl:Restriction ;
                                    owl:onProperty ?property .
                        }
            OPTIONAL {?API2 rdfs:subClassOf ?o3 .
                        ?o3 rdf:type owl:Restriction ;
                                    owl:onProperty ?APIProperty2 .
                        ?APIProperty2 a owl:ObjectProperty ;
                                    rdfs:range ?range2 .
                        ?range2 a owl:Class ;
                                    rdfs:subClassOf ?objectInner2 .
                        ?objectInner2 rdf:type owl:Restriction ;
                                    owl:onProperty ?property .
                        }
            BIND(IF(BOUND(?objectInner2) , ?property ,  IF(BOUND(?o2), ?property , ?null) ) as ?property2) .

            OPTIONAL {?equivalentProp owl:equivalentProperty ?property .}
}
```

propertyOfAPI1	samePropertyInAPI2	equivalentPropertyInAPI2
supplier		companyId
productName	productName	
task	task	
productionGoal		parentProductName
orderid		

Figure 16: SPARQL comparison of APIs properties

4.2.1 SWRL Transformation of a Message between different APIs

The transformation of a message between different APIs may be facilitated and (semi-)automated using A Semantic Web Rule Language[8] which is based on a combination of the OWL DL with the Unary/Binary Datalog RuleML [1]. This approach resides in a

[8] https://www.w3.org/Submission/SWRL/

formulation of a transformation SWRL rule for pairs of services where the transformation is required.

In this section, the transformation between "*TeamDescription*" message and "*ProductionDescription*" message will be introduced. Both of these messages and their properties were described in detail in previous paragraphs. The definition of the rule for the transformation is illustrated in Figure 17.

```
TeamDescription(?teamMsg)
^swrlx:makeOWLThing(?productionMsg,?teamMsg)^contain(?teamMsg,?team)
^swrlx:makeOWLThing(?DPP,?team)^hasAssignment(?team, ?assign)
^swrlx:makeOWLThing(?operation,?assign)^productName(?assign,?pn)
^task(?assign,?asgTask)^companyId(?assign,?compId)
->
ProductionDescription(?productionMsg)^DistributedProductionPlan(?DPP)
^contain(?productionMsg,?DPP)^Operation(?operation)
^hasDPPOperation(?DPP,?operation)^productName(?operation,?pn)
^task(?operation,?asgTask)^supplier(?operation,?compId)
^orderid(?DPP,"")^orderid(?DPP,"Verify orderID and production goal!")
```

Figure 17: Definition of SWRL transformation rule between "TeamDescription" message and "ProductionDescription" message

The presented rule creates for every instance of "*TeamDescription*" and its message body a corresponding instance of "*ProductionDescription*" message together with its message body according to the implemented rule logic. The last line of the rule specifies the order id of the distributed production plan. This ID as well as precise overall production goal is not specified in the "*TeamDescription*" message and have to be filled in within the contracting component which triggers SCM. Thus, the consistency violation of the distributed production plan concept is used as a user warning because of the order id and production goal verification - the order id has cardinality exactly 1, and therefore the reasoner is able to detect this violation as depicted in Figure 18.

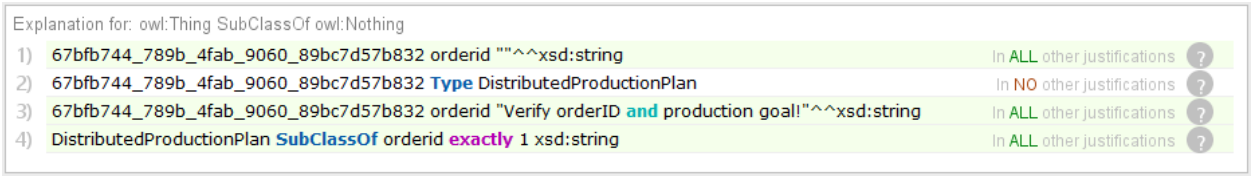

Figure 18: User warning using reasoner because of order id and production goal verification

The following figures show transformed individuals related to the "*ProductionDescription*" message - Figure 19: "*ProductionDescription*" individual, Figure 20: "*DistributedProductionPlan*" individual, and Figure 21: "*Operation*" individual (only one of the operation individual is shown because of their similarity).

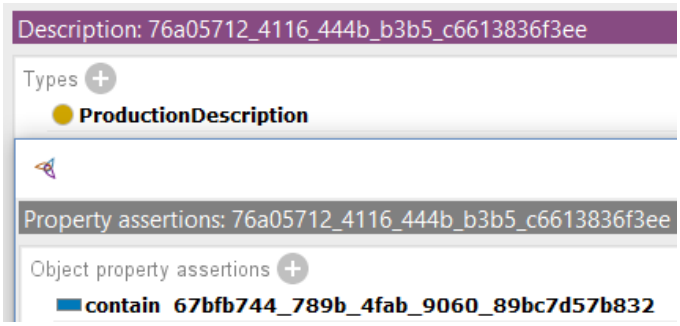

Figure 19: Generated "ProductionDescription" individual by SWRL transformation rule

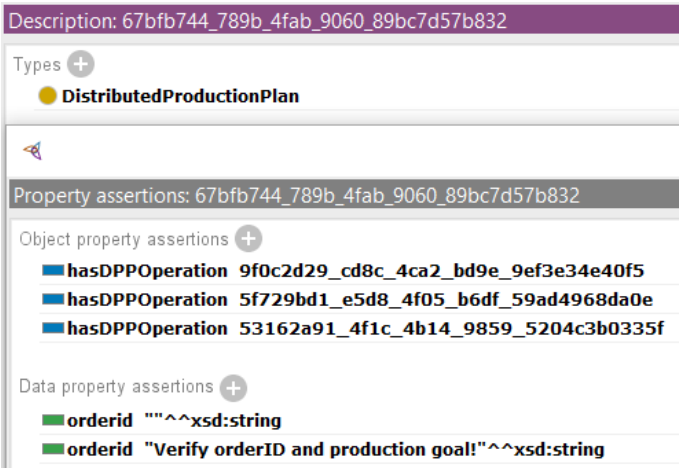

Figure 20: Generated "DistributedProductionPlan" individual by SWRL transformation rule

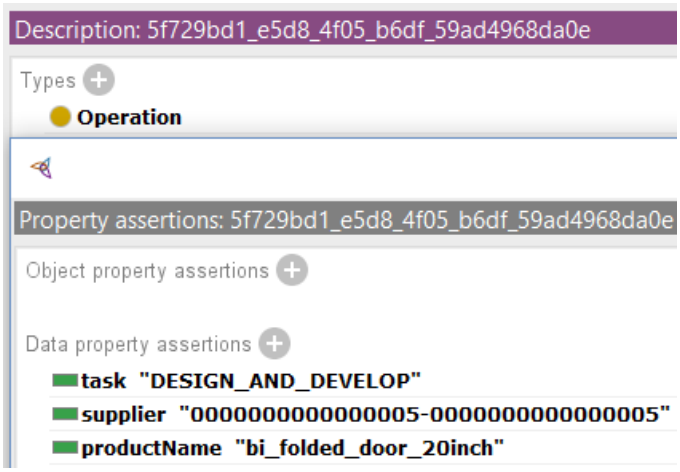

Figure 21: Generated "Operation" individual by SWRL transformation rule

The main benefit of SWRL exploitation for message transformation is the explicit specification of the transformation which may be kept outside a given component. This fact offers easier maintenance of the transformation rule as well as better change management across various services. Moreover, this approach provides consistency maintenance of the outcome message from the transformation using a reasoner.

4.3 DEO - Outlook

One of the most important subsequent steps in future research is to offer tight integration of data exchange process and DEO. This resides primarily in providing a way how to control building of messages directly by DEO. Such an ontology-based service message construction would provide many benefits including easy change management of API not only concerning given services but also across all interoperating services.

5 Utilization of Semantics within DIGICOR Tools: Semantic Control and Monitoring Tool

If a developer design and develop a DIGICOR tool then he has the possibility to exploit DIGICOR Core ontology for storage and maintenance of corresponding knowledge. In this section, several interesting benefits resulting from the utilization of Semantic Web technologies are introduced. These benefits are demonstrated with the help of Semantic Control and Monitoring tool and previously introduced automotive domain description using DCO.

SCM may be divided into two parts which are complementary to each other - the controlling and the monitoring part.

- The controlling part is responsible for the transformation of a combination of a demanded product, a required task, and a given supplier into PDDL goal. The form of PDDL goal may be for example "*finished_door_panel*" Subsequently, this PDDL goal is sent to an executer tool of a supplier.
- The monitoring part is responsible for monitoring and presentation of the current state of production. Furthermore, this part is able to derive all possible target product variants based on observations from the production.

5.1 Controlling Part of SCM Tool

The controlling part of SCM is responsible for a transformation of information about the requested product, corresponding task, and a given supplier into the form of PDDL goal which may be sent to an appropriate executor tool and subsequently into PDDL planner.

In general, the formation of PDDL goal resides in a linking of information about a requested task and a given product. For example, if the task is specified as "*Develop*" and the product is defined as "*Front Door*" then the corresponding PDDL goal is "*Developed Front Door*". However, this process cannot be fully automated in some cases because there is the direct dependency with PDDL domain and PDDL problem description, which is defined by a given supplier and serves as input data for PDDL planner. Thus, DIGICOR Core Ontology is exploited for the transformation into PDDL goal.

The new concept "*Transformation*" together with the specialized concept "*PDDL Goal*" are defined in DCO. The PDDL Goal concept has the following data properties - *task* and *goal*. They are supplemented by the object property "*isApplicableOn*", which represents a relation to a particular (sub-)product. The transformation symbolizes the linking of two different vocabularies, the first - the vocabulary of requested tasks, and the second - the vocabulary of PDDL goals related to the tasks. During the transformation process, the algorithm verifies, if there is an individual of PDDL Goal having the task data property with the same value as the requested task. Then, the algorithm verifies, if PDDL Goal is applicable to the given Product concept. Finally, if all of the conditions are fulfilled, then the value of the goal property of the PDDL Goal individual is linked with the (sub-)product name and is used as the input for the PDDL planner. The overview of the DCO part covering PDDL Goal conceptualization is illustrated in Figure 22. If there is not any suitable PDDL Goal individual, then the algorithm of SCM tool ask a user for assistance.

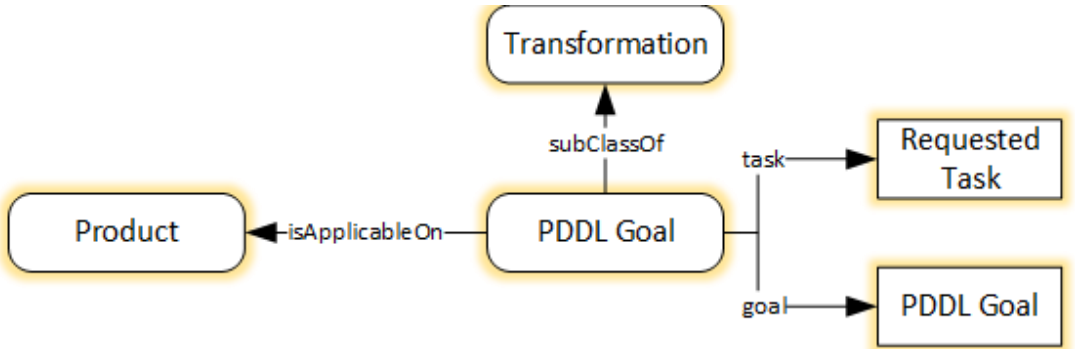

Figure 22: The DCO part covering PDDL Goal conceptualization

The presented approach for determination of PDDL goal is beneficial especially in the situation when distributed production re-planning is required because of an unexpected outage of a supplier.

5.2 Monitoring Part of SCM Tool

The first responsibility of the monitoring part of the SCM tool (monitoring and presentation of a current state of production) does not utilize Semantic Web technologies and therefore is not relevant for this document. On the other hand, the derivation of all possible target product variants based on observations from the production may easily benefit from the ontological description of products and corresponding operations.

The basic task may find out a listing of all relevant operations which are required for producing a given product using DL query and a reasoner. The sample DL query for listing all operations of 2WD car assembling is as follows:

```
Operation and isOperationOf some 2WD
```

Next, more complex task (which also demonstrates the complementing nature of SCM parts) is an acquisition of all operations of 2WD car assembling which have no preceding operations. This task is required for example at the beginning of the production for the determination of possible starting operations. This may be generalized for an application at every stage of the production concerning already conducted operations based on observations. The listing of all operations which have no preceding operations is depicted in Figure 23.

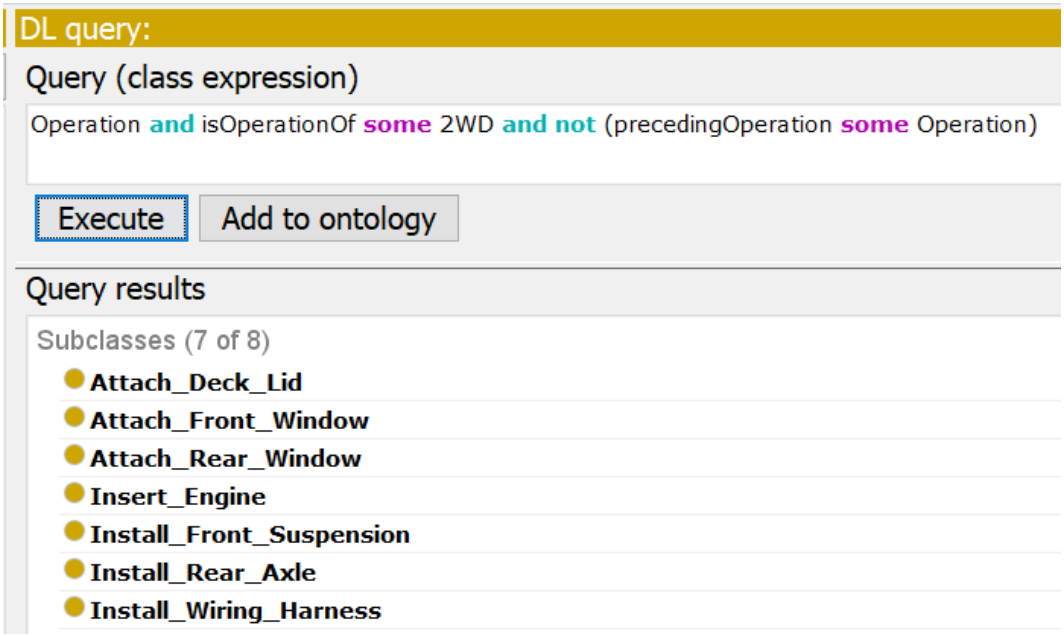

Figure 23: Listing of all operations of 2WD car assembling which have no preceding operations

Finally, the most interesting and most complex task is monitoring/classification of a given production using observations from suppliers. This information of conducted operations as well as not achievable operations (for example because of a supplier outage) are exploited for DL query formulation. The list of all possible final products is the result of such a query. Figure 24 illustrates a discovery of possible target products based on information that four wheels were already installed and installing of a drive shaft is unavailable.

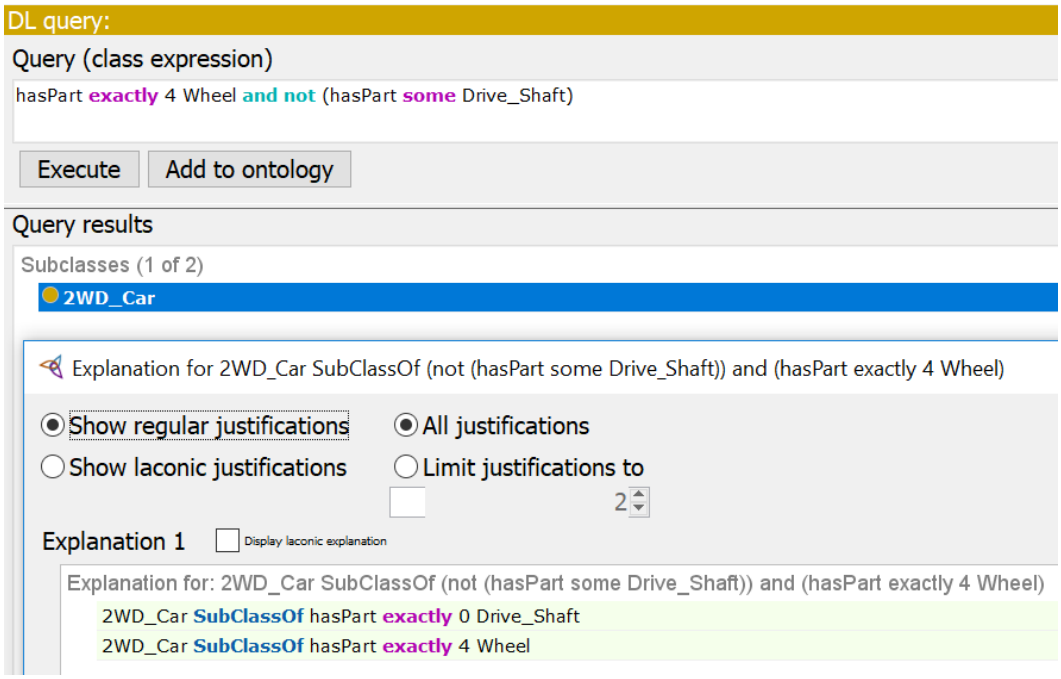

Figure 24: DL query for finding out all possible finish products based on observations from a production

6 Collaboration Ontology for TDMS and DigiGov services

The abstract design of the collaboration ontology is presented in deliverable D3.5. The present deliverable extends that ontology to accommodate functionality of the DIGICOR services that deal with formation of supplier collaborations (Tender Decomposition and Matchmaking Service, or TDMS), and definition and monitoring of governance within such collaborations (DigiGov Service). Specifically, the TDMS functionality requires a semantic model for representing, on the one hand, call-for-tender requirements, and, on the other, capabilities of the companies registered on the platform — in order to match demand with supply and thus propose prospective supplier collaborations. The DigiGov functionality requires, among others, a semantic model for representing various roles to be assumed by companies and persons within supplier collaborations. The following subsections outline the principal elements of such models, as implemented in the collaboration ontology.

6.1 Basic exposition of the ontological model for products

Deliverable D4.11 outlines the concept and basic functionality of the TDMS and presents an example of representing company capabilities in the ontology. This section describes in a greater detail the ontology representation of the so-called *specialty capabilities* of the companies — which indicates the products they can deal with and tasks they can perform on them.

Specifically, a specialty capability is defined as a pair *item–goal*, where *item* is either a specific product or a product category, and *goal* — one of *Plan&Manage*, *Design&Develop*, *IntegrateDesign*, *Source*, *Make*, *Assemble*, and *Deliver*. In ontological terms, every specific product represents an individual of a certain product class — that is intended to describe a certain category of products. Classes themselves are organised in a hierarchical fashion: every product class may have one or more subclasses, so that the resulting class hierarchy represents a classification of products in categories, sub-categories, sub-sub-categories, and so forth. Any product class may (but does not need to) have individual products as its members, which is represented by the relationship ‘has individual’ (see Figure 25). This kind of relationship between product classes and specific products is essential for matchmaking, as it allows (i) the companies to specify their capabilities in terms of broader categories than just specific product variants, and (ii) the TDMS to perform an approximate matching of the tender requirements against companies’ capabilities. As a result, when the requested item–goal task cannot be fulfilled by any of the companies then the search for suppliers is progressively expanded to include companies capable of dealing with the products of the same class, or that class as a whole, or its parent classes.

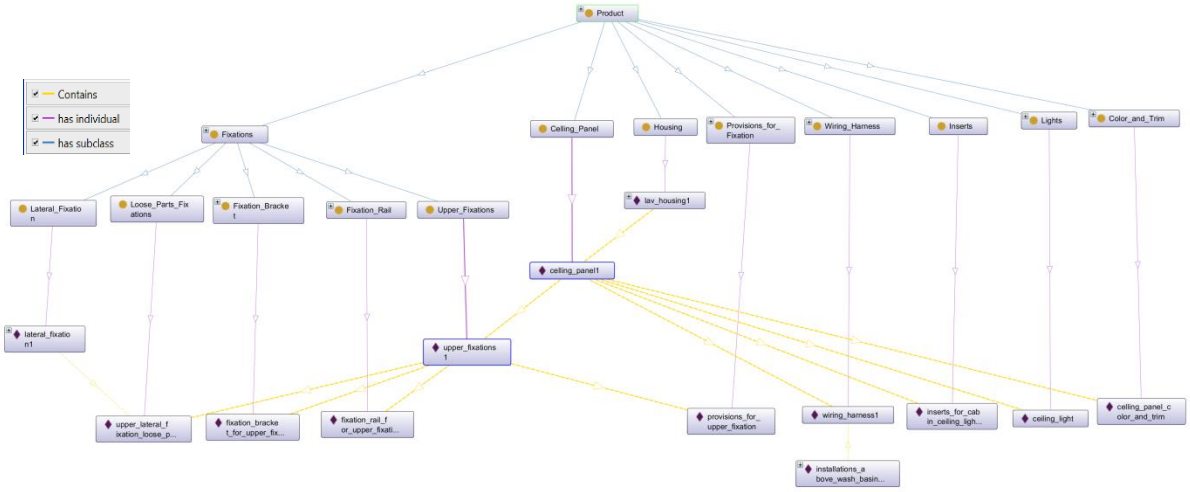

Figure 25: Example of relationships between products and product classes in the Collaboration Ontology.

Apart from that, a third kind of relationship is intended to relate individual products among themselves, as follows: if one product is an immediate component of another, as per the product structure of the latter, then the latter product is related to the former by means of the relationship 'contains'. Using this kind of relationship, the ontology is capable to define all products in terms of their structure, which is essential for finding prospective supplier collaborations that would be able to provide the requested product by making the components and assembling them to the final product — for example, when none of the companies registered on the platform is capable of providing the product on its own. These three kinds of relationships are illustrated in Figure 1: the product class hierarchy involves `Product` as the root class, which has `Fixations` as one of its subclasses. That subclass has, in turn, its own subclasses — such as `Lateral_Fixation` and `Upper_Fixations`. `celling_panel1` represents a specific product that is an individual of the class `Celling_Panel`. It contains `upper_fixations1` and other products as its immediate components — which belong to their own product classes. In turn, `upper_fixations1` contains its own components — such as `provisions_for upper_fixations`, and others.

The TDMS provides an API for other DIGICOR services to access product structures and product classification in terms of categories. For example, the Company Service can request the list of root products (which are not contained in any other product) as well as the hierarchical structure of every such product and display them in a graphical form to let the user specify his or her company capabilities in terms of the specific products. In the same way, the Tender Service can request this information to let the user specify the target product in a call for tenders. The specific API call for obtaining the list of root products is

https://tdms.digicor-platform.eu/rootproducts

which returns a comma-separated list of product names. The specific API call for obtaining the product structure of a specific product (not necessarily a root one) is

https://tdms.digicor-platform.eu/treeStructure/{Product}

where {Product} has to be substituted with the required product name. This call returns a JSON data structure representing the product structure in a hierarchical form. Figure 26shows the result of executing the above call for a root product called `pre_assembled_and_tested_lavatory_module_1`. The same call equally works for non-root products — see, for example:

https://tdms.digicor-platform.eu/treeStructure/celling_panel1

Figure 26: Example of a product structure provided by the TDMS service through an API call.

6.2 Collaboration ontology model for implementing selected governance rules

Flexible design of the collaboration ontology supports creation and expansion of vocabularies (i.e. classes and individuals) for any service embedded in the DIGICOR platform. Communication between DIGICOR services is based on an event-driven service-oriented architecture (EDSOA), so that each service publishes events and other services listen to these, which promotes loose coupling between services [2]. As previously indicated, DigiGov is a service that supports definition and monitoring of governance within supplier collaborations. The governance is defined in terms of specific rules that may stem from a number of different sources — such as industry policies, rules specified by a call for tenders, and rules agreed between the partners. The collaboration ontology is extended with classes

to accommodate definition of such governance rules. The Actor, Role and Event are the examples of such classes (see Figure 27).

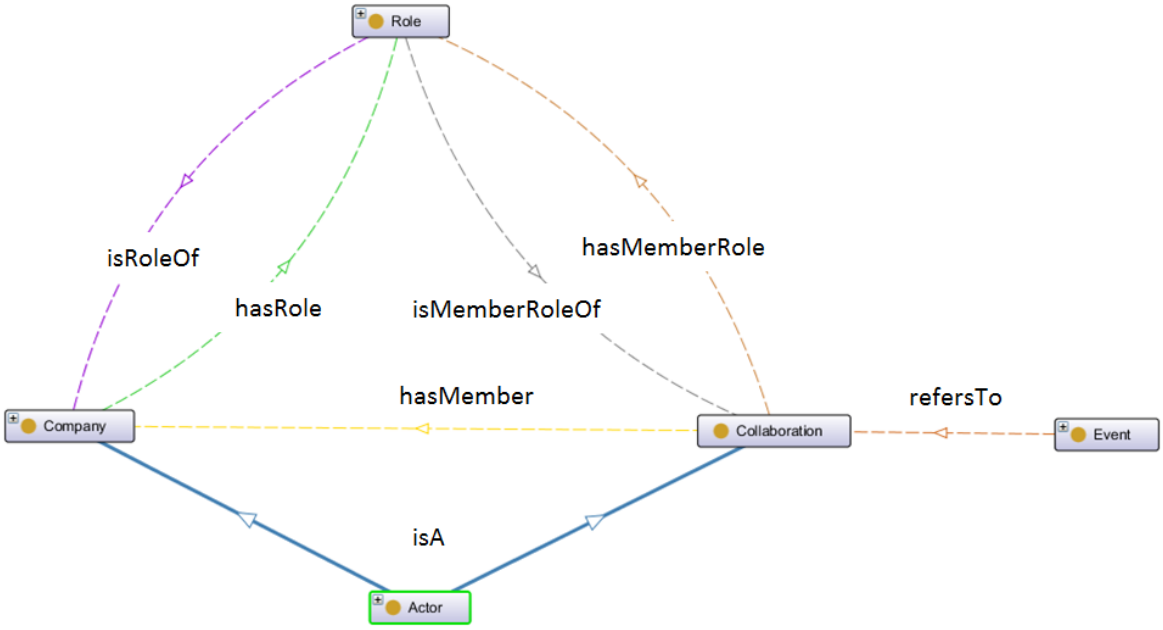

Figure 27: Class relationship diagram.

Figure 3: Class relationship diagram.

The Actor class is extended with the Company and Collaboration classes and has an ‘isA’ relationship with them. When an event is published then it refers to a specific class in the collaboration ontology. For instance, a CollaborationCreated event refers to the Collaboration class that has an association with a member company. A company has a Role (e.g. Network Coach); hence it has a relationship with the Role class via the hasRole property — which is the inverse property of isRoleOf. In a similar way, a role (e.g. Network Coach) is a member company’s role that has association with the given collaboration — which is, in turn, inversely linked with that role via the hasMember role property. Figure 28 is demonstrating the same concepts using instances of the classes — to which the DigiGov service refers when checking governance rules in actual collaborations in response to the incoming events.

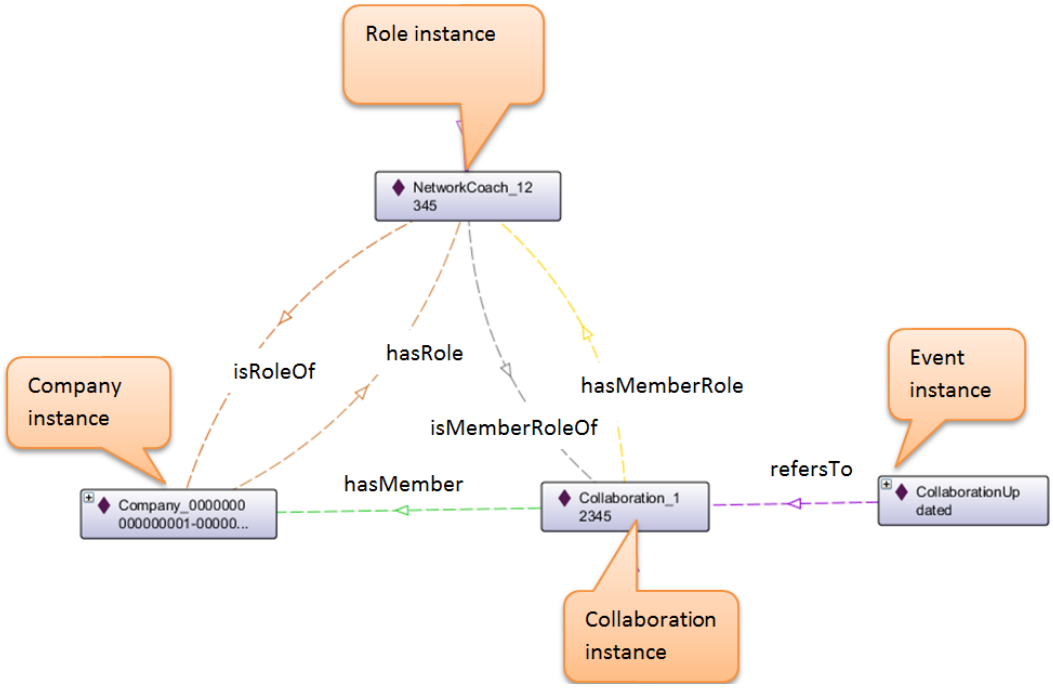

Figure 28: Instance relationship diagram.

Figure 29 illustrates definition of a governance rule that requires from any running collaboration to have a member company with the Auditor role. The user is able to select the **event** type, in response to which the rule has to be checked, and specify a **condition** on classes and their attributes (Collaboration.status = RUNNING and Collaboration.Member.Role.Auditor ≠ BLANK). Violation of the condition triggers an **action**: sending an alert to the facilitator company of that specific collaboration. The governance rule is translated into the Drools language (as shown in the box at the bottom of Figure 5). The Drools engine applies that rule to all actual collaborations.

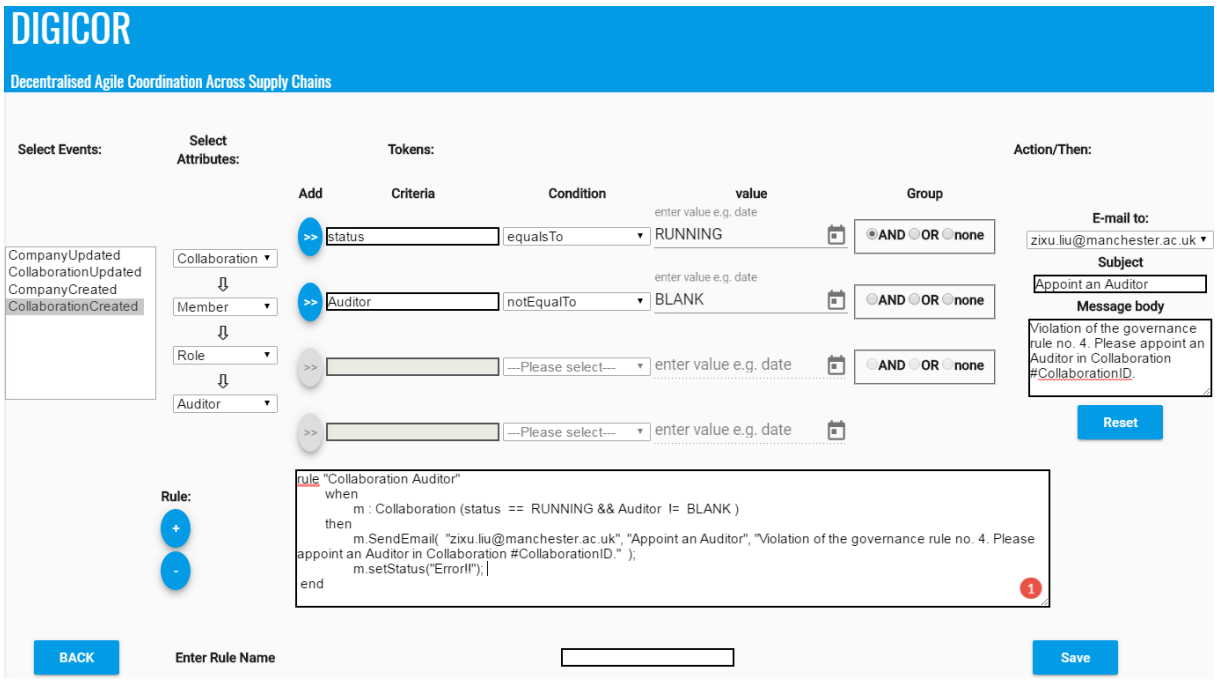

Figure 29: Definition of a governance rule

7 Collaboration policies

A fundamental aspect of DIGICOR collaborative projects is the exchange of documents between partners. In sales projects this leads to the joint submission of a tender, whereas in development projects it leads to the creation of the final product design.

Access to and usage of the exchanged documents is regulated by means of semantic attribute-based collaboration policies, i.e., rules defined over attributes that actors and resources bear. These rules can be set throughout a project's lifecycle via the collaborations' administration panel, ensuring that partners' intellectual property is protected, as well as, that partners have access only to the resources they need to.

In the context of a given collaboration, each company, as part of a supply chain, undertakes specific assignments. Such assignments are reflected by the combination of attributes "Task" and "Product"; thus, in the case where a company is assigned, possibly among others, with the task of building a plane, we assume that this company holds a role that is further characterised by attributes "Task" and "Product". Similarly, collaboration documents concern specific tasks and products, thus being also further described through the same attributes. For example, a company assigned with the task to build the wings of a plane, should also provide the documentation on the Work Breakdown Structure for building it.

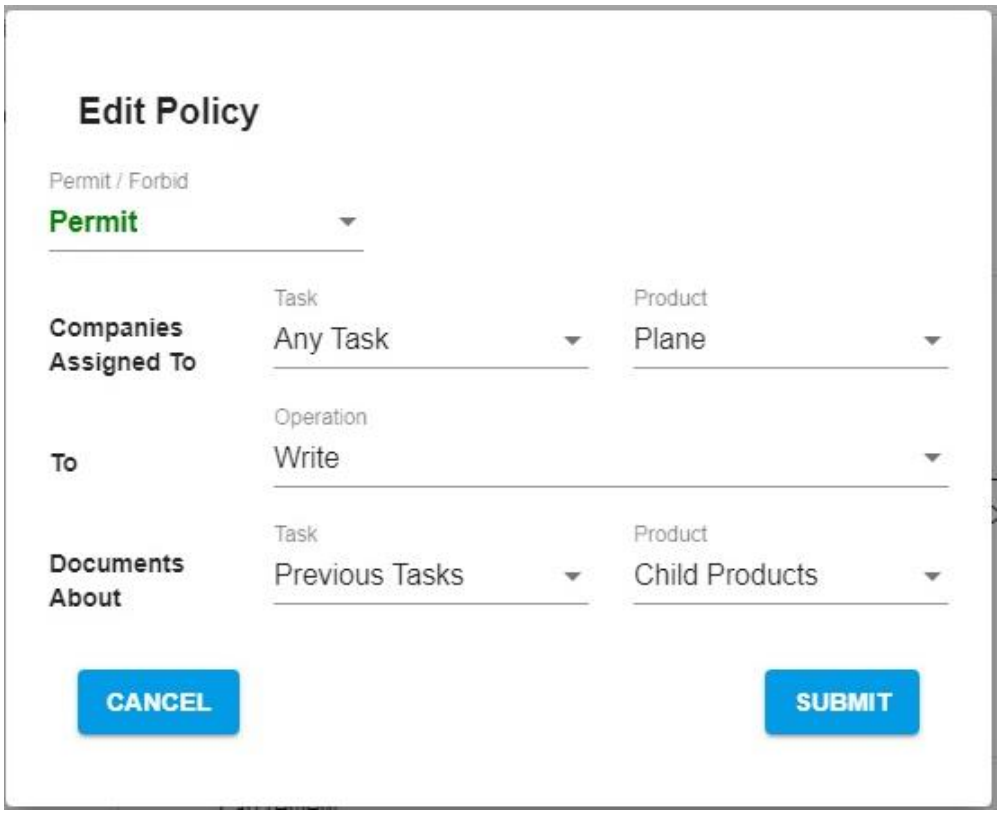

Figure 30 User interface for collaboration policy definition

In Figure 30 an example of a collaboration policy is illustrated, prescribing that companies with assignments pertaining to task "AnyTask" and product "Plane" are allowed to perform operation "Read" on documents about tasks characterised by the relation "Previous Tasks"

with respect to the actor's task, and about products being "Child Products" of the actor's product.

In other words, attribute based collaboration rules are centred around tasks, goals and their relations, while their definition may involve specific attribute values of actor and resource (e.g., Task = "AnyTask)", or definition of resource constraints related to the ones of the actor (e.g., Task = "Previous Tasks", referring to tasks that precede the actor's task).

The UI of Figure 30 enables the definition of the collaboration policies with increasing user-friendliness, hiding the complexity of the underlying ontological rules expressed by means of the entities of the Policy Model Ontology (PMO) illustrated in Figure 31.

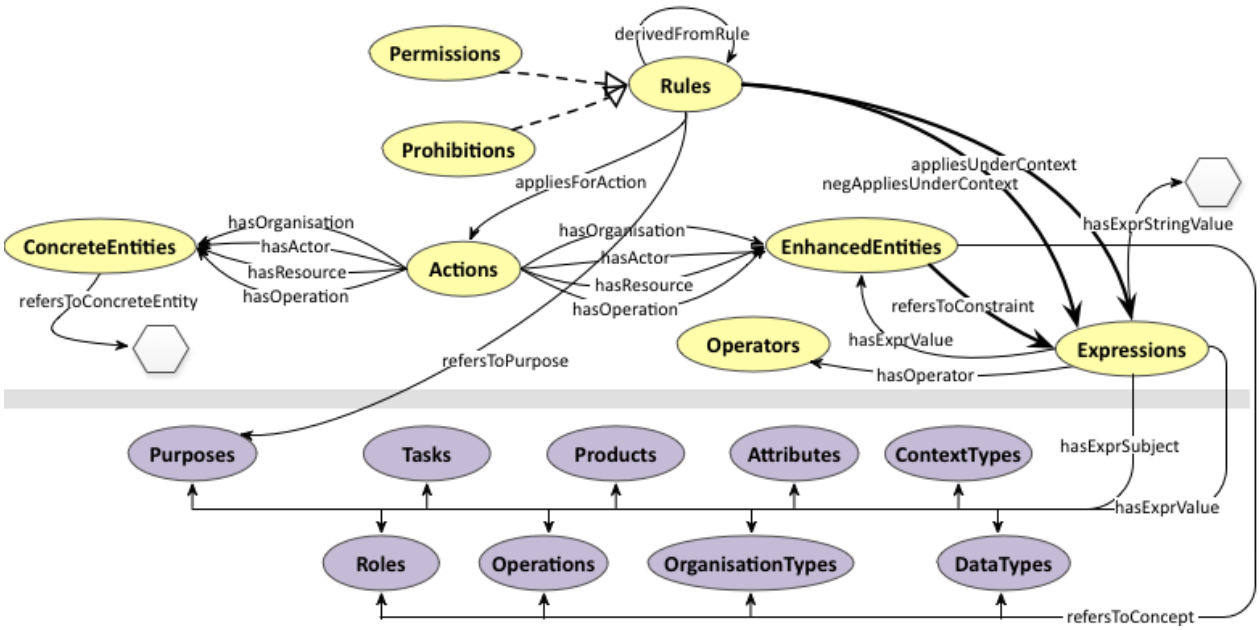

Figure 31 Semantic Policy Model (D3.3)

Moving forward with the example, the collaboration policy of Figure 30 will lead to the definition of a `Permissions` class instance grounded upon an action, which is comprised by an actor, an operation and a resource. In this case the action elements are the following:

Actor:	`AnyCollaborationRole`[9] WHERE (`Task EQUALS AnyTask` **AND** `Product EQUALS Plane`)
`AnyCollaborationRole` constitutes a `Roles` class instance, `Task` and `Product` are `Attributes` class instances, `EQUALS` an `Operators` class instance, `AnyTask` a `Tasks` class instance and `Plane` a `Products` class instance	
Operation:	`Write`
`Write` is an `Operations` class instance	
Resource:	`Document` WHERE (`Task PRECEDES` *Actor*.`Task` **AND** `Product IS_PART_OF` *Actor*.`Product`)

[9] Collaboration policies in DIGICOR are role- and datatype- agnostic, meaning that only the semantic values of attributes Task and Product matter and not the semantic type itself of actor and resource.

Document constitutes a DataTypes class instance, Task and Product are Attributes class instances, PRECEDES and IS_PART_OF are Operators class instances

It should be noted here that in the current example, constraints upon the resource are not defined in an absolute way, but in relation to the attribute values of the actor, by leveraging the appropriate *Actor* variable.

Actor, operation and resource are defined as instances of the EnhancedEntities class, which allow the definition of constraints upon attributes of the associated roles, operations, datatypes, etc. That is, WHERE clauses are defined as instances of the Expressions class, with Task constituting an attribute of both a collaboration role and a document. In addition, logical relations of constraints are also supported (e.g., Task EQUALS AnyTask **AND** Product EQUALS Plane).

The policy of our example is generic and, in order for it to be enforced, meta-rules concerning more specific concepts and absolute attribute values have to be generated. In that respect, the insertion of a rule in the policy engine through the UI of Figure 31 will be followed by a two-level meta-rules extraction.

First, meta-rules will be extracted through inheritance across the actor's, operation's and resource's corresponding semantic hierarchies; for instance, a rule defined for operation AnyOperation will be inherited to more specific operations of the Operations graph of Figure 32, i.e., operations related to AnyOperation through the isA relation, thus leading to the generation of five meta-rules.

The second phase concerns the extraction of meta-rules according to attribute hierarchies. Based on the graphs of Figure 32, rules will be inherited across the Tasks and Products hierarchies by means of the isA, isPartOf and precedes object properties (the inverse ones are also defined although not illustrated in Figure 32); isA models the generalisation/specialization of a concept (a relation met in all graphs of the information model), precedes is used to denote that a task precedes another task (Build task precedes Deliver task) and isPartOf is leveraged for denoting inclusion of a product into another one (a plane contains lavatory and seats).

In the context of collaboration policies definition, isA relation enables the definition of generic rules that will be inherited from high level concepts to more specific ones, while precedes, succeeds and isPartOf relations allow the definition of *relative* constraints based on the *Actor* variable.

Thus, after the two-level meta-rules extraction process, the following more specific permissions will apply for the current collaborative project:

a) Because of the Design isA AnyTask, Lavatory isPartOf Plane, Seat isPartOf Plane relations (we assume that tasks with no previous tasks precede themselves):

Rule 1:	Permit Companies assigned to Design Plane to Write Documents about Design Lavatory
Rule 2:	Permit Companies assigned to Design Plane to Write Documents about Design Seat

b) Because of the `Build isA AnyTask, Design precedes Build, Lavatory isPartOf Plane, Seat isPartOf Plane` relations:

Rule 3:	Permit Companies assigned to `Build Plane` to `Write` Documents about `Design Lavatory`
Rule 4:	Permit Companies assigned to `Build Plane` to `Write` Documents about `Design Seat`

c) Because of the `Deliver isA AnyTask, Deliver precedes Design, Deliver precedes Build` (implicit relation) `Lavatory isPartOf Plane, Seat isPartOf Plane` relations:

Rule 5:	Permit Companies assigned to `Deliver Plane` to `Write` Documents about `Design Lavatory`
Rule 6:	Permit Companies assigned to `Deliver Plane` to `Write` Documents about `Design Seat`
Rule 7:	Permit Companies assigned to `Deliver Plane` to `Write` Documents about `Build Lavatory`
Rule 8:	Permit Companies assigned to `Deliver Plane` to `Write` Documents about `Build Seat`

It is noted that meta-rules are associated to the initial collaborative rule through the `derivedFromRule` property, so that changes to the initial rule will be propagated to all the associated ones.

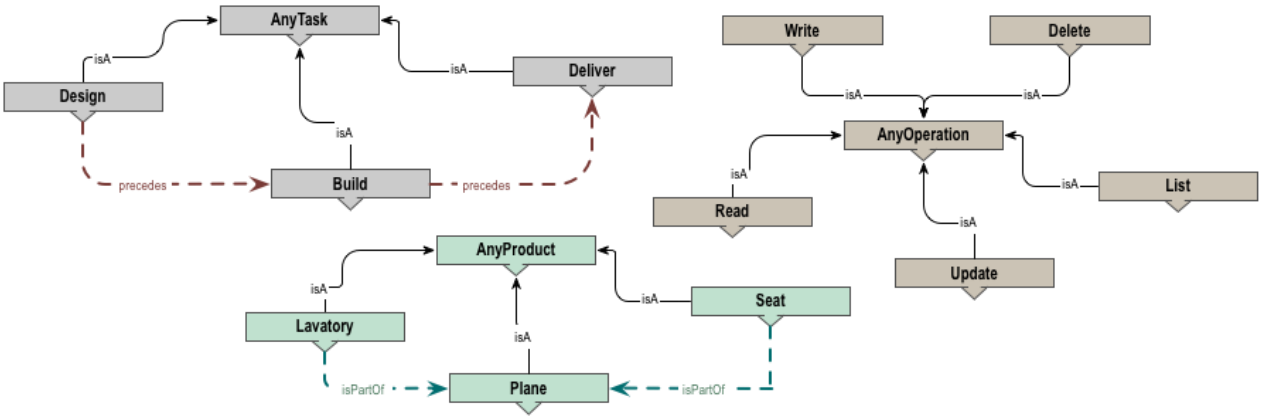

Figure 32 Information Model internal hierarchies: `Tasks`, `Products` and `Operations` instances

For the management of the base information model entities (i.e., addition, deletion, list and update of tasks, operations and products, graph traversal) the appropriate InformationModel API has been implemented, allowing to either pre-populate the underlying ontology or to expose a UI for its per case creation and management by the collaboration users themselves. Similarly, the Rules API is leveraged for the management of collaboration rules.

References

[1] Boley, Harold. "The RuleML family of web rule languages." International Workshop on Principles and Practice of Semantic Web Reasoning. Springer, Berlin, Heidelberg, 2006

[2] Yang Zhang, Li Duan and Jun Liang Chen, 2014, Event-Driven SOA for IoT Services, '14 Proceedings of the 2014 IEEE International Conference on Services Computing, Pages 629-636.

Annex A: History

Version	Date	Changes
A01	**2019-03-15**	**First draft**
A02	**2019-03-27**	**Internal review draft**
A03	**2019-03-29**	**Final draft**

.

Annex B: Part of presented Data Exchange Ontology together with SWRL rules and generated axioms

@prefix : <http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix xml: <http://www.w3.org/XML/1998/namespace> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@base <http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO> .

<http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO> rdf:type owl:Ontology ;

rdfs:comment "Data exchange ontology - describing APIs of DIGICOR components/services within the paltform for facilitating their interoperability"@en .

###

Annotation properties

###

http://swrl.stanford.edu/ontologies/3.3/swrla.owl#isRuleEnabled

<http://swrl.stanford.edu/ontologies/3.3/swrla.owl#isRuleEnabled> rdf:type owl:AnnotationProperty .

###

Object Properties

###

http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#consumeMessage

:consumeMessage rdf:type owl:ObjectProperty .

http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#contain

```
:contain rdf:type owl:ObjectProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#hasAssignment
:hasAssignment rdf:type owl:ObjectProperty ;
               rdfs:domain :Team ;
               rdfs:range :Assignment .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#hasDPPOperation
:hasDPPOperation rdf:type owl:ObjectProperty ;
                 rdfs:domain :DistributedProductionPlan ;
                 rdfs:range :Operation .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#hasResponse
:hasResponse rdf:type owl:ObjectProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#hasTool
:hasTool rdf:type owl:ObjectProperty ;
         rdfs:domain :DIGICOR_Platform ;
         rdfs:range :Tool .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#isBuiltOn
:isBuiltOn rdf:type owl:ObjectProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#produceMessage
:produceMessage rdf:type owl:ObjectProperty .
```

```
### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#provideService
:provideService rdf:type owl:ObjectProperty ;
            rdfs:domain :Tool ;
            rdfs:range :Service .


#################################################################
#    Data properties
#################################################################

### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#ATA
:ATA rdf:type owl:DatatypeProperty .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#AnnualTurnOver
:AnnualTurnOver rdf:type owl:DatatypeProperty ;
            rdfs:subPropertyOf owl:topDataProperty .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Certification
:Certification rdf:type owl:DatatypeProperty .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#ContractType
:ContractType rdf:type owl:DatatypeProperty .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Department
:Department rdf:type owl:DatatypeProperty .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Location
:Location rdf:type owl:DatatypeProperty .
```

```
### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Material
:Material rdf:type owl:DatatypeProperty .

### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#NumberOfEmployees
:NumberOfEmployees rdf:type owl:DatatypeProperty .

### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Speciality
:Speciality rdf:type owl:DatatypeProperty .

### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#TargetRegion
:TargetRegion rdf:type owl:DatatypeProperty .

### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Technology
:Technology rdf:type owl:DatatypeProperty .

### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#acceptCurrencyNegotiation
:acceptCurrencyNegotiation rdf:type owl:DatatypeProperty ;
                           rdfs:subPropertyOf owl:topDataProperty .

### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#assignmentId
:assignmentId rdf:type owl:DatatypeProperty ;
              rdfs:subPropertyOf owl:topDataProperty .

### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#assignmentProperties
```

```
:assignmentProperties rdf:type owl:DatatypeProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#cftid
:cftid rdf:type owl:DatatypeProperty ;
      rdfs:subPropertyOf owl:topDataProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#company
:company rdf:type owl:DatatypeProperty ;
        rdfs:comment "Company name"@en .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#companyId
:companyId rdf:type owl:DatatypeProperty ;
          owl:equivalentProperty :supplier ;
          rdfs:subPropertyOf owl:topDataProperty ;
          rdfs:isDefinedBy :Company .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#companyProperties
:companyProperties rdf:type owl:DatatypeProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#endpoint
:endpoint rdf:type owl:DatatypeProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#hasName
:hasName rdf:type owl:DatatypeProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#headers
```

```
:headers rdf:type owl:DatatypeProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#id
:id rdf:type owl:DatatypeProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#orderid
:orderid rdf:type owl:DatatypeProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#parentProductName
:parentProductName rdf:type owl:DatatypeProperty ;
                   owl:equivalentProperty :productionGoal ;
                   rdfs:subPropertyOf owl:topDataProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#path
:path rdf:type owl:DatatypeProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#productName
:productName rdf:type owl:DatatypeProperty ;
             rdfs:subPropertyOf owl:topDataProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#productionGoal
:productionGoal rdf:type owl:DatatypeProperty ;
                rdfs:subPropertyOf owl:topDataProperty .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#queryParameter
:queryParameter rdf:type owl:DatatypeProperty .
```

```
### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#risk
:risk rdf:type owl:DatatypeProperty .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#riskScore
:riskScore rdf:type owl:DatatypeProperty ;
          rdfs:subPropertyOf owl:topDataProperty .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#serialization
:serialization rdf:type owl:DatatypeProperty .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#shortDescription
:shortDescription rdf:type owl:DatatypeProperty ;
                  rdfs:comment "Short description of the company"@en .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#supplier
:supplier rdf:type owl:DatatypeProperty ;
         rdfs:subPropertyOf owl:topDataProperty .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#task
:task rdf:type owl:DatatypeProperty ;
     rdfs:subPropertyOf owl:topDataProperty .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#teamid
:teamid rdf:type owl:DatatypeProperty ;
       rdfs:subPropertyOf owl:topDataProperty .
```

```
#################################################################
#    Classes
#################################################################

###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Assignment
:Assignment rdf:type owl:Class ;
        rdfs:subClassOf :MessageModelElement ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :parentProductName ;
                          owl:someValuesFrom xsd:string
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :productName ;
                          owl:someValuesFrom xsd:string
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :task ;
                          owl:someValuesFrom xsd:string
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :assignmentId ;
                          owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                          owl:onDataRange xsd:string
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :companyId ;
                          owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                          owl:onDataRange xsd:string
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :riskScore ;
```

```
                owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                owl:onDataRange xsd:double
               ] .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Company
:Company rdf:type owl:Class ;
        rdfs:subClassOf :MessageModelElement ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :ATA ;
                          owl:someValuesFrom xsd:string
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :AnnualTurnOver ;
                          owl:someValuesFrom xsd:string
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :Certification ;
                          owl:someValuesFrom xsd:string
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :ContractType ;
                          owl:someValuesFrom xsd:string
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :Department ;
                          owl:someValuesFrom xsd:string
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :Location ;
                          owl:someValuesFrom xsd:string
                        ] ,
                        [ rdf:type owl:Restriction ;
```

```
  owl:onProperty :Material ;
  owl:someValuesFrom xsd:string
] ,
[ rdf:type owl:Restriction ;
  owl:onProperty :NumberOfEmployees ;
  owl:someValuesFrom xsd:string
] ,
[ rdf:type owl:Restriction ;
  owl:onProperty :Speciality ;
  owl:someValuesFrom xsd:string
] ,
[ rdf:type owl:Restriction ;
  owl:onProperty :TargetRegion ;
  owl:someValuesFrom xsd:string
] ,
[ rdf:type owl:Restriction ;
  owl:onProperty :Technology ;
  owl:someValuesFrom xsd:string
] ,
[ rdf:type owl:Restriction ;
  owl:onProperty :acceptCurrencyNegotiation ;
  owl:someValuesFrom xsd:string
] ,
[ rdf:type owl:Restriction ;
  owl:onProperty :risk ;
  owl:someValuesFrom xsd:string
] ,
[ rdf:type owl:Restriction ;
  owl:onProperty :shortDescription ;
  owl:someValuesFrom xsd:string
] ,
[ rdf:type owl:Restriction ;
  owl:onProperty :company ;
```

```
                owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                owl:onDataRange xsd:string
               ] ,
               [ rdf:type owl:Restriction ;
                owl:onProperty :id ;
                owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                owl:onDataRange xsd:string
               ] .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#DELETE
:DELETE rdf:type owl:Class ;
      rdfs:subClassOf :REST .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#DIGICOR
:DIGICOR rdf:type owl:Class ;
       rdfs:subClassOf :Service .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#DIGICOR_Platform
:DIGICOR_Platform rdf:type owl:Class ;
            rdfs:subClassOf :System ,
                    [ rdf:type owl:Restriction ;
                     owl:onProperty :hasTool ;
                     owl:someValuesFrom :SCM
                    ] ,
                    [ rdf:type owl:Restriction ;
                     owl:onProperty :hasTool ;
                     owl:someValuesFrom :TDMS
                    ] .
```

```
### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#DistributedProductionPlan
:DistributedProductionPlan rdf:type owl:Class ;
                    rdfs:subClassOf :MessageModelElement ,
                                    [ rdf:type owl:Restriction ;
                                      owl:onProperty :hasDPPOperation ;
                                      owl:someValuesFrom :Operation
                                    ] ,
                                    [ rdf:type owl:Restriction ;
                                      owl:onProperty :orderid ;
                                      owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                                      owl:onDataRange xsd:string
                                    ] ,
                                    [ rdf:type owl:Restriction ;
                                      owl:onProperty :productionGoal ;
                                      owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                                      owl:onDataRange xsd:string
                                    ] .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#External
:External rdf:type owl:Class ;
        rdfs:subClassOf :System .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#GET
:GET rdf:type owl:Class ;
    rdfs:subClassOf :REST .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#ListOfTeamDescription
:ListOfTeamDescription rdf:type owl:Class ;
                rdfs:subClassOf :Message ,
                                [ rdf:type owl:Restriction ;
```

```
                owl:onProperty :contain ;
                owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                owl:onClass :Team
               ] .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Message
:Message rdf:type owl:Class ;
        rdfs:subClassOf [ rdf:type owl:Restriction ;
                          owl:onProperty :contain ;
                          owl:allValuesFrom :MessageModelElement
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :id ;
                          owl:someValuesFrom xsd:string
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :serialization ;
                          owl:someValuesFrom xsd:string
                        ] .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#MessageModelElement
:MessageModelElement rdf:type owl:Class .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Operation
:Operation rdf:type owl:Class ;
         rdfs:subClassOf :MessageModelElement ,
                         [ rdf:type owl:Restriction ;
                           owl:onProperty :task ;
                           owl:someValuesFrom xsd:string
                         ] ,
```

```
                [ rdf:type owl:Restriction ;
                  owl:onProperty :productName ;
                  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                  owl:onDataRange xsd:string
                ] ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :supplier ;
                  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                  owl:onDataRange xsd:string
                ] .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#POST
:POST rdf:type owl:Class ;
    rdfs:subClassOf :REST ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :produceMessage ;
                  owl:someValuesFrom :Message
                ] .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Platform
:Platform rdf:type owl:Class .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#ProductionDescription
:ProductionDescription rdf:type owl:Class ;
                rdfs:subClassOf :Message ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :contain ;
                          owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                          owl:onClass :DistributedProductionPlan
                        ] .
```

```
### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#REST
:REST rdf:type owl:Class ;
    rdfs:subClassOf :Service ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :endpoint ;
                  owl:someValuesFrom xsd:string
                ] ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :headers ;
                  owl:someValuesFrom xsd:string
                ] ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :path ;
                  owl:someValuesFrom xsd:string
                ] ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :queryParameter ;
                  owl:someValuesFrom xsd:string
                ] .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#SCM
:SCM rdf:type owl:Class ;
    rdfs:subClassOf :Tool ;
    rdfs:comment "Semantic control and monitoring tool" .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#SCM_Service
:SCM_Service rdf:type owl:Class ;
          rdfs:subClassOf :DIGICOR .
```

```
###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#SOAP
:SOAP rdf:type owl:Class ;
     rdfs:subClassOf :Service .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Service
:Service rdf:type owl:Class ;
        rdfs:subClassOf [ rdf:type owl:Restriction ;
                          owl:onProperty :produceMessage ;
                          owl:someValuesFrom :Message
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :hasName ;
                          owl:someValuesFrom xsd:string
                        ] .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#System
:System rdf:type owl:Class ;
       rdfs:subClassOf [ rdf:type owl:Restriction ;
                         owl:onProperty :hasTool ;
                         owl:someValuesFrom :Tool
                       ] ,
                       [ rdf:type owl:Restriction ;
                         owl:onProperty :isBuiltOn ;
                         owl:someValuesFrom :Platform
                       ] ,
                       [ rdf:type owl:Restriction ;
                         owl:onProperty :hasName ;
                         owl:someValuesFrom xsd:string
                       ] .
```

```
###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#TDMS
:TDMS rdf:type owl:Class ;
    rdfs:subClassOf :Tool ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :provideService ;
                  owl:someValuesFrom :TDMS_Service
                ] ;
    rdfs:comment "Tender decomposition and matchmaking service"@en .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#TDMS_Service
:TDMS_Service rdf:type owl:Class ;
          rdfs:subClassOf :DIGICOR ;
          rdfs:comment "Services provided by TDMS"@en .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Team
:Team rdf:type owl:Class ;
    rdfs:subClassOf :MessageModelElement ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :hasAssignment ;
                  owl:someValuesFrom :Assignment
                ] ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :cftid ;
                  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                  owl:onDataRange xsd:string
                ] ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :teamid ;
                  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                  owl:onDataRange xsd:string
```

```
                ] .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#TeamDescription
:TeamDescription rdf:type owl:Class ;
                rdfs:subClassOf :Message ,
                                [ rdf:type owl:Restriction ;
                                  owl:onProperty :contain ;
                                  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                                  owl:onClass :Team
                                ] .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#Tool
:Tool rdf:type owl:Class ;
    rdfs:subClassOf [ rdf:type owl:Restriction ;
                      owl:onProperty :provideService ;
                      owl:someValuesFrom :Service
                    ] ,
                    [ rdf:type owl:Restriction ;
                      owl:onProperty :hasName ;
                      owl:someValuesFrom xsd:string
                    ] .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#UPDATE
:UPDATE rdf:type owl:Class ;
      rdfs:subClassOf :REST .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#cfTs
:cfTs rdf:type owl:Class ;
    rdfs:subClassOf :GET ,
```

```
                :TDMS_Service ;
    rdfs:comment "TDMS - Call for tenders"@en .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#companies
:companies rdf:type owl:Class ;
        rdfs:subClassOf :GET ,
                        :TDMS_Service ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :hasResponse ;
                          owl:someValuesFrom :Company
                        ] ;
        rdfs:comment "TDMS - list of companies"@en .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#deleteCfT
:deleteCfT rdf:type owl:Class ;
        rdfs:subClassOf :DELETE ,
                        :TDMS_Service ;
        rdfs:comment "TDMS - delete call for tender by a tender ID"@en .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#deletecompany
:deletecompany rdf:type owl:Class ;
            rdfs:subClassOf :DELETE ,
                            :TDMS_Service ;
            rdfs:comment "TDMS - delete company by company ID"@en .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#deleteteam
:deleteteam rdf:type owl:Class ;
         rdfs:subClassOf :DELETE ,
                         :TDMS_Service ;
```

```
        rdfs:comment "TDMS - delete team by team ID"@en .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#matchmaking
:matchmaking rdf:type owl:Class ;
         rdfs:subClassOf :GET ,
                         :TDMS_Service .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#materialslist
:materialslist rdf:type owl:Class ;
           rdfs:subClassOf :GET ,
                           :TDMS_Service ;
           rdfs:comment "TDMS - list of material"@en .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#monitor
:monitor rdf:type owl:Class ;
       rdfs:subClassOf :SCM_Service .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#produce
:produce rdf:type owl:Class ;
       rdfs:subClassOf :POST ,
                       :SCM_Service ,
                       [ rdf:type owl:Restriction ;
                         owl:onProperty :produceMessage ;
                         owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                         owl:onClass :ProductionDescription
                       ] .


### http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#teamById
```

```
:teamById rdf:type owl:Class ;
        rdfs:subClassOf :GET ,
                        :TDMS_Service ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :produceMessage ;
                          owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                          owl:onClass :TeamDescription
                        ] .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#teams
:teams rdf:type owl:Class ;
      rdfs:subClassOf :GET ,
                      :TDMS_Service ,
                      [ rdf:type owl:Restriction ;
                        owl:onProperty :produceMessage ;
                        owl:someValuesFrom :TeamDescription
                      ] ;
      rdfs:comment "TDMS - list of teams"@en .


#################################################################
#    Individuals
#################################################################

###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#TeamMessage
:TeamMessage rdf:type owl:NamedIndividual ,
                      :TeamDescription ;
           :contain :team1 .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#assign1
:assign1 rdf:type owl:NamedIndividual ,
```

```
            :Assignment ;
        :assignmentId "ae611388d27f412f877a8c57772d18f4" ;
        :companyId "0000000000000005-0000000000000005" ;
        :parentProductName "door_panel1" ;
        :productName "bi_folded_door_20inch" ;
        :task "DESIGN_AND_DEVELOP" .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#assign2
:assign2 rdf:type owl:NamedIndividual ,
            :Assignment ;
        :assignmentId "0960f010f3a940249a783de9bf8ec00a" ;
        :companyId "0000000000000007-0000000000000007" ;
        :parentProductName "lavatory_door_module1" ;
        :productName "door_panel1" ;
        :task "SOURCE" .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#assign3
:assign3 rdf:type owl:NamedIndividual ,
            :Assignment ;
        :assignmentId "7541af79811548fda1c5a3f6d9665710" ;
        :companyId "0000000000000007-0000000000000007" ;
        :parentProductName "lavatory_door_module1" ;
        :productName "door_panel1" ;
        :task "PLAN_AND_MANAGE" .


###  http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO#team1
:team1 rdf:type owl:NamedIndividual ,
            :Team ;
       :hasAssignment :assign1 ,
                      :assign2 ,
```

:assign3 ;

:cftid "0000000000000000-0000000000000005" ;

:teamid "b0571aca076c41b69011c738a88eddb3" .

http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##ad5745ef_a325_42df_a909_02dcfa4b4045

<http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##ad5745ef_a325_42df_a909_02dcfa4b4045> rdf:type owl:NamedIndividual ,

:ProductionDescription ;

:contain <http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##fb1e9bac_7f29_49a4_947e_fdf8bb6ee560> .

http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##fb1e9bac_7f29_49a4_947e_fdf8bb6ee560

<http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##fb1e9bac_7f29_49a4_947e_fdf8bb6ee560> rdf:type owl:NamedIndividual ,

:DistributedProductionPlan ;

:hasDPPOperation <http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##fb909e1f_3520_4b30_9de8_cf6936af1f29> ,

<http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##39dd507c_bdc3_4094_98a6_644624d43800> ,

<http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##8d7b8044_c563_42e6_b703_33e338f075df> ;

:orderid ""^^xsd:string ,

"Verify orderID and production goal!"^^xsd:string .

http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##fb909e1f_3520_4b30_9de8_cf6936af1f29

<http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##fb909e1f_3520_4b30_9de8_cf6936af1f29> rdf:type owl:NamedIndividual ,

:Operation ;

:productName "bi_folded_door_20inch" ;

:supplier "0000000000000005-0000000000000005" ;

:task "DESIGN_AND_DEVELOP" .

http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##39dd507c_bdc3_4094_98a6_644624d43800

<http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##39dd507c_bdc3_4094_98a6_644624d43800> rdf:type owl:NamedIndividual ,

:Operation ;

:productName "door_panel1" ;

:supplier "0000000000000007-0000000000000007" ;

:task "SOURCE" .

http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##8d7b8044_c563_42e6_b703_33e338f075df

<http://www.ciirc.ctu.cz/INTSYS/ISI/DIGICOR/DEO##8d7b8044_c563_42e6_b703_33e338f075df> rdf:type owl:NamedIndividual ,

:Operation ;

:productName "door_panel1" ;

:supplier "0000000000000007-0000000000000007" ;

:task "PLAN_AND_MANAGE" .

###

General axioms

###

[rdf:type owl:AllDisjointClasses ;

owl:members (:Assignment

:Company

:DistributedProductionPlan

:Operation

:Team

)

] .

```
[ rdf:type owl:AllDisjointClasses ;
  owl:members ( :ListOfTeamDescription
                :ProductionDescription
                :TeamDescription
              )
] .


[ rdf:type owl:AllDisjointClasses ;
  owl:members ( :Message
                :MessageModelElement
                :Platform
                :Service
                :System
                :Tool
              )
] .


#################################################################
#    Rules
#################################################################

:teamMsg rdf:type <http://www.w3.org/2003/11/swrl#Variable> .

:productionMsg rdf:type <http://www.w3.org/2003/11/swrl#Variable> .

:team rdf:type <http://www.w3.org/2003/11/swrl#Variable> .

:DPP rdf:type <http://www.w3.org/2003/11/swrl#Variable> .

:assign rdf:type <http://www.w3.org/2003/11/swrl#Variable> .
```

```
:operation rdf:type <http://www.w3.org/2003/11/swrl#Variable> .

:pn rdf:type <http://www.w3.org/2003/11/swrl#Variable> .

:asgTask rdf:type <http://www.w3.org/2003/11/swrl#Variable> .

:compId rdf:type <http://www.w3.org/2003/11/swrl#Variable> .

[ <http://swrl.stanford.edu/ontologies/3.3/swrla.owl#isRuleEnabled> "true"^^xsd:boolean ;
  rdfs:comment ""^^xsd:string ;
  rdfs:label "TDMS_Team2SCM_produce"^^xsd:string ;
  rdf:type <http://www.w3.org/2003/11/swrl#Imp> ;
  <http://www.w3.org/2003/11/swrl#body> [ rdf:type <http://www.w3.org/2003/11/swrl#AtomList> ;
                        rdf:first [ rdf:type <http://www.w3.org/2003/11/swrl#ClassAtom> ;
                              <http://www.w3.org/2003/11/swrl#classPredicate> :TeamDescription ;
                              <http://www.w3.org/2003/11/swrl#argument1> :teamMsg
                             ] ;
                        rdf:rest [ rdf:type <http://www.w3.org/2003/11/swrl#AtomList> ;
                              rdf:first [ rdf:type <http://www.w3.org/2003/11/swrl#BuiltinAtom> ;
                                    <http://www.w3.org/2003/11/swrl#builtin> <http://swrl.stanford.edu/ontologies/built-ins/3.3/swrlx.owl#makeOWLThing> ;
                                    <http://www.w3.org/2003/11/swrl#arguments> ( :productionMsg
                                                                                 :teamMsg
                                                                                )
                                   ] ;
                              rdf:rest [ rdf:type <http://www.w3.org/2003/11/swrl#AtomList> ;
                                    rdf:first [ rdf:type <http://www.w3.org/2003/11/swrl#IndividualPropertyAtom> ;
                                          <http://www.w3.org/2003/11/swrl#propertyPredicate> :contain
;
                                          <http://www.w3.org/2003/11/swrl#argument1> :teamMsg ;
                                          <http://www.w3.org/2003/11/swrl#argument2> :team
                                         ] ;
                                    rdf:rest [ rdf:type <http://www.w3.org/2003/11/swrl#AtomList> ;
```

```
                                        rdf:first [ rdf:type <http://www.w3.org/2003/11/swrl#BuiltinA-
tom> ;
                                                <http://www.w3.org/2003/11/swrl#builtin>
<http://swrl.stanford.edu/ontologies/built-ins/3.3/swrlx.owl#makeOWLThing> ;
                                                <http://www.w3.org/2003/11/swrl#arguments> ( :DPP
                                                                :team
                                                               )
                                              ] ;
                                        rdf:rest [ rdf:type <http://www.w3.org/2003/11/swrl#AtomList>
;
                                                rdf:first [ rdf:type <http://www.w3.org/2003/11/swrl#Indi-
vidualPropertyAtom> ;
                                                        <http://www.w3.org/2003/11/swrl#propertyPredi-
cate> :hasAssignment ;
                                                        <http://www.w3.org/2003/11/swrl#argument1>
:team ;
                                                        <http://www.w3.org/2003/11/swrl#argument2>
:assign
                                                      ] ;
                                                rdf:rest [ rdf:type
<http://www.w3.org/2003/11/swrl#AtomList> ;
                                                        rdf:first [ rdf:type
<http://www.w3.org/2003/11/swrl#BuiltinAtom> ;
                                                                <http://www.w3.org/2003/11/swrl#builtin>
<http://swrl.stanford.edu/ontologies/built-ins/3.3/swrlx.owl#makeOWLThing> ;
                                                                <http://www.w3.org/2003/11/swrl#argu-
ments> ( :operation
                                                                                :assign
                                                                              )
                                                             ] ;
                                                        rdf:rest [ rdf:type
<http://www.w3.org/2003/11/swrl#AtomList> ;
                                                                rdf:first [ rdf:type
<http://www.w3.org/2003/11/swrl#DatavaluedPropertyAtom> ;

<http://www.w3.org/2003/11/swrl#propertyPredicate> :productName ;
                                                                        <http://www.w3.org/2003/11/swrl#ar-
gument1> :assign ;
```

<http://www.w3.org/2003/11/swrl#argument2> :pn

] ;

rdf:rest [rdf:type <http://www.w3.org/2003/11/swrl#AtomList> ;

rdf:first [rdf:type <http://www.w3.org/2003/11/swrl#DatavaluedPropertyAtom> ;

<http://www.w3.org/2003/11/swrl#propertyPredicate> :task ;

<http://www.w3.org/2003/11/swrl#argument1> :assign ;

<http://www.w3.org/2003/11/swrl#argument2> :asgTask

] ;

rdf:rest [rdf:type <http://www.w3.org/2003/11/swrl#AtomList> ;

rdf:first [rdf:type <http://www.w3.org/2003/11/swrl#DatavaluedPropertyAtom> ;

<http://www.w3.org/2003/11/swrl#propertyPredicate> :companyId ;

<http://www.w3.org/2003/11/swrl#argument1> :assign ;

<http://www.w3.org/2003/11/swrl#argument2> :compId

] ;

rdf:rest rdf:nil

]

]

]

]

]

]

] ;

<http://www.w3.org/2003/11/swrl#head> [rdf:type <http://www.w3.org/2003/11/swrl#AtomList> ;

rdf:first [rdf:type <http://www.w3.org/2003/11/swrl#ClassAtom> ;

```
                    <http://www.w3.org/2003/11/swrl#classPredicate> :ProductionDescription ;
                    <http://www.w3.org/2003/11/swrl#argument1> :productionMsg
                   ] ;
            rdf:rest [ rdf:type <http://www.w3.org/2003/11/swrl#AtomList> ;
                    rdf:first [ rdf:type <http://www.w3.org/2003/11/swrl#ClassAtom> ;
                            <http://www.w3.org/2003/11/swrl#classPredicate> :Distribut-
edProductionPlan ;
                            <http://www.w3.org/2003/11/swrl#argument1> :DPP
                           ] ;
                    rdf:rest [ rdf:type <http://www.w3.org/2003/11/swrl#AtomList> ;
                            rdf:first [ rdf:type <http://www.w3.org/2003/11/swrl#IndividualProp-
ertyAtom> ;
                                    <http://www.w3.org/2003/11/swrl#propertyPredicate> :contain
;
                                    <http://www.w3.org/2003/11/swrl#argument1> :produc-
tionMsg ;
                                    <http://www.w3.org/2003/11/swrl#argument2> :DPP
                                   ] ;
                            rdf:rest [ rdf:type <http://www.w3.org/2003/11/swrl#AtomList> ;
                                    rdf:first [ rdf:type
<http://www.w3.org/2003/11/swrl#ClassAtom> ;
                                            <http://www.w3.org/2003/11/swrl#classPredicate> :Op-
eration ;
                                            <http://www.w3.org/2003/11/swrl#argument1> :opera-
tion
                                           ] ;
                                    rdf:rest [ rdf:type <http://www.w3.org/2003/11/swrl#AtomList>
;
                                            rdf:first [ rdf:type <http://www.w3.org/2003/11/swrl#Indi-
vidualPropertyAtom> ;
                                                    <http://www.w3.org/2003/11/swrl#propertyPredi-
cate> :hasDPPOperation ;
                                                    <http://www.w3.org/2003/11/swrl#argument1>
:DPP ;
                                                    <http://www.w3.org/2003/11/swrl#argument2>
:operation
                                                   ] ;
```

```
                                                rdf:rest [ rdf:type
<http://www.w3.org/2003/11/swrl#AtomList> ;
                                                        rdf:first [ rdf:type
<http://www.w3.org/2003/11/swrl#DatavaluedPropertyAtom> ;
                                                                <http://www.w3.org/2003/11/swrl#proper-
tyPredicate> :productName ;
                                                                <http://www.w3.org/2003/11/swrl#argu-
ment1> :operation ;
                                                                <http://www.w3.org/2003/11/swrl#argu-
ment2> :pn
                                                               ] ;
                                                        rdf:rest [ rdf:type
<http://www.w3.org/2003/11/swrl#AtomList> ;
                                                                rdf:first [ rdf:type
<http://www.w3.org/2003/11/swrl#DatavaluedPropertyAtom> ;

<http://www.w3.org/2003/11/swrl#propertyPredicate> :task ;
                                                                        <http://www.w3.org/2003/11/swrl#ar-
gument1> :operation ;
                                                                        <http://www.w3.org/2003/11/swrl#ar-
gument2> :asgTask
                                                                      ] ;
                                                                rdf:rest [ rdf:type
<http://www.w3.org/2003/11/swrl#AtomList> ;
                                                                        rdf:first [ rdf:type
<http://www.w3.org/2003/11/swrl#DatavaluedPropertyAtom> ;

<http://www.w3.org/2003/11/swrl#propertyPredicate> :supplier ;

<http://www.w3.org/2003/11/swrl#argument1> :operation ;

<http://www.w3.org/2003/11/swrl#argument2> :compId
                                                                              ] ;
                                                                        rdf:rest [ rdf:type
<http://www.w3.org/2003/11/swrl#AtomList> ;
                                                                                rdf:first [ rdf:type
<http://www.w3.org/2003/11/swrl#DatavaluedPropertyAtom> ;

<http://www.w3.org/2003/11/swrl#propertyPredicate> :orderid ;
```

<http://www.w3.org/2003/11/swrl#argument1> :DPP ;

<http://www.w3.org/2003/11/swrl#argument2> ""^^xsd:string

] ;

rdf:rest [rdf:type

<http://www.w3.org/2003/11/swrl#AtomList> ;

rdf:first [rdf:type

<http://www.w3.org/2003/11/swrl#DatavaluedPropertyAtom> ;

<http://www.w3.org/2003/11/swrl#propertyPredicate> :orderid ;

<http://www.w3.org/2003/11/swrl#argument1> :DPP ;

<http://www.w3.org/2003/11/swrl#argument2> "Verify orderID and production goal!"^^xsd:string

] ;

rdf:rest rdf:nil

]

]

]

]

]

]

]

]

] .

Generated by the OWL API (version 4.2.8.20170104-2310) https://github.com/owlcs/owlapi

DIGICOR

www.digicor-project.eu