NIMBLE Collaborative Platform: Microservice Architectural Approach to Federated IoT

Johannes Innerbichler
The IoT Group
Salzburg Research GmbH
Salzburg, Austria

Suat Gonul
Software Research, Development and Consultancy
Ankara, Turkey

Violeta Damjanovic-Behrendt
The IoT Group
Salzburg Research GmbH
Salzburg, Austria

Benjamin Mandler
The Cloud and Distributed Middleware Group
IBM Israel Science and Technology Ltd
Haifa, Israel

Felix Strohmeier
The IoT Group, Salzburg Research GmbH
Salzburg, Austria

Abstract—This paper presents our architectural approach to building a collaborative Industry 4.0 platform that enables IoT-based real-time monitoring, optimization and negotiation in manufacturing supply chains. The platform is called NIMBLE and is currently being developed in a European research project. The presented approach utilizes microservice technology, and implements the core business functionality of the platform through a composition of decentralized, scalable services. The communication among services, and with platform users, manufacturers, suppliers, sensors and Web resources, is supported through simple protocols and lightweight mechanisms. Core business services of the implemented architecture are released as open source software, enabling multiple prospective platform-providers to establish B2B marketplaces for collaboration within their own industrial sector or region. To demonstrate microservices in practice, we present two scenarios, both related to manufacturing of wooden home buildings: one is IoT-based data sharing in a supply chain, and the other deals with product driven logistics planning. The further development of the platform will be driven by the requirements of at least four different use cases throughout Europe, and by incorporating advanced business models to support the growth of powerful network effects of the platform.

Index Terms—microservice, cloud, IoT, IoT-based supply chain, platform, Industry 4.0, federation.

I. Introduction

The IoT has the ability to support collaboration in manufacturing and supply chains and is likely to disrupt many currently established processes. It forces companies to make their processes capable of utilizing IoT-enabled devices and attendant data. It has also the potential to significantly improve supply chain efficiency by providing real-time insights into the different manufacturing states of components that are distributed geographically and in time.

This paper presents our approach to building a federated IoT-based collaboration platform called NIMBLE, which is currently being developed in a European research project. NIMBLE's governance model is a federation of platforms for multi-sided B2B trade, with mandatory interoperation facilities between instances of the platform and with optional, added-value business functions that can be provided by third parties. Prospective NIMBLE platform-providers can take the open source infrastructure with its core services, bundle it with sectoral, regional or functional added value services and launch the new platform in the federation (Figure 1).

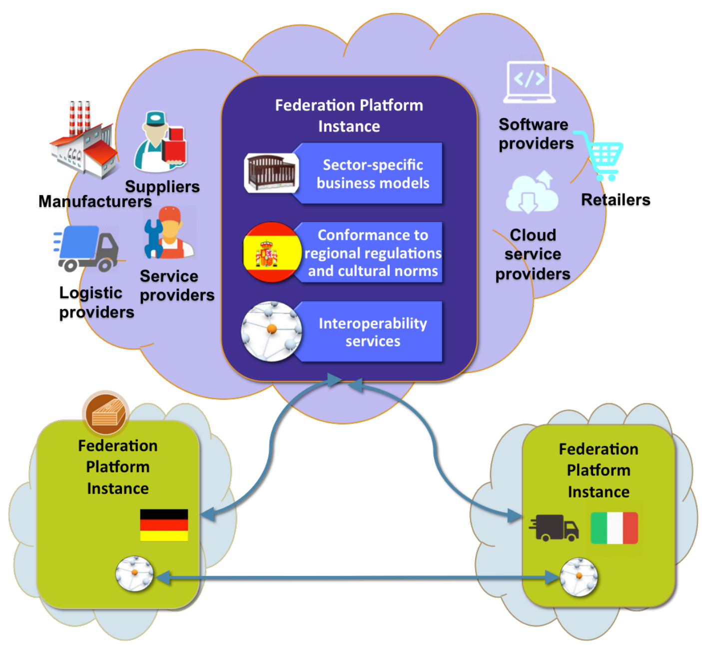

Fig. 1. NIMBLE federated collaboration network with sector-specific platform instances.

We adopted the microservice approach for the realisation of the platform. Microservices offer a number of advantages over traditional solutions: for example, they can be used in more than one application and in various application architectures, thus providing better scalability, better agility and continuous delivery [1]. A single application that is built in microservice manner is a composition of services communicating with lightweight mechanisms, each running in its own process and being managed in a decentralized way. Microservices can use different data storage technologies, different programming languages to define services and can even be managed by different teams [2]. They are often choreographed using simple REST protocols like HTTP request-response with resource APIs and lightweight messaging, e.g. binary protocols, or a lightweight messaging bus. In the IoT, cloud native microservices allows for devices to be interpreted as a set of microservices with the possibility to communicate with each other, or with internal and external microservices, facilitating new levels of customization, security, workload changes, and significantly contributing to the overall system scalability, ease of development and of maintenance, simplifying validation and testing [3]. They can be often seen as the major technology behind public services, e.g. Netflix, eBay, Twitter, Paypal, Bluemix Soundcloud, The Guardian, etc.

The NIMBLE microservices provide the core business functions of the platform: registration of users on the platform, publishing machine-readable catalogs for products and services, searching for suitable supply chain partners, negotiation contracts and supply logistics, developing private and secure B2B and M2M (Machine to Machine) information exchange channels to optimise business workflows.

The remaining parts of the paper are structured as follows: Section 2 describes our industrial use case, which is about manufacturing wooden multi-storey homes. Section 3 presents the microservice architecture of the NIMBLE platform and discusses the role of its major components. Section 4 introduces the core business microservices. Section 5 puts the microservices in action through two scenarios: (i) data sharing in a supply chain, and (ii) product-driven logistics planning. Section 6 concludes the paper.

II. Industrial Use Case: Collaborative Planning and Tracking of Assets in a Manufacturing System for Wooden Buildings

The impact of the IoT on a business can be significant in areas such as reducing costs, optimizing staff and resources, and improving customer satisfaction. The first and foremost capability is that of assets tracking. Such capability can be realized in different stages of the supply chain, from the location of expected supplies to the location of finished goods that are ready for delivery. Similar capabilities existed before the era of IoT but due to the cost involved were generally restricted to larger bulks of items and without providing the real-time view. Nowadays, with location reporting IoT devices as part of the portfolio, we can track in real-time the location and movement of individual items, gain important insights, and as a consequence devise adequate and realistic plans for logistics. For example, an early detection that a required resource for production is late may enable the company to adjust production lines or acquire supplies from a different source. Moreover, when multiple entities are involved in the supply chain we can gain insights, as to the responsible party at each location the shipment is in, and hold that entity accountable. In addition, different IoT devices can provide additional knowledge concerning the tracked items, such as conditions, which are of interest to the cargo in question. Finally, some of the information gained can be shared with customers, leading to enhanced customer satisfaction.

Our use case presents a company with expertise in wooden construction of apartment buildings. The production processes are organized in a linear manner, i.e. the wooden raw material is processed and assembled into room-size modules. Each module represents one or more rooms and is ready for on-site assembly. The modules are transported to the location of the building for final assembly. Transport is organized via trucks that deliver the modules. Some of the value-adding activities in the product life-cycle are transferred to the suppliers, e.g. bathroom units made from composite material, the units with tiles, etc. The current challenges include (i) automated logistics management (e.g. dealing with redundancy and delay), (ii) tracking of the modules during assembly and transportation, as well as (iii) monitoring of the environmental conditions (temperature, humidity) during transport, construction of the building and during use of the building.

With respect to these identified challenges, the NIMBLE collaborative platform is expected to support the manufacturer and their suppliers in the following ways:

- to dynamically establish logistic chains: New partners need to be identified, evaluated (e.g. size, legal form, liquidity, experience, certificates), bound by contract and integrated into the information architecture of the current processes.
- to support negotiation processes: The platform needs to support the negotiation and the evaluation of different alternatives, e.g. if one partner is exchanged for another, what is the impact on, for instance, cost, quality or time.
- to relay quality information of supplier's products: Information exchange between the actors in logistic chains need to be based on a common information standard. The exchanged information may concern static values like the characteristics of the glass fibre material and the resin, or plaster to glue the tiles.
- to monitor transport and on-site construction: All actors in the logistics chain need to have access to at least some of the monitored data. While the logistics provider can use the data to adapt transport conditions (e.g. actively reduce moisture), the manufacturer may trace the condition to, for instance, convince insurance companies that potential quality problems (e.g. growth of mold) are not related to inappropriate transport conditions.
- to monitor environmental conditions: Dynamic environmental parameters must be collected

continuously for control purpose and traceability of quality issues.

Overall, collaboration via the NIMBLE platform is expected to lead to a decrease in manual data tracing and e-mail communication, which will benefit quality control, lead to the reduction of throughput time, lower costs for manufacturing and personnel costs spent on administrative tasks, reduce time to connect suppliers and logistics provider, etc. The NIMBLE architecture has been devised to provide support for such capabilities.

III. NIMBLE Platform Architecture

The NIMBLE platform is designed to ingest data stemming from edge devices into the platform. Once data has reached the platform, data are stored for future uses such as offline analytics and auditing. In addition, processing of the data as it flows into the platform can be performed in real-time. Offline analytics complements the real-time view by enabling companies to gain insights from their data leading to more efficient processes.

We follow the microservice approach for the NIMBLE architecture. Each microservice offers a specific and narrowly defined functionality that is deployed as an independent service [2]. It puts emphasis on real-time application monitoring, including architectural elements and business relevant metrics, e.g. number of orders per minute [2]. In addition, this approach provides decentralized data management and decentralized data storage decisions based on a polyglot persistence method. Microservices can still be combined with monolithic enterprise applications (monolith's APIs) built as single units, fostering the federation and collaboration of various independent services in the Cloud.

Figure 2 depicts the top-level architecture, which is inspired by the microservice infrastructure presented in [4]. Figure 3 then shows the core business services of the platform in more detail. *The Gateway Proxy* is the public entry point to the microservice application. All user requests are received via the Gateway Proxy and routed to the appropriate service. *The Service Logging* aggregates log outputs of each individual microservice and makes them available via a Web interface. The *Service Monitoring* collects data, which are necessary to monitor failures of services; this is implemented using the Circuit Breaker pattern [5]. The *Service Monitoring* can be used to monitor remote calls, or in any situation where parts of a system need to be protected from failures in other parts. After the number of erroneous executions exceeds a defined threshold, all future calls are immediately returned with an error without the actual function being executed.

The Service Discovery enables registration of microservices, as well as mapping between service identifier and the endpoints of instances (e.g. URL or IP address). The role of the *Service Configuration* is to provide configuration settings for each service.

The service communication is enabled either via REST (i.e. HTTP) or by using the *Messaging Service* (based on Apache Kafka), which provides asynchronous and scalable communication. Finally, *the User Account and Authentication (UAA) & Identity Management* administrates user identities on the platform. It combines the three standard protocols: OAuth2 for delegated authorization [6], OpenID Connect for session and authentication information [7], and the System for Cross-domain Identity Management (SCIM) for user and group management [8].

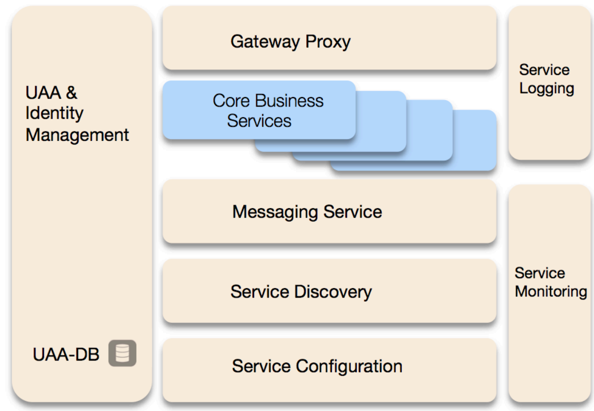

Fig. 2. Top level of the NIMBLE microservice architecture.

IV. NIMBLE Microservices

Figure 3 shows the core microservices in NIMBLE, thus giving details for Figure 2.

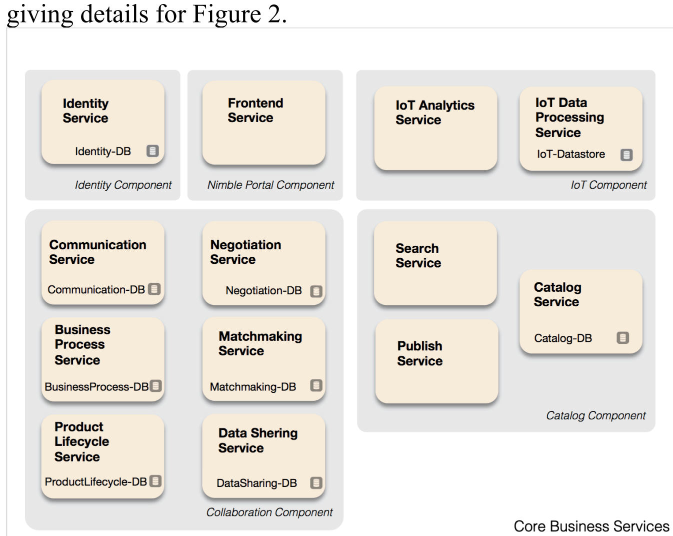

Fig. 3. Core business microservice of the NIMBLE platform.

Services with related functionality are grouped into the following service components:

- *The Portal component* provides the central *Frontend Service* to support user interactions and to delegate requests to other services; for example, registration requests are delegated to the *Identity Service*.
- *The Identity component* administrates the identities of the users and enterprises on the platform via the *Identity Service*, which communicates with the UAA & Identity Management (see Figure 2). The *Identity Service* receives requests to create, read, update and

delete (CRUD) for both users and company entities. It also stores data related to the company identity (Identity-DB).

- *The Catalog component* contains three services: (i) A *Search Service* provides search functionality, communicates with a C*atalog Service,* which stores products and user services persistently, and manages caching of searches. (ii) A *Publish Service* receives CRUD requests for products and services, pre-processes CRUD operations, delegates requests to a *Catalog Service* and resets the cache of the search service. This service makes products, IoT-enabled devices and intangible assets (e.g. services, production plans, catalogs) discoverable through the NIMBLE platform. (iii) A C*atalog Service* maintains the published catalogs and offers semantically enhanced search features for catalogs, based on (a) the *GoodRelations* ontology [9] for e-commerce, supported by Google's schema.org and used in BestBuy, Yahoo, O'Reilly, and (b) the eClassOWL ontology [10], which covers the commercial aspects of offers and demand, e.g. prices, payment, or delivery options, and is often used in combination with *GoodRelations*.
- *The IoT component* enables communication among IoT devices and the NIMBLE platform and its services. The *IoT Data Processing Service* receives IoT data via MQTT, performs the first preprocessing of data, and stores data persistently for further analysis. The *IoT Analytics Service* is responsible for the analytics of collected data, and provides insights into the results of analyses for other services.
- *The Collaboration component* enables collaboration between the users/companies interacting via the platform, and includes several services: (i) *The Communication Service* handles communication between the users, and stores communication history persistently in Communication-DB. (ii) *The Negotiation Service* monitors negotiation among platform entities, augments the negotiation process, and stores data about actions and events that happened during the negotiation phase in Negotiation-DB. This service gives information about company size, legal form, liquidity, experience, etc. It also contains negotiation models suggesting which business actor to choose first, having options for side deals, and specifying whether the negotiation is one-time or an ongoing process, etc. (iii) *The Matchmaking Service* collects appropriate data for matchmaking from other services, e.g. from the *Communication Service*, the *Negotiation Service*, the *Identity Service*, and from the *Search Service*. Furthermore, it performs matchmaking based on collected data, and stores the matchmaking results for further analysis. In case of a negotiation failure, this service should enable continuation of the negotiation process, recommending alternative business actors. (iv) *The Business Process Service* provides the major functionality to support development of collaborative business models. It provides ready-to-run business process constructs for supply chain operations, such as ordering, fulfillment, freight management, shipment, payment, eventually leading to standards-based collaboration processes, such as Collaborative Planning, Forecasting and Replenishment (CPFR), CPFR-CTM (an extension of CPFR for Collaborative Transportation Management), Vendor Managed Inventory (VMI), etc. Small enterprises will be able to tailor the predefined templates to their specific needs. (v) *The Product Lifecycle Service* retrieves and stores product lifecycle data, search data, analyse lifecycle data, etc. (vi) *The Data Sharing Service* setups data sharing channels, handles data privacy, policies and right-management for data sharing. It does not store any data that is being shared, except metadata necessary for managing data sharing channels.

V. Microservices in Practice

To demonstrate the microservices in practice, we created two scenarios which relate to the wooden home building use case from Section 2: firstly, IoT-based data sharing in a supply chain, and secondly, product driven logistics planning.

A. Data Sharing in a Supply Chain

Figure 4 illustrates the first scenario: IoT sensors in wooden home manufacturing discover some problem in the production stage of the lifecycle. The *IoT Analytics Service* detects relevant features of the problem, e.g. nature of the problem, location, time, possible causes, temperature, humidity of the raw wood, humidity in the production room, etc. Depending on the amount of manufacturing knowledge acquired on the platform (e.g. product and service ontology), the *IoT Analytics Service* automatically assesses the likely production delay. The *IoT Analytics Service* sends information about the delay in production to *the Data Sharing Service*.

Delayed delivery affects other parties in the supply chain. Therefore, the *Data Sharing Service* contacts the *Negotiation Service*, searches for all relevant parties involved in the specific product lifecycle and contacts the *Business Process Service* to check if there exists any predefined business procedure to be applied for the case of the specific delay in manufacturing. Based on the suggested business procedures, the *Business Process Service* consults the *Negotiation Service* that initiates a negotiation process on how to overcome the problem in production. The parties involved in data sharing and negotiation are notified via the *Communication Service*. The affected parties start a negotiation process, which ends with a collaboratively agreed solution. Last steps in the procedure involve an update of the product lifecycle features, and notification of affected parties (Figure 4).

B. Production Driven Logistic Planning

The IoT brings to companies an opportunity to dynamically develop new business models by taking into consideration gathered production data, previous decisions, manufacturing results, the real-time events from the manufacturing shop-floor, etc. Inbound and outbound business services can be developed by making use of data coming from business partners and product users. For example, it is possible to monitor production quality of the supplier of the bathroom module (inbound), or by sharing data generated on-premise with partners (outbound).

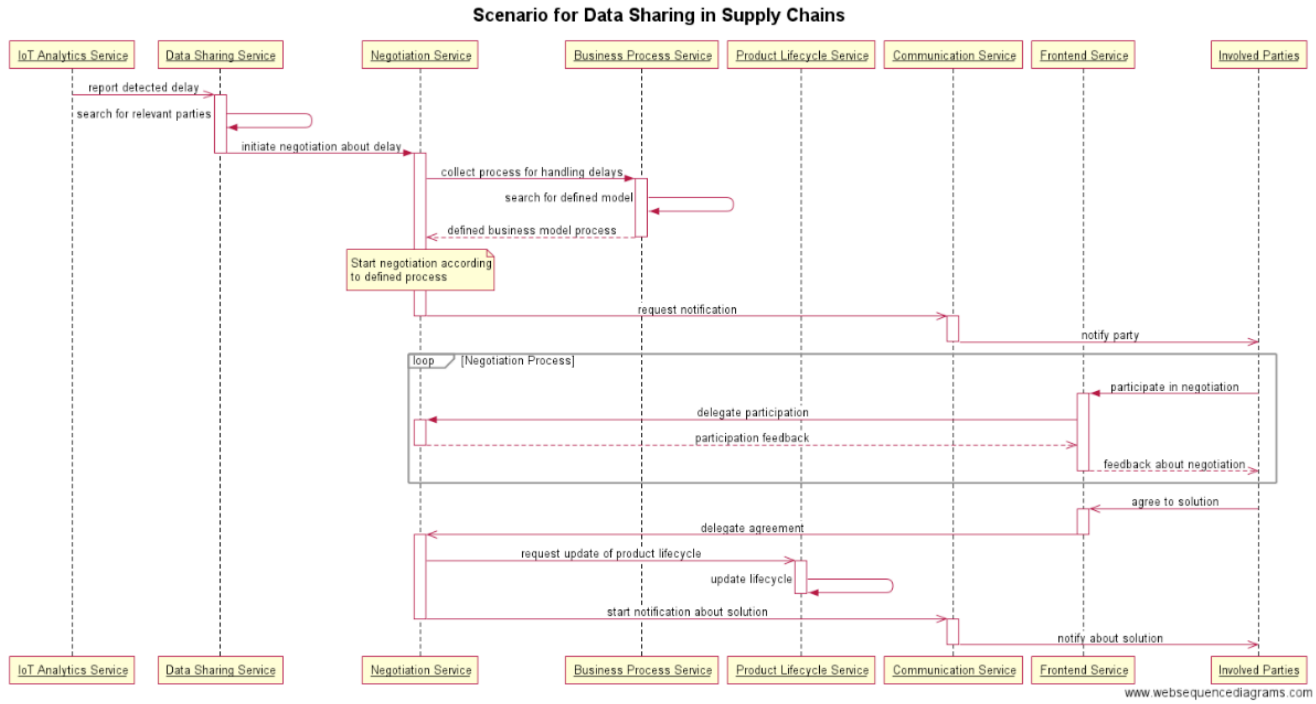

Fig. 4. Microservice in action: data sharing for supply chain optimization.

In addition to raw data sharing, NIMBLE envisions logistics planning based on tracking of production data for a specific order. It also incorporates information from legacy systems, and information from humans. Synchronization of production and logistics processes is done via services for data integration, data channels and data transformations. Data integrators implement the connection to the IoT objects using different IoT protocols like MQTT for publish/subscribe communication or CoAP for push/pull. They are coupled with data channels to allow companies to configure which data to share with which partner and under which conditions. Data transformation provides conversions for data, in case the respective data consumer uses another data format or data model. All three services are discoverable through the *Service Discovery* module (Figure 2). In addition to being a registry of tangible products, the *Catalog Registry* (Figure 2) also offers the functionality for discovery of such services. Both the data management components and data-driven services of the platform can be reused to create dynamic business processes.

VI. Conclusion and Future Work

The NIMBLE platform is designed to support horizontal B2B collaboration for manufacturing, business and logistics in Europe. With enhanced data management capabilities in place, it helps to further strengthen and deepen collaborations between different entities by providing insights into the relevant data. The presented microservice architecture with its core business services ensures a scalable approach, which can be easily extended with more specific value-added business services from third parties. The possibility of integrating enterprise IoT data via the platform will not only enable the optimization of current business processes, but also allow the development of completely new business models for the future.

Dynamic data sharing is governed by the security policies of the sharing entity such that only the specifically relevant items of data are shared with other entities. Documented negotiation mechanisms may help strengthening the trust in supply chain relationships. Enabling dynamic real-time IoT-based multi-stakeholder supply chain processes, such as optimizing production lines use, lies at the heart of the platform presented.

In summary, we envision that enhanced supply chain services need to be IoT-driven and enable the entities involved to take more informed and timely decisions. The presented architecture is a step on the way to realize that vision. To show the potential of this IoT-enabled B2B platform, the developed system will be tested at least in four different use cases throughout Europe, one of them has been described in this paper.

Acknowledgment

This research has been funded by the European Commission within the H2020 project NIMBLE (Collaborative Network for Industry, Manufacturing, Business and Logistics in Europe), No. 723810, for the period between 01 October 2016 - 31 September 2019.

References

[1] R. Buest: “Microservice: Cloud and IoT applications force the CIO to create novel IT architectures”, May 2015, Online available: http://analystpov.com/cloud-computing/microservice-cloud-and-iot-applications-force-the-cio-to-create-novel-it-architectures-25117, (Last accessed: January 13, 2017)

[2] J.Lewis and M. Fowler, “Microservices, a definition of this new architectural term”, 2014. Online available: https://www.martinfowler.com/articles/microservices.html#footnote-etymology (Last accessed: January 11, 2017)

[3] V. Benedichuk, 2015. Data stream processing and microservice architecture. Online available: https://www.slideshare.net/vyacheslavbenedichuk/stream-processing-and-micro-service-architecture (last accessed March 2017)

[4] J. Adersberger, M.L. Reimer, and A. Zitzelsberger, “Über den Wolken”, Javamagazin 4.2016, pp. 70-74 (In German), “Over the Clouds”, Javamagazin 4.2016, pp. 70-74, 2016.

[5] M. Fowler, “Circuit Breaker”, 2014. Online available: https://martinfowler.com/bliki/CircuitBreaker.html (Last accessed: January 11, 2017)

[6] OAuth2. Online available: http://tools.ietf.org/html/rfc6749 (Last accessed: January 15, 2017)

[7] OpenID Connect, Online available: http://openid.net/connect/ (Last accessed: January 15, 2017)

[8] SCIM, Online available: http://www.simplecloud.info/ (Last accessed: January 15, 2017)

[9] GoodRelations ontology, Online available: http://wiki.goodrelations-vocabulary.org/Vocabularies (Last accessed: January 15, 2017)

[10] eClassOWL ontology, Online available: http://www.heppnetz.de/projects/eclassowl/ (Last accessed: January 15, 2017).