D3.5 PROPOSAL OF AN OMG SPECIFICATION

Project Acronym:	DiDIY
Project Name	Digital Do It Yourself
Grant Agreement no.	644344
Start date of the project	01/01/2015
End date of the project	30/06/2017
Work Package producing the document	WP3 – Analysing how DiDIY is reshaping organization and work
WP Lead Partner	LIUC
Other Partner(s) involved	All
Deliverable identifier	D3.5
Deliverable lead beneficiary	LIUC
Due date	M24 (December 2016)
Date of delivery	31/12/2016
Version	1.0
Author(s)	LIUC
License	Creative Commons Attribution ShareAlike 4.0
Classification	PUBLIC
Document Status	APPROVED
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644344.	
Disclaimer: The views expressed in this document do not necessarily reflect the views of the EC.	

Disclaimer

This document is provided “As Is”; it is a study introducing the main research topics in the presented context. Any feedback, suggestions and contributions to make this document better and more useful are very welcome. Please let us know through the contact page http://www.didiy.eu/contact. We will seek to incorporate relevant contributions in the document and add your name to the list of contributors.

Executive summary

Deliverable D3.5, “Proposal of an OMG specification”, is aimed at providing a formalization of some of the main outcomes of the WP3. The emerging nature of the DiDIY phenomenon, largely discussed in previous deliverables, makes it unfeasible to develop from scratch a new OMG specification. Thus in D3.5, we discuss the topic of formalizing DiDIY impact on organization and work, and we conclude that, to such purpose, the most adequate existing OMG specification is UML Use Cases. We therefore apply this standard to each of the five Research Topics, as detailed in D3.6, that constitute organizational application domains of DiDIY: the result is a set of five UML use case diagrams representing the five domains.

Revision history			
Version	**Date**	**Created / modified by**	**Comments**
0.1	10/09/2016	LIUC	First, incomplete draft.
0.2	18/12/2016	LIUC	First full draft
0.3	21/12/2016	LIUC	Second full draft
0.4	29/12/2016	LIUC	Third full draft
0.5	30/12/2016	LIUC	Fourth full draft
1.0	31/12/2016	LIUC	Approved version, submitted to the EC Participant Portal.

Table of Contents

1. Introduction

The purpose of this deliverable, as defined in the original Project proposal, is the formalization of some of the aspects that characterize the phenomenon of Digital DIY. The assumption is that it is possible to describe DiDIY, or part of it, using the models introduced by the OMG.

The Object Management Group (OMG) is an international consortium whose goal is to define the technological standards in the field of computer science. Initially his rule was the standardization of distributed systems. It then expanded its area of intervention to modelling in the context of software engineering, in particular to the creation of “model-based standards”.

In general, we can imagine different ways of using the OMG specifications for the formalization of a particular application domain, being this a context or a phenomenon. One possible way is the application of an OMG specification. The second type of usage is the adaptation of a specification to an application domain of interest. A third method consists in extending an existing specification. The fourth type of usage is the creation of an ex-novo specification. In all these cases it is important first to identify – among the several standards developed by OMG – one that allows to highlight the characteristics of the most relevant domain than the pre-identified purposes.

1.1 OMG and OMG specifications

The aim of OMG is to develop standards for a broad set of technologies and to apply them to a number of different industries (OMG 2016). Among these standards, there are the modelling specifications, among which the best known are UML (*Unified Modeling Language*) and MDA (*Model Driven Architecture*), which allow the visual design, execution and maintenance of software and business process.

For the aims of the current deliverable D3.5, it is relevant to note that the proposal of any specification is assessed also in terms of applicability. If a proposed specification does not have an implementation plan, i.e., it is not immediately usable, then it cannot be adopted by OMG.

An OMG specification normally consists of a document typically following a template articulated into at least the six sections presented hereafter.

The **scope**, where the purpose and the domain of application of the specification is described, for example in terms of how the specification meets the challenges of the application domain. For example, in VDML (*Value Delivery Metamodel*), the challenge of modelling business transformations in environments of high variability is met because VDML “provides operational definition of critical business frameworks (Osterwalder, balanced scorecard, value stream)” (OMG 2015a).

The **conformance**, describing the criteria and the method according to which the application of the OMG standard is evaluated conforming to the standard itself. For example, in BMM (*Business Process Maturity Model*) one of the method to assess conformance is “interviews with individuals or groups who perform a process” (OMG 2008).

Normative References, the formal references upon which the specification is developed. They can consist of other OMG standards or any scientific paper used as a theoretical background for the standard.

Terms and definitions and **symbols**, i.e., a glossary of the terms and the symbols, whose syntax and semantics shall be respected in order to have a conforming application of the standard.

Explanation of the OMG standard, which is the core of the document, describing precisely and formally the semiotics of the specification.

Notation, a section dedicated to show how the symbols introduced before should be used to represent the concepts modelled by means of the specification.

The OMG web site contains the whole standard documentation related to these specifications, which are classified according to the following structure:

- Business Modelling specifications;
- Middleware specifications;
- Language Mapping specifications;
- ISO adopted specifications;
- Modelling and Metadata specifications;
- UML profile specifications;
- Modernization specifications;
- Platform Independent Models, Platform Specific Models, Interface specifications;
- CORBA related specifications;
- OMG domain specifications;
- Signal- and Image-processing specifications.

Some of these specifications (such as the Middleware specifications and the Language Mapping specifications) are exclusively inherent the technological components of an enterprise. Others, however, cover a wider representation domain, which includes information, activities, and organizational roles. This second category includes some of the most broadly used OMG specifications such as UML, BPMN and SysML (Watson 2008).

UML was created in 1997 and has rapidly become the international standard for software design. The versatility of the UML, however, has allowed to use it for the representation of both hardware systems, and business processes.

BPMN was created in 2004 by the Business Process Management Initiative, a consortium (outside of the OMG) of manufacturers of instruments who judged UML unfit for representing business processes. In 2006, the BPMN was adopted by OMG despite a clear overlap: the main diagram of the BPMN (*Business Process Diagram*) is a flow chart, just like the UML Activity Diagram. The two charts show only a few notational differences. Noteworthy, the idea of integrating UML and BPMN, i.e., two standards partially overlapping issued and managed by the same body, has so far been hampered by the producers of modelling tools, with the consequence that the two specifications still coexist. Both UML and BPMN have been recognized by ISO, thus belonging to the category of ISO adopted specifications.

To represent “non-software” systems, OMG published in 2007 a new standard, SysML, which includes some of the UML diagrams. SysML is a derivation of UML, compatible with the base language, and extends it to allow modelling a different set of entities, and it is typically used in Systems Engineering.

In general, UML and UML related specifications (such as SysML) have been proposed to be used by software engineers, to support requirements engineering and software design and development. On the contrary, BPMN, together with a few standards belonging to the Business Modelling specifications section, has been proposed with the explicit aim to support a business-centric perspective.

In particular, BPMN focuses on operational business processes (the set of activities that jointly contribute to the provision of a product or a service of an organization). BMM (*Business Motivation Model*) "provides a scheme or structure for developing, communicating, and managing business plans in an organized manner" (OMG 2015b), thus incorporates a 360° view of the organization, at a higher level of abstraction than BPMN. VDML aims at providing "a standard modeling language for analysis and design of the operation of an enterprise, with particular focus on the creation and exchange of value" (OMG 2015a). In this sense, VDML represents the attempt to formalize the level of abstraction of an organization that in the management literature is known as "strategic level".

1.2 OMG specifications and DiDIY

Within the DiDIY Project, several deliverables already published, and in particular three of them:

- D3.1, "Research model";
- D2.4, "Knowledge framework, revised version";
- D3.2, "Integrative modelling report (work and organization)"

proved that the DiDIY phenomenon, although evident in several social and economical domains, is proceeding at a lower speed than what we hypothesized in the Project proposal.

Several Project deliverables show DiDIY as an emerging phenomenon, still in its introductory stage and still very dynamic in its evolution. In particular, D3.6, "Reviewed research model", provides a collection of evidences that the DiDIY impact on organization and work can vary and has a different degree of maturity from a context to another, but – overall – is still limited.

This unanticipated immaturity of the subject of our research makes it unfeasible, so far, to reach any definite general conclusion about the properties of DiDIY. As a consequence, it would be merely unrealistic to attempt to formalize DiDIY (or its impact on organization and work) into a new OMG specification, following the strict constraints of the template summarized above, and with no possibility to provide an implementation plan (a pre-requisite for approval of any proposal of an OMG specification).

For the same limitations in the generalizability of the available results, the research carried out so far does not support the possibility to extend or adapt an already existing specification, even the ones which model organizations at a high level of abstraction. The business-centric OMG specifications such as BMM and VDML are reasonable candidates to do so, but require a degree of formalization that is not compatible with what has actually emerged from the research.

Nevertheless, we can exploit the versatility of UML to provide a formalization of the research results of WP3. In fact, the structure itself of the Research model presented in D3.1 and of its Review in the D3.6 is coherent with the type of representation provided by one of the most popular conceptual tools included in the UML specification: Use Case Diagrams.

1.3 Unified Modeling Language (UML)

"UML is a family of graphical notations that are based on a single metamodel and are used to support the description and design of software systems" (Fowler 2004). UML is a meta-model, i.e., a set of rules, constraints and theories used for modelling a class of problems, where a *model* is an abstraction of the real world.

UML is a visual language easy to learn, but semantically rich. The UML syntax is expressed in the UML Notation Guide, that describes the rules by which the elements of a language are assembled in expressions, the modelling elements (classes, interfaces, use cases, etc), relationships (associations, generalizations, etc), the diagrams (classes, use cases, interaction, etc). UML Semantics describes the rules by which the syntactic expressions are assigned a meaning.

From a technological point of view, one of the winning characteristics of UML is the fact that it incorporates the best experience of software development at the industrial level, thanks to the contribution of the partners of the OMG consortium. Moreover, UML is a flexible tool that adapts to most production systems, as well as the modern software systems development needs (scalability, architecture distribution, concurrence, etc). Finally, UML is independent on programming languages.

The versatility of UML is evident also from the various ways in which it can be employed. The most common use of UML, in the so-called forward engineering, is to sketch. The system (or part) is described by UML diagrams before the drafting of the code. UML enables to plan the architecture of the software and the UML diagrams can be part of the document of Technical Specifications of the system. In reverse engineering, UML diagrams are built starting from the software code. In this case UML allows to document and describe portions of the system (Niemann 2004).

UML allows modelling a "system" (in its broader sense, i.e., it can be an organization, or a portion of an organization, or a computer-based system) from a variety of perspectives, that correspond to as many diagrams. The perspective at the higher level of abstraction / generality corresponds to the Use Case Diagrams.

1.4 UML Use Case Diagrams and DiDIY

Use Case Diagrams are commonly referred to as behaviour diagrams. They describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

The specifications of these diagrams can be found on the OMG website (OMG 2015c), while Fowler (2004) provides a synthetic introduction to the UML standard.

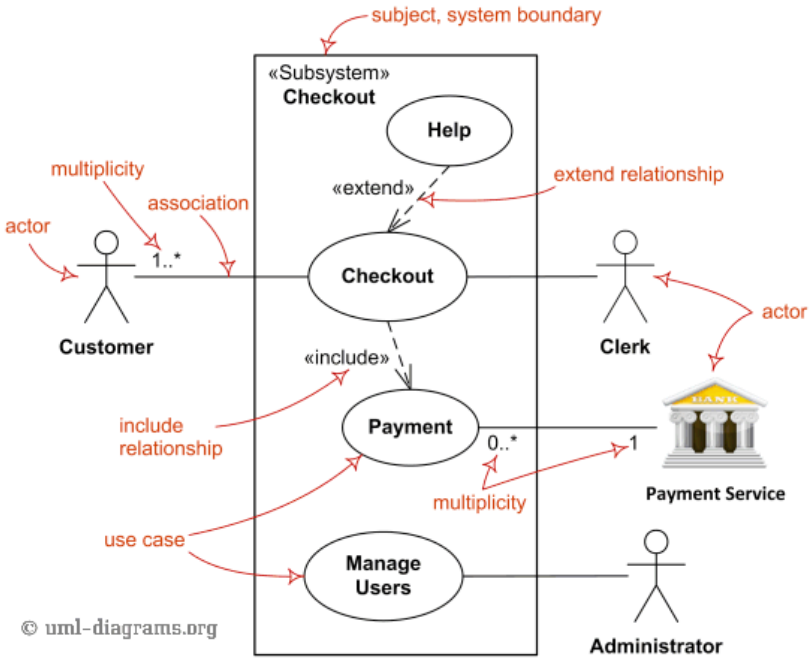


Figure 1 – Elements of UML Use Case Diagrams: actor, use case, subject, include and extend relationships (source: uml-diagrams.org).

Although business modelling was declared as one of the goals of the UML, UML specifications provide no notation specific to business needs. However, the term *business use cases* was introduced in Rational Unified Process (RUP) to represent business function, process, or activity performed in the modelled business. Given the absence of any specific standard regarding the business domain of application of use cases, business use cases can be informally defined as a particular type of use case, where a business actor represents a role played by some person or system external to the modelled business, and interacting with the business. A business use case should produce a result of observable value to a business actor.

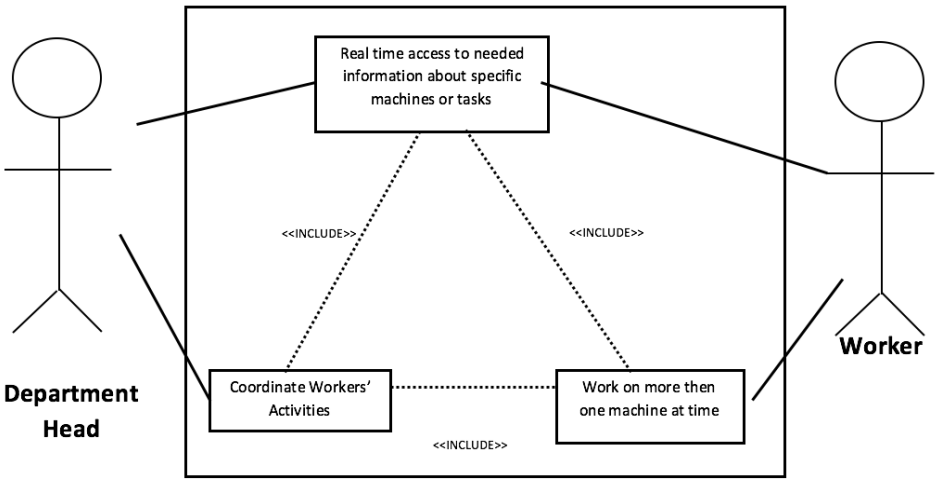
2. Application of UML Business Use Case Diagrams to the outcomes of WP3

The discussion of the previous section leads to conclude that UML Use Case Diagrams are the OMG specification that at the moment better suits the objective of formalization of the DiDIY phenomenon, as well as the current state of the research.

In other terms, the semantics of Business Use Cases is compliant with the twofold definition of DiDIY: *as a mindset* – thus characterizing a business actor, an individual playing a certain organizational role –, and *as an activity*, thus a business use case, a set of activities producing a result of observable value. Under this assumption, we could associate to each Research Topic (RTs, as presented in D3.6) one or more business use cases, where the main business actor is represented by the DiDIYer subject of the RT. The following sections present the result of this formalization.

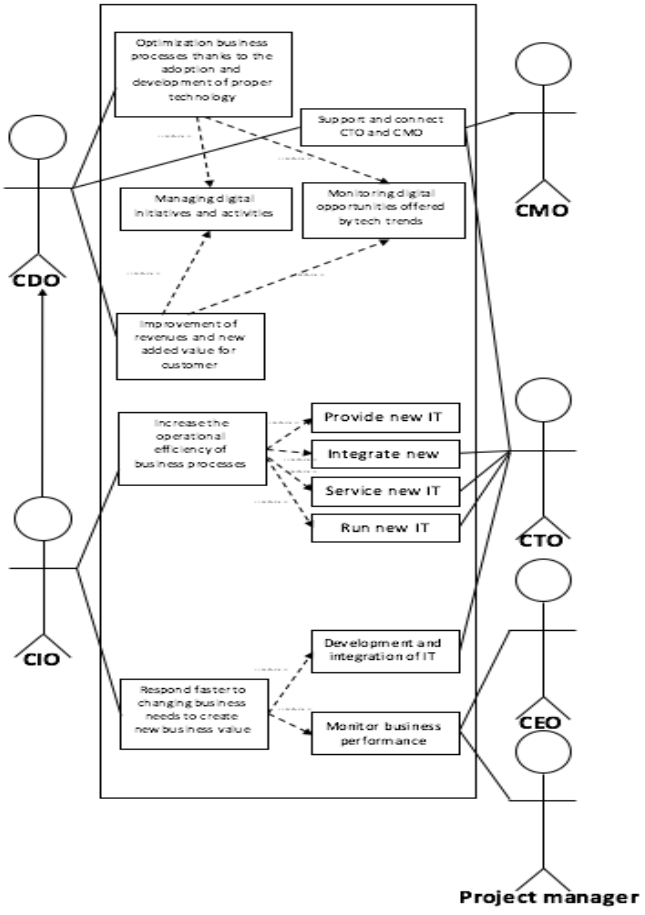
2.1 RT1: Worker in a manufacturing enterprise

Use case description: management of manufacturing operations
Actors: worker, department head
Main flow: The worker exploits digital technology to get real time access to information regarding the process. This allows him to a) take decision at the operational level such as sort term programming of the production, b) perform tasks on multiple machines at the same time. The head of the department is freed of some of the short term management tasks and fully dedicates his job on taking decisions at a higher level of responsibility and monitoring outcomes of the production process.
Secondary flow: The worker is not allowed to perform managerial tasks, therefore the information available to her through digital tools is used to perform multiple tasks on multiple machines, limitedly to increase the efficiency of the process.



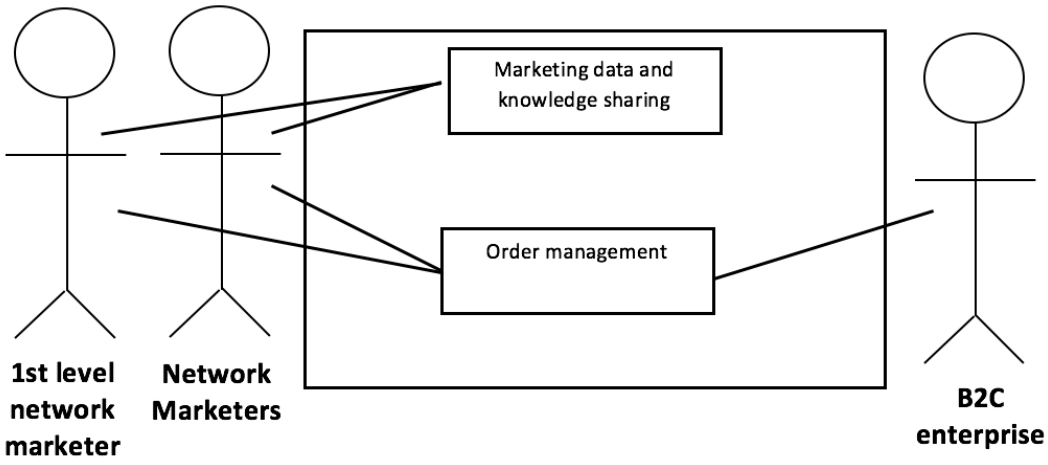
2.2 RT2: Manager

Use case description: management of digital innovation, in conjunction between the new role of the Chief Digital Officer (CDO) and Chief Information Officer (CIO)
Actors: CDO, CIO, CEO, CTO, Project Manager, CMO
Main flow: The CDO is mainly involved in three activities: optimization of processes through the adoption and development of appropriate technologies; increasing revenues and improving the perceived value by consumers; thirdly, it has the task of creating liaisons between the CMO and CTO and supporting them. The CIO is mainly involved in two activities. The first is aimed at increasing the efficiency of business processes, and to do that collaborates with the CTO to provide and integrate new IT resources. The second, however, aims to create new value through the response to changing business needs through both the development and IT integration (the CTO) and performance monitoring (the results of which are useful for both the CEO possibly for a Project Manager)
Secondary flow: The CDO activities are based on the monitoring of tech trends in order to assess what possible technologies can be adopted and the management of all "digital" initiatives and activities necessary to achieve the objectives



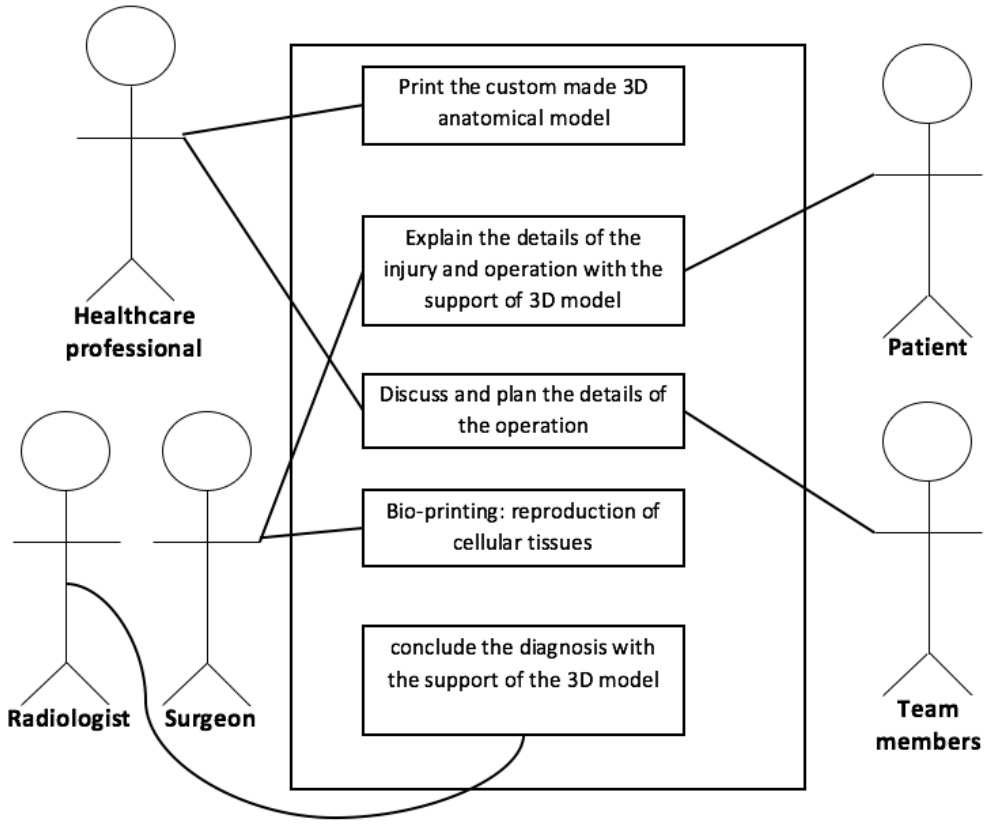
2.3 RT3: Networker

Use case description: order management through networking
Actors: network marketers; 1st level network marketer; B2C enterprise
Main flow: The 1st level networker shares information about the products and instructs the networkers in her network about how to expand their own network. The B2C enterprise manages the orders collected by the 1st level marketer through his network.



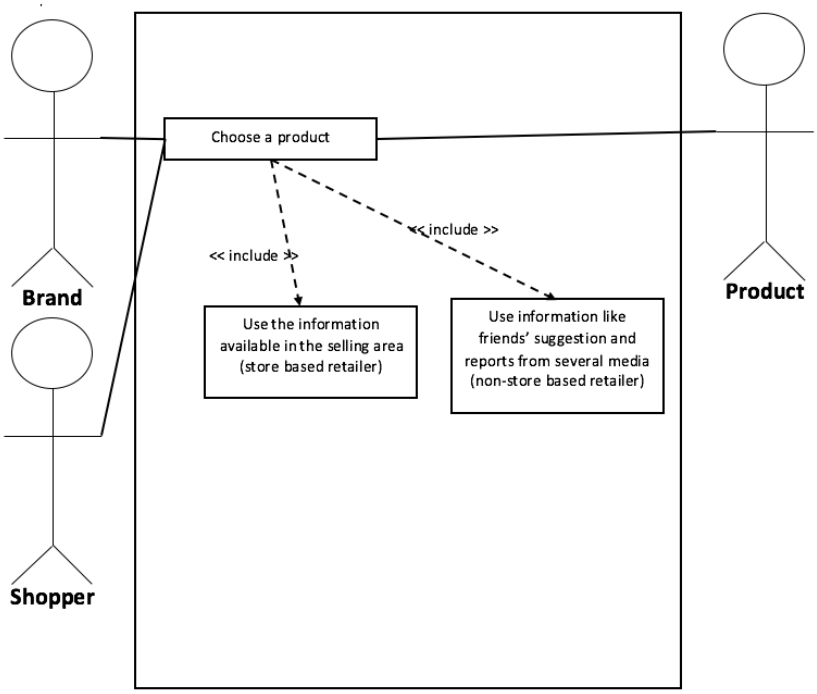
2.4 RT4: Healthcare professional

Use case description: diagnosis and choice of therapy
Actors: radiologist; surgeon; team members; patient
Main flow: "Healthcare Professional" refers especially to two actors: the radiologist and the surgeon. The first mainly uses 3D models in order to make the diagnosis. The second, exploits 3D models to plan with his team the operation and to clarify to the patient all the details regarding this operation.
Secondary flow: Printing a 3D model requires a set of multidisciplinary knowledge (from data acquisition to real prototyping and through its different techniques). Healthcare Professionals, in fact, use the community (online and offline) in order to implement this model.



2.5 RT5: Shopper

Use case description: buying process
Actors: shopper, branded company, product
Main flow: When choosing a product, the shopper bases her decision on several aspects: the range of the product, the brand (and the "shopping" activities). Depending on the type of shopper, these weighs in different measure the importance of these three aspects. A second characteristic of a shopper refers to the time it takes to search for a product. A third characteristic is the basis of information used for the choice: either available in the store, or a series of information provided from the outside, from friends on social media.



3. Conclusions

The UML use cases presented in this deliverable are the best attempt of formalizing – through an ISO standard – a phenomenon like DiDIY that is still in its introductory stages.

UML Use Cases, by definition, do not aim to provide a detailed representation of a certain organizational system, but contribute to define a high-level framework that can easily be discussed together with the experts of the modelled domain, and only later detailed, also using other modelling tools such as UML Activity Diagrams and UML Class Diagrams.

Therefore, besides being an ideal tool for the preliminary analysis of an emerging context, UML Use Cases are also a useful tool for dissemination purposes, to communicate the characteristics of this context to people who do not have a deep knowledge about it.

In this sense, the results presented in this deliverable can be used in the next stages of the DiDIY Project:

- to be enriched with additional use cases and more detailed descriptions of scenarios;
- to be integrated into policy patterns;
- in general, to develop recommendations and guidelines on how to DiDIY can reshape organization and work.

References

Fowler, M. (2004). UML distilled: a brief guide to the standard object modeling language. Addison-Wesley Professional.

Niemann, S. (2004). Executable Systems Design with UML 2.0. OMG Whitepapers on UML.

OMG (2008). Business Process Maturity Model (BPMM), OMG Document Number: formal/2008-06-01, (http://www.omg.org/spec/BPMM/1.0/PDF).

OMG (2015a). Business Motivation Model (BMM). OMG Document Number: formal/2015-05-19 (http://www.omg.org/spec/BMM/1.3).

OMG (2015b). Value Delivery Metamodel (VDML). OMG Document Number: formal/2015-10-05 (http://www.omg.org/spec/VDML/1.0).

OMG (2015c). OMG Unified Modeling Language (UML). OMG Document Number: formal/2015-03-01 (http://www.omg.org/spec/UML/2.5).

OMG (2016). About OMG (http://www.omg.org/gettingstarted/gettingstartedindex.htm).

Watson, A. (2008). Visual Modelling: past, present and future. Online, Object Management Group (http://www.uml.org).