Axiomatic Selection of Health and Social Care Web Services on the Basis of Use Cases

Chiara Parretti[1,*], *Fernando* Rolli[1], *Elaheh* Pourabbas[2], *Paolo* Citti[1]

[1]Department of Innovation and Information Engineering, Guglielmo Marconi University, Via Plinio 44 - 00193 Rome, Italy
[2]Institute for System Analysis and Computer Science "A. Ruberti", National Research Council, Via dei Taurini, 19 - 00185 Rome, Italy

Abstract, in recent years remote patient monitoring systems, which are conceived to monitor patient and store the recorded data, employ Service Oriented Architectures (SOA). In health care systems, if on the one hand the SOA features allow enriching the offer of services to patients and their families, on the other hand the problem arises of selecting the most appropriate set of web services for homogeneous groups of users or for each individual clinical case. In this paper, we propose a methodology based on the Axiomatic Design for the composition and selection of web services, which are based on the case stories studied, built using semi-structured interviews to different stakeholders involved in the care process. The proposed methodology detects in a systematic way all functional requirements by taking into account the different stakeholders involved in the process and their interactions. Thus, it identifies the interested actors for a specific health protocol, and consider all information to be exchanged and the Web Services to be implemented. The development of this methodology opens interesting scenarios even in areas not currently affected by such systems, such as health and safety at work for monitoring patients with chronic illnesses or for their work reintegration following acute clinical events. A specific case study is considered in order to illustrate the proposed approach.

1 Introduction

1.1 Context

In health care domain, the progressive aging of the population and the recent migration flows, in particular in Italy, are leading to a substantial increase in the costs of health services. Paradoxically, also, the progress in medical and scientific fields has increased the cost of individual health care services [1], which increasingly require the contribution of more specialists and advanced technologies [2]. It should also be noted that the increase in chronic illnesses and the lengthening of working life open important scenarios in the field of health and safety at work with regard to the monitoring and management of the worker suffering from chronic diseases. [3, 4]. For this reason, numerous approaches and strategies capable of optimizing the provision of health care services are proposed, see for instance [5-7]. Their main objectives are, in essence, to reduce waste, guarantee optimal levels of service for users, and to effectively prevent the emergency. In this context, the reduction of patient hospitalization length is an evident factor. In fact, this reduction involves saving resources in terms of more available beds, less care in hospital and personnel costs, abatement of social costs related to the management of a professional illness. Furthermore, the possibility of providing satisfactory home-based services allows the patient and his family to be assured of a better quality of life. This objective can be achieved by involving in a coordinated way all the operators and health and non-medical specialists involved in the treatment of the patient.

1.2 Service-oriented architecture

In recent years remote patient monitoring systems, which are conceived to monitor patient and store the recorded data, employ Service-Oriented Architecture (SOA). They create federated, distributed and scalable architectures, and are mainly based on web services [8]. SOA is a design approach based on the service concept. Service is component of distinctive functional meaning that typically encapsulate a high-level business concept. In this architecture, an application consists of software services and software service consumers (also known as clients or service requesters). This approach allows interoperability between different systems and technologies. Web services (WS) are self-contained, modular, distributed, dynamic applications that can be published or invoked over the network to create products and processes. These applications can be local, distributed, or web-based. This interoperability is due to the use of open standards.

1.3 Identification of web services

One of the challenging tasks in the SOA design is the identification of web services to be implemented. Accordingly, in health care systems if on the one hand the SOA features allow enriching the offer of services to patients and their families, on the other hand the problem arises of selecting the most appropriate set of web services for homogeneous groups of users or for each individual clinical case. Based on the description of the medical practices to be provided, it is possible to define the actors involved, the activities to be performed and the tools to be

[*] Corresponding author: c.parretti@unimarconi.it

used [9, 10]. Since the different actors involved can belong to different organizations, institutions and professional categories, the adoption of a SOA makes interoperability its strength, and enable patients and the different actors involved to access available web services. These requirements lead to the formalization of the conceptual design of the system, which allows the identification of web services candidates for implementation.

1.4 Design optimization

In this context, Axiomatic Design (AD) [11] can be used as a powerful tool to optimize the selective process [12-16]. An interesting application of AD in health care is the optimization of the patient flow in hospitals [17]. Essentially, this approach consists of decomposing the web services and their identified functional requirements to a more detailed level in order to highlight logical inconsistencies and suggest improvements to the initial design. In particular, the axioms of independence and information allow, at each level of decomposition, to identify the minimum set of web services to be implemented in compliance with the criteria of completeness, consistency and plausibility of the information to be processed. In this paper, we propose a methodology based on the AD for the composition and selection of web services, which are based on the case stories studied in the context of the Health@Home project [9]. In the mentioned paper, the case stories are built using semi-structured interviews to different stakeholders involved in the care process, and Web services are defined based on the business process modelling approach. Whereas, our methodology that is based on the AD paradigm detects in a systematic way all functional requirements by taking into account the different stakeholders involved in the care process and their interactions. The purpose of this work is two folds, i.e., firstly, to identify the interested actors for a specific health protocol, and secondly, to consider all those information to be exchanged and the web services to be implemented. Finally, a specific case study is considered in order to illustrate the proposed approach.

2 Methodology

2.1 Premise

In this paper we will refer to terms *case study*, *case story* and *use case*. A case study is a factual representation of what happened along with some analysis that provides insights and learning for the future. In the social and life sciences, a case study is a research method involving in-depth and detailed examination of a subject of study (the *case*), as well as its related contextual conditions [18, 19]. Whereas, a case story depicts what happened through people, place, and plot and brings emotional context into the portrayal of what happened. Finally, a use case is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modelling Language-UML as an actor) and a system to achieve a goal [20].

2.2 Methodological approach

An earlier work published by F. Pecoraro et alii [9] proposes to identify, through the use of case stories and Unified Modelling Language (UML) modelling, the web services to be implemented in relation to a specific clinical case. In this article, we propose an approach based on AD. The proposed methodology is based on the use of UML contextual modelling techniques and an axiomatic decomposition process of the functional requirements (FR) of the system. The main goal of methods of system analysis, which are based on use cases, is to capture from the real world the set of requirements needed for the system implementation. As a first step, it is necessary to generate the case stories, i.e. semi-structured interviews with the actors involved in the use of the system. The goal of the case stories is describing the functions of the system from the point of view of a particular user-actor involved. Subsequently they are grouped, analyzed and translated as use cases in Unified Modeling Language (UML) [7]. This level of formalization allows us to identify the interactions between the parts and to define an accurate diagram of the sequences. Already at this stage, we could identify the web service candidates for implementation, but the result would be strongly linked to the experience of software designers [9, 21]. Instead, we want to introduce a process to optimize the whole selection process based on quantitative evaluations [21, 22]. In this sense, AD approach provides us with a powerful evaluation tool, based on a solid theoretical basis, made up of axioms and corollaries, easily applicable also in this context [23, 24]. In fact, by reporting the relationships between the different components of the system in terms of project matrices, it is possible to keep track of the decomposition operations and assign a measurement to different decompositions, but always referable to the same level of detail. The integration of the two methodological approaches derives from the correspondence between the concept of FR of the AD approach and that of use case [22]. Therefore, initiating an axiomatic decomposition process, starting from the level of maximum abstraction, it is possible to obtain a logical process to the construction of the interaction diagram and the activities [22]. The decomposition can be pushed to the logical design of the system [22-27]. In fact, all levels of decomposition have a double representation, in terms of diagrams and in terms of matrices [26, 27]. While diagrammatic forms provide the tools to design and implement the system, representations of the matrix guide the identification of the most solid engineering solutions. Figure 1 schematically illustrates the whole process of design solution selection [28].

2.3 Modalities of application of axiomatic selection

The application of the axiomatic decomposition expected a mapping process between four conceptual domains

(Figure 2). In this context, the case stories represent the description in semi-structured language of the behavior of the system with respect to the various actors. To start the axiomatic selection we can place the case stories as the Customer Attributes (CA) of the process and the use cases identified as high level FR. Design Parameters (DP) are instead created by the designers and correspond to the interactions between the various uses case. In this way, the first level decomposition describes the network of interactions between use cases.

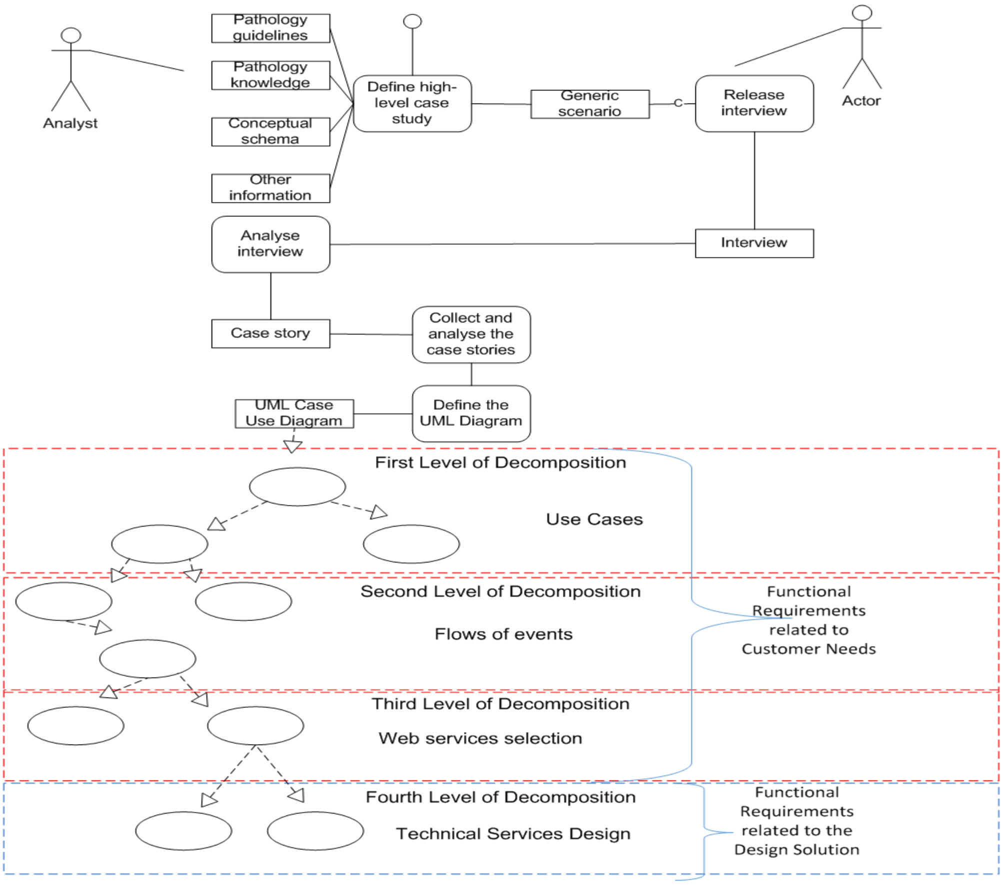

Analyst
Pathology guidelines
Pathology knowledge
Conceptual schema
Other information
Define high-level case study
Generic scenario
Release interview
Actor
Analyse interview
Interview
Case story
Collect and analyse the case stories
UML Case Use Diagram
Define the UML Diagram
First Level of Decomposition
Use Cases
Second Level of Decomposition
Flows of events
Functional Requirements related to Customer Needs
Third Level of Decomposition
Web services selection
Fourth Level of Decomposition
Technical Services Design
Functional Requirements related to the Design Solution

Fig.1. Axiomatic Selection Health and Social Care Web Services on the Basis of Case Stories.

Finally, we can define the process variables (PV) that, in this specific case coincide with the applications WS that implement the services to select. Therefore, in this paper web services (WS) are the design process variables (PV). They represent the tools for the implementation of the data communication system, described in a formal way using UML diagrams. The axiomatic design continues with successive decompositions, up to a level of detail that allows the implementation by the developers. In the last phase of decomposition, the decomposed FR are basic functions, while the design variables (DP) are data sets, corresponding to the objects of the upper classes [26, 27]. This decomposition can be extended to the process domain. This allows us to perform a mirror decomposition between design parameters (DP) and Process Variables (PV), i.e., a mirror decomposition between parameters of the construction of the two project matrices through a zigzagging process between three domains, as illustrated in Figure 2. The Process Domain allows us to identify the most appropriate technical solutions to meet customer needs. This zigzagging process has the advantage of focusing the designer's attention on FR [28]. The choice of solutions to be implemented takes place later, when the framework of functional requirements is clearer. The zigzagging process can initiate an iterative cycle if the identified solution has an impact on functional decomposition. Therefore, this process of cross-domain zigzagging is iterative.

2.4 Project constraints

During the axiomatic decomposition phase of the FR it is necessary to consider the constraints of the project. Constraints (Cs) are bounds on acceptable solutions. Such [23] has introduced two kinds of constraints: input constraints and system constraints. Input constraints are imposed as part of the design specifications. System constraints are imposed by the system in which the design solution must function. We can distinguish two classes of limitations in the implementation of web services in a health care setting (Figure 2). The first limitation concerns the regulatory restrictions related to the processing of health data (input constraints). In fact, these data are classified as highly sensitive. Therefore, the legislature has established a set of strict rules to protect patient privacy, where non-compliance is a criminal offense. The second limitation concerns non-functional constraints or system constraints [23]. For example, medical equipment must be as compatible as possible with the SOA, as to limit the manual intervention of communication of health parameters to the management platform. Figure 2 schematically illustrates the modalities of application of AD. The first limitation concerns the regulatory restrictions related to the processing of health data. In fact, these data are classified as highly sensitive. Therefore, the legislature has established a set of strict rules to protect patient privacy, where non-compliance is a criminal offense.

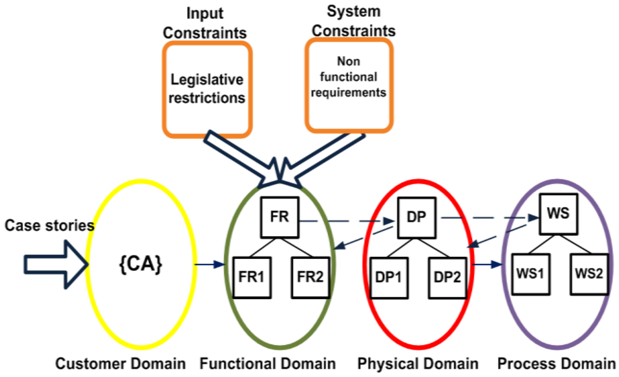


Fig. 2. Functional decomposition process.

The second limitation concerns non-functional constraints. For example, medical equipment must be as compatible as possible with the SOA, as to limit the manual intervention of communication of health parameters to the management platform. Figure 2 schematically illustrates the modalities of application of AD.

2.5 Considerations about the axiom of independence

This paper proposes the design of a robust web services selection system. It uses the FR selection approach based on the verification of the axioms of independence and information [29, 30]. The axiom of independence consists in trying to keep the identified FRs independent. In terms of matrix representation, it means that the relationships between FR and project parameters (DP) are represented by a diagonal matrix (Eq. 1). When the matrix is triangular, the design is called decoupled (Eq. 2). In particular, a quantitative parameter R (Eq. 3), called Reangularity [22, 31] was introduced, capable of measuring the level of orthogonality of the matrix. In the presence of different design solutions of decoupled type, you need to select the solutions with a higher R [22, 31]. The sparse or rectangular matrices, on the other hand, present sub-optimal engineering solutions and are defined as coupled (Eq. 4). The validity of the independence axiom ensures compliance with the criteria of completeness, consistency and plausibility of the information to be processed.

$$\begin{bmatrix} a11 & \\ & a22 \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} a11 & \\ a12 & a22 \end{bmatrix} \tag{2}$$

$$R = \prod_{\substack{i=1,n-1 \\ j=1+i,n}} \left(1 - \frac{\left(\sum_{k=1}^{n} a_{ki} \;\; a_{kj}\right)^2}{\left(\sum_{k=1}^{n} a_{ki}^2\right) \;\; \left(\sum_{K=1}^{n} a_{kj}^2\right)} \right)^{\frac{1}{2}} \tag{3}$$

$$\begin{bmatrix} a11 & a21 \\ a12 & a22 \end{bmatrix} \tag{4}$$

2.6 Information axiom considerations

2.6.1 General aspects

The axiom of the information allows the selection of the most robust solution, in case of presence of more substantial decompositions the robustness of a software development project translates into the development of a software solution, consisting of logically consistent web services that must meet the requirements, with the minimum value of information content. Often in the field of software engineering, recourse to the information axiom is neglected, because it is problematic to quantify the information content of the systems. In this context, the axiomatic planning is limited to the verification of the consistency of the decompositions. The choice of the most suitable solution remains instead linked to the designer's experience and creativity. This can be technically dangerous, because it risks causing developmental delays due to subsequent iterations or the need to release procedures with anomalies that have not yet been resolved, which may have important consequences in the operation of the system. With the approach proposed in this paper, the designer will be able to evaluate among several functional decompositions, the solution that requires the least effort in terms of resources and development time [26, 27]. This is equivalent to choosing functional decomposition with the least information content. For this reason, it is proposed to measure the information content of the system by means of the Function Point (FP) counting technique, in [32, 33].

2.6.2 Function Point Analysis

WS are software objects made up of data and methods. In terms of Function Point Analysis (FPA), the data sets correspond to the data files i.e., Internal Logical Files (ILF) and External Interface Files (EIF). . Instead, the methods are the transactional functions [32, 33]. These findings can provide a measure in terms of function points for each FRi functional requirement. The ILF correspond to the data structures maintained (modified) by the system. The EIF correspond to data structures that are only read by the system. Transactional functions can result in the following operations:

- Including \ modifying \ deleting data in the chart (External Input - EI);
- Visualization of the information contained in the chart (External Inquiry - EQ);
- Data production through the information process contained in charts (External Output - EO).

The function point analysis method is performed as follows:

- Determine all functions of all function types (ILF, EIF, EI, EO, EQ);
- Rate the complexity of every function (Low, Average, High);
- Calculate the total unadjusted function point count, as seen in Table1.

The complexity is determined according to the following parameters [32, 33]:

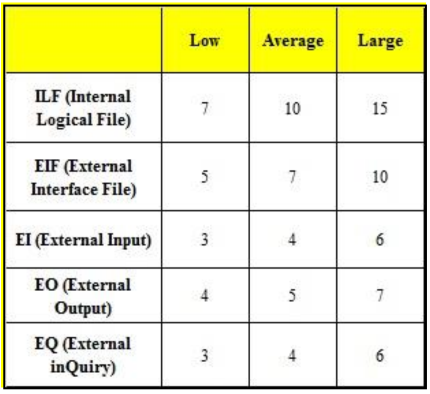
DET (Data Element Type): the field is not repeated, it is recognizable by the user. For the transactional functions, these are the fields where EI / EO / EQ have access in read / modify mode. For data files, it is the number of fields making up the structure.

FTR (File Type Referenced): For the FTR transactional functions, it is the number of data files or their subgroups to which EI / EO / EQ have access in read / modify mode.

RET (Record Element Type): recognizable (by the user) data subgroup within an ILF / EIF. A data structure may be composed of homogeneous subsets of data.

Table 1 summarizes the values in function points in the various cases.

Table 1. Count FP matrix.



	Low	Average	Large
ILF (Internal Logical File)	7	10	15
EIF (External Interface File)	5	7	10
EI (External Input)	3	4	6
EO (External Output)	4	5	7
EQ (External inQuiry)	3	4	6

2.6.3 Productivity and robustness of the software design

In fact, the measurement of the information level in terms of FP of the selected decomposition allows us to define the productivity of the software implementation. Moreover, the use of algorithms such as Constructive Cost Model II (COCOMO II) allow to estimate the time and resources needed for the project compared to the FP measurement of the selected decomposition [34-36]. Indeed, COCOMO II allow determining an estimate of the time required to complete the project. Then, the knowledge of a particular type of project allows categorizing the estimated man/resources by their professional profile, and matching them to the various phases of the software's life cycle. Therefore, the concept of robustness in AD terms translates into selecting the minimum set of web services, which requires less human resources for equal functional requirements. Figure 3 illustrates the process of applying the axiom of independence and information for a specific level of functional decomposition. This process is iterative. It could be prolonged for a long time. This would mean increasing the costs of the project and endangering its realization. This leads to a trade-off instance, as to whether to release the application first -implying a high risk of non-conformities for the final customer- or to continue with the analysis activities. For this purpose, Benavides [37] have introduced in AD a new concept of quality, which is based on the following formula: Quality = FR satisfied / resources spent.

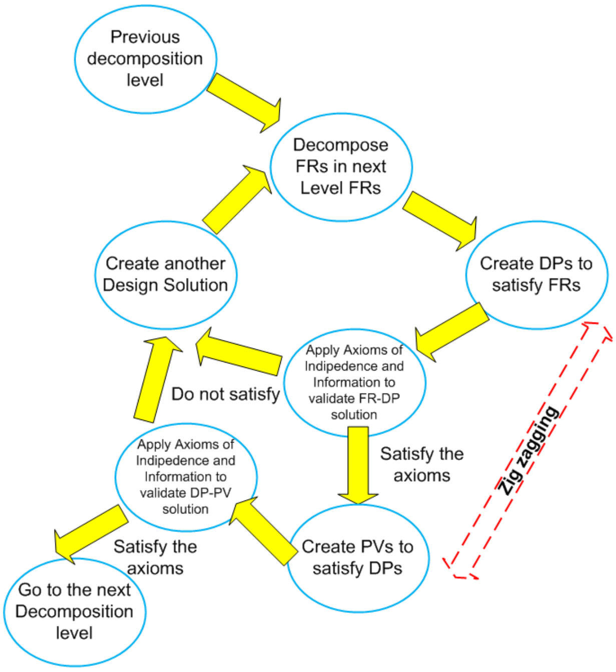

Previous decomposition level
Decompose FRs in next Level FRs
Create DPs to satisfy FRs
Create another Design Solution
Apply Axioms of Indipedence and Information to validate FR-DP solution
Do not satisfy
Satisfy the axioms
Zig zagging
Apply Axioms of Indipedence and Information to validate DP-PV solution
Satisfy the axioms
Create PVs to satisfy DPs
Go to the next Decomposition level

Fig. 3. Application of axioms of independence and information.

3 Case study

3.1 Clinical case

To illustrate the application of the proposed methodology we refer to the clinical case represented by a patient suffering from hypertension, which is studied in [7]. It follows the general scheme of Figure 1.

3.1.1 Definition of a general scenario

In the level of maximum abstraction of functional requirements, the general scenario of Figure 4 can represent the prescribed medical treatment. A scenario describes the software at a high level and give a rationale for each feature of the system existing. Instead, the use cases give a detailed account of what each feature does. A use case is a set of actions or event steps to achieve a specific goal. The use case actor can be a human or other external system [38, 39]. In Figure 5 each use case of the scenario is related to the actors involved in its execution. In this Figure, on the top-right side, we insert the web services coordination platform. For practical reasons, from now on, we will call this infrastructure simply Platform.

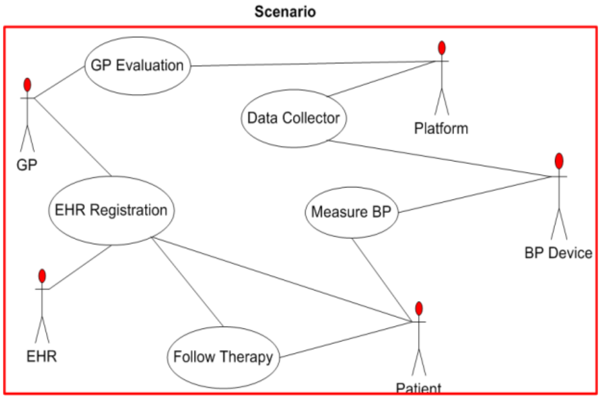


Fig. 4. Use case describing the general treatment of hypertension.

3.1.2 Health protocol analysis

The general scenario of the case study can be inferred, as suggested by F. Pecoraro et alii [9], through the analysis of the health protocol for the pathology under examination. This allows us to identify the actors involved in the treatment. With the same identified actors, the case stories are defined by using semi-structured interviews. This allows us to draw the information flows between the different parts. In designing an information system, we must start from defining the application limits, that is, the boundaries that specify the scope of the services to be implemented. If we consider the clinical case of hypertension control we can, at least in a first phase, consider only two actors: the patient and the general practitioner. We can collect the stories of both actors in the form of semi-structured interviews.

3.1.3 Collaboration diagram

By putting together the two interviews, we can schematize the execution of the process that involves them both. This allows us to define the general scenario of the system (Figure 4). The collaboration diagram of use cases (Figure 5) [38, 39] describe general scenario of the system. In summary, we can say that this two-actor process is activated by the periodic measurement of pressure by the patient in the home environment. Every 10 days a report is sent to the Platform, which is responsible for sending it to the general practitioner. In Figure 5 the red arrows represent this data flow. These arrows are interactions (also called collaborations) among the following use cases: “Measure BP”, “Data Collector” and “GP Evaluation”. The general practitioner (GP) analyzes the report. If he/she considers abnormal the values of pressure, he/she can change the dosage of the drug, can prescribe another drug, or can prescribe a visit with the specialist doctor. These prescriptions are recorded in the patient's health record (EHR). This event generates the sending of a message (via email and text message) to the patient. The patient signs in to his EHR account and picks up the prescription. In Figure 5 the green arrows represent this data flow. These arrows are interactions (also called collaborations) among the following use cases: “GP Evaluation”, “EHR Registration” and “Follow Therapy”.

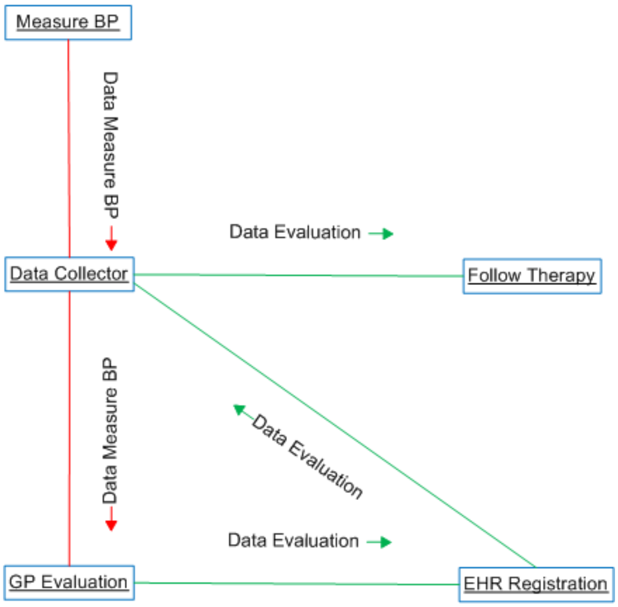


Fig. 5. Collaboration diagram describing the general treatment of hypertension.

3.1.4 Extensionality of the general scenario

Figure 4 shows the clinical case of pressure control as the only interaction between patient and general practitioner. We can define this scenario as a "restricted scenario". In reality, other actors can also participate in the process. The inclusion of new actors involves the execution of new activities (new use cases) and the intensification of communications between the parties. For simplicity in this paper, we focus on the analysis of the "restricted scenario" (only two actors).

3.2 Functional decomposition

The classical techniques of modelling processes to be automated aim to capture functional requirements from reality. The process usually started by identifying the main actors and activities. Later we proceeded to define the communications between the parts and the elementary actions. Formalization can take place through structured languages, such as UML diagrams. Therefore, we pass from scenarios of representation of the general processes to schemes that are more detailed. In practice, traditional user analysis techniques perform functional decomposition. It proceeds from use cases up to the elementary functions to be implemented in terms of classes and objects [22]. The AD allows a further step forward. It allows you to translate the diagrammatic representations of the traditional approaches into matrices [23, 24]. In a context characterized by a strong intangibility, this step allows to apply quantitative measuring instruments derived from the algebra of matrices [22, 27] that otherwise would not be possible to apply. The whole process has an iterative nature. In the case of identifying critical issues, it is necessary to go back to the previous step to eliminate them, and then continue. For the clinical case introduced in the previous paragraph, we will limit, for reasons of simplicity, the process of decomposition at only two steps. In the first stage of decomposition to the collaboration diagram of the use cases of Figure 5, a specific project matrix will be constructed. The matrix representation of the collaboration diagram will be a quantitative assessment tool, which follows the completeness, consistency and plausibility criteria of the information flow. In the second phase of decomposition, the collaboration diagram is decomposed into a sequence diagram It describes the activities and the relationships between them. The dual representation in terms of the second level project matrix will allow to select the web services to be implemented. Respecting the axioms of independence and information will ensure that the collaboration diagram in Figure 5 can be represented in terms of the project matrix as in Table 1. In this case, the rows represent the various use cases; the columns constitute the collaborations between use cases (interactions) [22]. An “X” at the cell relating a given use case and a collaboration indicates that this collaboration realizes this use case. Similarly, an “X” at the cell relating a given use case and a collaboration of another use case indicates that the realization of this use case requires a method of a software object (class) used in this interaction identified design solution is robust. They are described below.

3.3 First level decomposition

This means that these use cases are dependent one another [22]. For example, the use case “FR_2 Data Collector” depends on the realization of the use case “FR_1 Measure BP”. We will see that the use case “FR_2 Data Collector” can be activated by the "Send (Set_BP)" method of the "DP1 Collaboration Measure" (Figure 7). This indicates that this use case can be activated, only when the "Send (Set BP)" method records the patient's periodic data in the Platform. This means each of these methods constitutes the transactional functions of web services to be implemented.

3.3.1 Application axiom of independence

The project matrix of Table 1 is diagonal lower (decoupled). For this reason, we can say that the first level decomposition respects the axiom of independence. However, the process allows of the iterations. In fact, in the presence of decoupled matrix representations we should have revised the general scenario of use cases. This would have meant reviewing the definition and processing of customer needs.

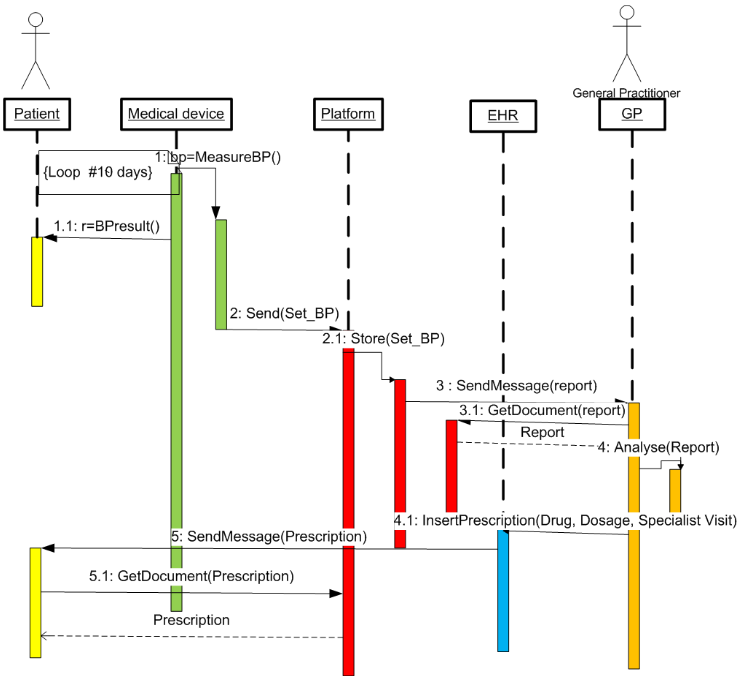
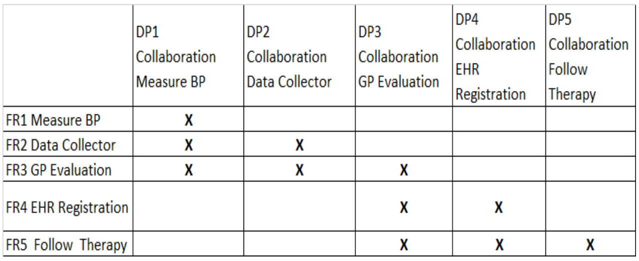


Fig. 6. Sequence diagram related to the use case that describes the general treatment of hypertension.

The iteration process continues until the design solution is satisfying. From engineering point of view, the validity of the axiom of independence guarantees us the respect of the criteria of completeness, consistency and plausibility of the information flow based on quantitative elements. In fact, the axiom of independence provide us not only with a visual tool to evaluate the quality of a decomposition (based on the structure of the project matrix), but also a tool to measure the independence of the functional requirements, based on the degree of regularity of the project matrix. In this case, using algebra of matrices, we have that an identity matrix has reangularity = 1. The decoupled matrices have a reangularity between 0 and 1. Therefore, in the presence of multiple scenarios of alternative use cases we can favor the one with a reangularity closer to 1. In this case, the matrix of Table 2 has reangularity R = 0,215166 (Eq. 3). In general, the level of functional independence can be increased by inserting a new use case. It leads to a decoupling of functional dependencies, increasing the degree of reangularity.

Table 2. Project matrix related to the case that describes the general treatment of hypertension.



	DP1 Collaboration Measure BP	DP2 Collaboration Data Collector	DP3 Collaboration GP Evaluation	DP4 Collaboration EHR Registration	DP5 Collaboration Follow Therapy
FR1 Measure BP	X				
FR2 Data Collector	X	X			
FR3 GP Evaluation	X	X	X		
FR4 EHR Registration			X	X	
FR5 Follow Therapy			X	X	X

However, the introduction of a new use case could increase the level of complexity of the system. This consideration introduces us to the next verification.

3.3.2 Axiom application of information

The axiom of information suggests us to select low complexity design solutions. The FP technique allows us to estimate the size of the software. This measurement represents the end user's point of view and is all the more accurate as the project details are evident. For high abstraction levels, we can consider derived estimation techniques such as the Early and Quick Function Point [35]. Therefore, the application of axiom results in the selection of design solutions with a lower value in terms of information with the same functional requirements [25, 26]. Moreover, always with the same defined functional requirements, the designer could find himself in the condition of having to evaluate if in the presence of alternative decompositions favor the design solution with greater functional independence (greater reangularity) or the one with less complexity. Naturally, the answer depends on the particular context.

3.3.3 The web services matrix

The mapping process between Physical Domain and Process Domain leads us to the definition of the so-called web services matrix. This matrix is mirrored to the matrix of project of Table 2, by effect to the process of zigging interdomain. It illustrates the dependencies between the interactions with the previously defined use cases and web services to be implemented. The interactions are the rows of the matrix, while the web services are the columns. The Process Domain allows the designer to assign to the functional requirements a mode of realization. The zigzagging process is an iterative process. For this reason, the choice of an implementation may require the reformulation of functional requirements. This means that functional decomposition takes place simultaneously on two domains. Therefore, if we wanted to report the web services in terms of functional requirements, it would be enough to make product between the matrix of project and the WS matrix

3.4 Second level decomposition

The use case diagram of Figure 4 and the collaboration diagram of Figure 6 can be used to implement the sequence diagram of Figure 6 [9]. This last diagram represents the sequence of activities related to the relationship between patient and general practitioner (GP) as regards pressure control [38, 39]. This diagram presents a series of more details in terms of flow and activities to be performed. In addition, the diagram of sequence has a representation in terms of the project matrix (Table 3). The previous use cases have been decomposed into activities and the activities inserted on the rows of the matrix. On the columns, instead, we insert the collaborations (interactions between activities) for specific activities. This new representation provides us with the tool to verify if the modelling performed is optimal or presents critical points in terms of completeness, consistency and plausibility of the information flow. For this reason, we can apply the axioms of independence and information.

3.4.2 Application axiom of independence

The structure of the project matrix in Table 3 respects the axiom of independence. For the sake of brevity, we will not elaborate on the measurement of the matrix reangularity. The same considerations as in the previous paragraph apply. So it is possible not only to verify that the identified activities are independent, but also, the degree of independence. The process structure is always iterative, so it is possible to improve the modelling of the activities.

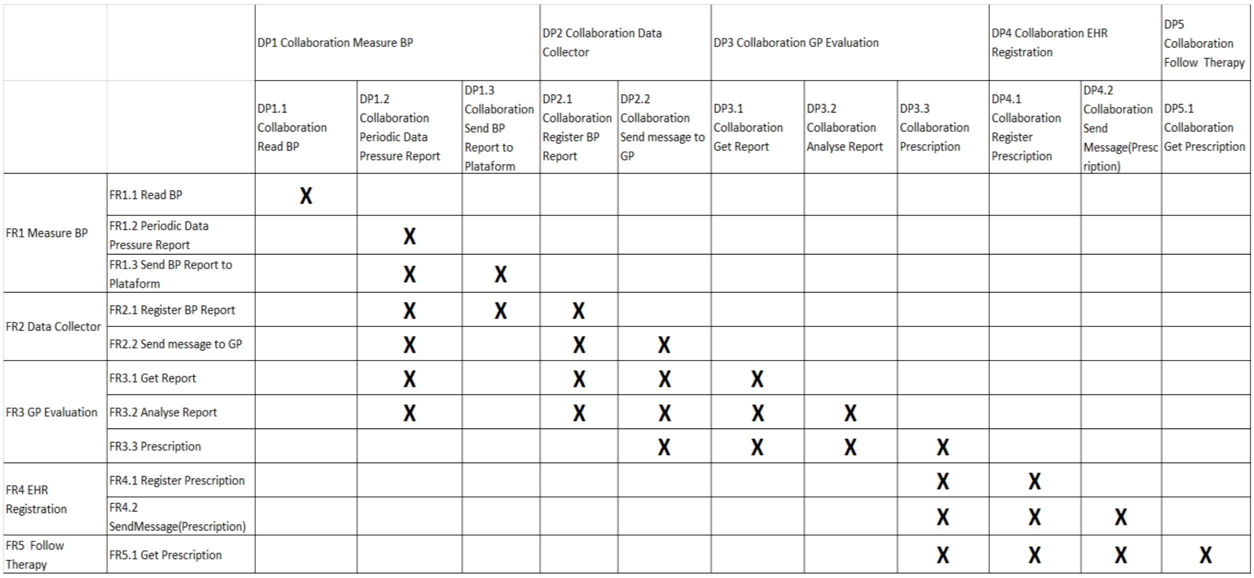
3.4.3 Axiom application of information

The activities that have been identified in this phase can be calculated in terms of the function point. At this level, we have the information to proceed with the calculation. The size of the system provides us with a tool to evaluate its complexity. Furthermore, this measure allows us to plan the effort needed to implement the system.

3.4.4 The web services matrix

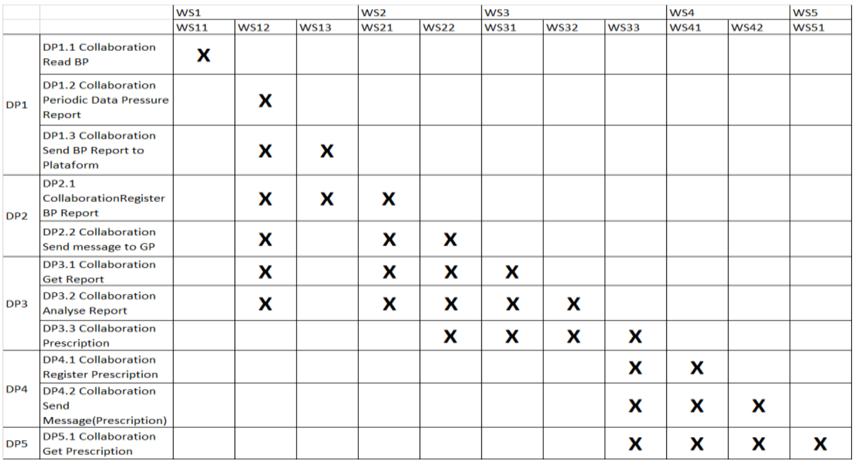
The mapping process between Physical Domain and Process Domain leads us to the definition of the matrix of second level web services (Table 4). This matrix is mirrored with respect to that of Table 3. It is obtained through a zigzagging interdomain process and schematizes the dependencies between the interactions and the web services to be implemented. The interactions are the rows of the matrix, while the web services are the columns. This allows us to report the activities decomposed here in terms of web services. This new mapping can be obtained as a product between the project matrix and the web service level matrix.

Table 3. Project matrix related to the sequence diagram that describes the general treatment of hypertension.



		DP1 Collaboration Measure BP			DP2 Collaboration Data Collector		DP3 Collaboration GP Evaluation			DP4 Collaboration EHR Registration		DP5 Collaboration Follow Therapy
		DP1.1 Collaboration Read BP	DP1.2 Collaboration Periodic Data Pressure Report	DP1.3 Collaboration Send BP Report to Plataform	DP2.1 Collaboration Register BP Report	DP2.2 Collaboration Send message to GP	DP3.1 Collaboration Get Report	DP3.2 Collaboration Analyse Report	DP3.3 Collaboration Prescription	DP4.1 Collaboration Register Prescription	DP4.2 Collaboration Send Message(Prescription)	DP5.1 Collaboration Get Prescription
FR1 Measure BP	FR1.1 Read BP	X										
	FR1.2 Periodic Data Pressure Report		X									
	FR1.3 Send BP Report to Plataform		X	X								
FR2 Data Collector	FR2.1 Register BP Report		X	X	X							
	FR2.2 Send message to GP		X		X	X						
FR3 GP Evaluation	FR3.1 Get Report		X		X	X	X					
	FR3.2 Analyse Report		X		X	X	X	X				
	FR3.3 Prescription					X	X	X	X			
FR4 EHR Registration	FR4.1 Register Prescription								X	X		
	FR4.2 SendMessage(Prescription)								X	X	X	
FR5 Follow Therapy	FR5.1 Get Prescription								X	X	X	X

Table 4. WS matrix related to the sequence diagram that describes the general treatment of hypertension.



		WS1			WS2		WS3			WS4		WS5
		WS11	WS12	WS13	WS21	WS22	WS31	WS32	WS33	WS41	WS42	WS51
DP1	DP1.1 Collaboration Read BP	X										
	DP1.2 Collaboration Periodic Data Pressure Report		X									
	DP1.3 Collaboration Send BP Report to Plataform		X	X								
DP2	DP2.1 CollaborationRegister BP Report		X	X	X							
	DP2.2 Collaboration Send message to GP		X		X	X						
DP3	DP3.1 Collaboration Get Report		X		X	X	X					
	DP3.2 Collaboration Analyse Report		X		X	X	X	X				
	DP3.3 Collaboration Prescription					X	X	X	X			
DP4	DP4.1 Collaboration Register Prescription								X	X		
	DP4.2 Collaboration Send Message(Prescription)								X	X	X	
DP5	DP5.1 Collaboration Get Prescription								X	X	X	X

3.5 Selection of web services

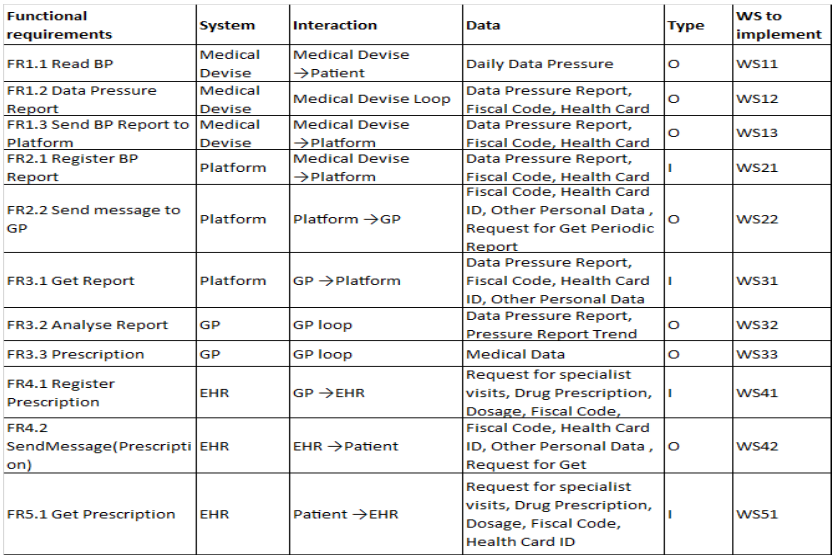
The web services defined in Table 4 represent the minimum set of services necessary to implement the system of management of communications between patient and general practitioner for the control of hypertension. The axiomatic decomposition process allows to directly associating each of them with specific functional requirements. Depending on the level of decomposition, the decomposed functional requirements are the use cases or specific activities or parts of them. This means that functional decomposition ensures that the case stories prepared in the analysis of user requirements are effectively translated into web services. Furthermore, the selection obtained must be considered optimized because the axioms of independence and information make it possible to provide quantitative measurements of the degree of independence of web services and their informative complexity. Table 5 shows the identified web services. They are broken down by system of competence according to information, communication direction (Interaction), and type of operation. The operations that the web service can perform are writing on archives (I), or reading (O).This information allows us to measure the effort in functional points of the system, but also to plan the other activities of the software life cycle, from the implementation to the testing and testing activities [36, 27]. It should also be noted that the development of a system of this type involves the allocation of implementation activities among the management teams of different systems. This means that the project documentation must be understandable and up-to-date at every stage of the software life cycle, to avoid delays or malfunctions in the services. In this sense, AD allows the preparation of very detailed and easily understandable documentation, even at the logical design level of the system.

4 Conclusions

The SOA allows the implementation of web services on interoperable platforms and systems, and create a virtual network of services to support patients in home therapy and health monitoring. For each specific clinical case, it is possible to define a series of services to be implemented, able to monitor the patient, manage the therapy, monitor his health according to the work activities and coordinate the activities of all the operators involved in the treatment and management of his healthHowever, the design of such networks poses important questions. For example, how can we capture user requirements for each specific case in a health context that may be evolving?

To answer questions of this kind it is necessary to focus on the quality of the services offered. In this paper, we proposed an approach based on the AD for the composition and selection of web services, which are based on the case stories studied, built using semi-structured interviews to different stakeholders involved in the care process.

Table 5. Table of web services selected for the general treatment of hypertension in the home system.



Functional requirements	System	Interaction	Data	Type	WS to implement
FR1.1 Read BP	Medical Devise	Medical Devise →Patient	Daily Data Pressure	O	WS11
FR1.2 Data Pressure Report	Medical Devise	Medical Devise Loop	Data Pressure Report, Fiscal Code, Health Card	O	WS12
FR1.3 Send BP Report to Platform	Medical Devise	Medical Devise →Platform	Data Pressure Report, Fiscal Code, Health Card	O	WS13
FR2.1 Register BP Report	Platform	Medical Devise →Platform	Data Pressure Report, Fiscal Code, Health Card	I	WS21
FR2.2 Send message to GP	Platform	Platform →GP	Fiscal Code, Health Card ID, Other Personal Data , Request for Get Periodic Report	O	WS22
FR3.1 Get Report	Platform	GP →Platform	Data Pressure Report, Fiscal Code, Health Card ID, Other Personal Data	I	WS31
FR3.2 Analyse Report	GP	GP loop	Data Pressure Report, Pressure Report Trend	O	WS32
FR3.3 Prescription	GP	GP loop	Medical Data	O	WS33
FR4.1 Register Prescription	EHR	GP →EHR	Request for specialist visits, Drug Prescription, Dosage, Fiscal Code,	I	WS41
FR4.2 SendMessage(Prescription)	EHR	EHR →Patient	Fiscal Code, Health Card ID, Other Personal Data , Request for Get	O	WS42
FR5.1 Get Prescription	EHR	Patient →EHR	Request for specialist visits, Drug Prescription, Dosage, Fiscal Code, Health Card ID	I	WS51

The purpose of the approach is to detect in a systematic way all functional requirements by taking into account the different stakeholders involved in the care process and their interactions. In this paper, we focused on the case of hypertension treatment, to which we applied our AD approach. The AD allows guiding the design based on quantitative measuring instruments, and define a dynamic and robust design mechanism, based on the dual representation of the system, in terms of UML diagrams and project matrices with multiple benefits. First, AD allows direct tracking between patient needs (CA), functional system requirements (FR), design parameters (DP) and implementation solutions (WS). This means that, by changing the needs of the patient, it is possible in a short time to start a readjustment of the system. Furthermore, the application of the axioms of independence and information is linked to the algebraic structure of the project matrix. Therefore, it is possible to derive parameters based on quantitative criteria to be attributed to the design solutions identified through the use stories technique, which the use of an iterative and incremental design improvement process. It facilitates the production of updated documentation at each stage of the software life cycle. This means facilitating the exchange of information, both between the members of the same team and between different teams of developers, with a view to interoperability between multiple technological and organizational systems.

As a future work, we plan to measure the effort needed to implement the composition and selection of web services, which are modelled according to the AD approach.

Acknowledgment

This study is part of the H@H (Health @ Home) – Smart Communities for the wellbeing of the citizens – project granted within the National Operational Programme for Research and Competitiveness 2007-2013. Identification code: SCN_00558.

References

1. G. Arcidiacono, J. Wang, K. Yang, (2015) *Operating room adjusted utilization study*, International Journal of Lean Six Sigma, Vol. 6, Issue 2; pp.111 – 137.
2. G. Arcidiacono, A. Pieroni, (2018) *The Revolution Lean Six Sigma 4.0*, International Journal on Advanced Science, Engineering and Information Technology, Vol. 8, Issue 1, pp. 141-149.
3. F. O'Hagan (2017). *Work, organizational practices, and margin of manoeuver during work reintegration. Disability and rehabilitation*, pp. 1-10.
4. I. Giorgi, G. Calsamiglia,, M. Negri, F. Scafa, R. Colombi, S. M. Candura, & O. Bettinardi. (2007). *Management of occupational stress among patients with cardiac diseases*. Giornale italiano di medicina del lavoro ed ergonomia, 29(3 Suppl), pp. 695-696.
5. G. Du , X. Liang, C. Sun, (2017) *Scheduling Optimization of Home Health Care Service Considering Patients' Priorities and Time Windows*, Sustainability 9, 253.

6. S. Landers, et al., (2016) *The Future of Home Health Care-A Strategic Framework for Optimizing Value*, Home Health Care Management & Practice, 28(4) 262 –278.

7. F. Wang (2012) *Measurement, Optimization, and Impact of Health Care Accessibility: A Methodological Review*, Annals of the Association of American Geographers, 102(5): 1104–1112.

8. K. Avila, P. Sanmartin, D. Jabba, M. Jimeno, (2017) *Applications Based on Service-Oriented Architecture (SOA) in the Field of Home Healthcare*, Sensors, 17(8): 1703.

9. F. Pecoraro, D. Luzi, E. Pourabbas, F.L. Ricci, 2017 *A methodology to identify health and social care web services on the basis of case stories*. Proceedings of E-Health and Bioengineering Conference (EHB).

10. S. Crowe, K. Cresswell, A.Robertson, G. Huby, A. Avery, A. Shikh, (2011) *The case study approach*, Medical Research Methodology, 1.

11. A.M. Farid, (2016) *An engineering systems introduction to axiomatic design*. In Farid AM, Suh NP (eds) Axiomatic design in large systems: complex products, buildings and manufacturing systems, Chap. 1, Springer, Berlin, Heidelberg, 1–47.

12. C. Monti, A. Giorgetti, A. Girgenti (2015) *An Axiomatic Design Approach for a Motorcycle Steering Damper*, Procedia CIRP, vol. 34, pp. 150–155.

13. F. Rolli, A. Giorgetti, P. Citti, (2015) *Integration of Holistic Non-Conformities Management and Axiomatic Design: a case study in Italian Income Tax Returns Management*, Proc. CIRP 34, 256-262 (2015);

14. A. Giorgetti, A. Girgenti, P. Citti, M. Delogu (2016) *A Novel Approach for Axiomatic-Based Design for the Environment*. In Axiomatic Design in Large Systems: Complex Products, Buildings and Manufacturing Systems, ed. Springer. doi:10.1007/978-3-319-32388-6_5.

15. G. Arcidiacono, K. Yang, J. Trewn, L. Bucciarelli, (2016) *Application of Axiomatic Design for Project-Based Learning Methodology*, Procedia CIRP, Vol. 53, pp. 166-172.

16. G. Arcidiacono, C. Brown, L. Bucciarelli, F. Melosi, (2016) *Axiomatic Design of Production Systems for Performance Improvement: A Project Identification and Prioritization Model*. In Axiomatic Design in Large Systems: Complex Products, Buildings and Manufacturing Systems, ed. Springer.

17. G. Arcidiacono, D.T. Matt, E. Rauch, (2017) *Axiomatic Design of a Framework for the Comprehensive Optimization of Patient Flows in Hospitals*, Journal of Healthcare Engineering, Volume 2017, 19 June 2017, Article number 2309265.

18. Z. Zainal, (2007) *Case study as a research method*, Jurnal Kemanusiaan bil.9.

19. B. Flyvbjerg, *Case Study* (2011). Norman K. Denzin and Yvonna S. Lincoln, eds., *The Sage Handbook of Qualitative Research,* 4th edition, (Thousand Oaks, CA: Sage, 2011), chapter17, pp. 301-316.

20. I. Jacobson, M. Christerson, JP. Onsson, G. Övergaard. (1992)*, Object-Oriented Software Engineering* - A Use Case Driven Approach, Addison-Wesley.

21. D.Tarenskeen, R.Bakker, S.Joosten, (2015) *Applying the V Model and Axiomatic Design in the Domain of IT Architecture Practice*; Procedia CIRP 34, 263-268.

22. P. Pimentel, C. Stadzisz, (2006) *A Use Case based Object-Oriented Software Design Approach using The Axiomatic Design Theory*. Proceedings of ICAD2006 Fourth International Conference on Axiomatic Design, 4:1-8.

23. N.P. Suh, Axiomatic Design - *Advances and Applications*, (Oxford University Press, 2001);

24. S.H. Do, N.P. Suh, (2000) *Object-oriented software design with axiomatic design*, Proc. CIRP 49, 278-84.

25. F. Rolli, A. Giorgetti, P. Citti, M. Rinaldi, (2016) *Information content evaluation to obtain robustness of the management in Italian fiscal process (Part 2): optimization of Italian income certification process of the year 2016*, Proc. CIRP 53, 63-69.

26. F. Rolli, A. Giorgetti, P. Citti, M. Rinaldi, (2016) *Improvement of the compilation process of the Italian income certifications: a methodology based on the evaluation of the information content (Part 1)*, Proc. CIRP 53, 56-62.

27. F. Rolli, C. Parretti, P. Citti, M. Rinaldi, (2017) *Functional case test design to optimize the software development in Italian tax processes (Part II): the F23 tax payment form*, MATEC Web Conf. Volume 127, 2017.

28. D.S. Cochran, W. Eversheim, G. Kubin, & M.L. Sesterhenn, (2000). *The application of axiomatic design and lean management principles in the scope of production system segmentation*, International Journal of Production Research, 38(6),pp. 1377-1396.

29. B. Pacifici, C. Parretti, A. Girgenti, and P. Citti, (2017) *Axiomatic Design for an Efficient Development of Optimized RPM Systems*, MATEC Web Conf. Volume 127, 2017.

30. L.E.S. Oliveira, A.J.Alvares , (2016) *Axiomatic Design Applied to the Development of a System for Monitoring and Teleoperation of a CNC Machine through the Internet*, Procedia CIRP 53, pp.198-205;

31. N.P. Suh, (1990) *The Principles of Design*, Oxford Series on Advanced Manufacturing.

32. A. J. Albrecht, (1979) *Measuring Application Development Productivity*, Proceedings SHARE/GUIDE IBM Applications Development Symposium, October 1979;

33. IFPUG - *Function Point Counting Practices Manual*, Release 4.1 - Westerville - Ohio, 1999;

34. T. Rollo, (2006) *Functional Size measurement and COCOMO–A synergistic Approach*. In: Proc. of Software Measurement European Forum. pp. 259-267.

35. R. Meli, L. Santillo, (1997) *Function point estimation methods: a comparative overview*, Data Processing Organization srl, IFPUG - Fall Conference - September 15-19,- Scottsdale, Arizona USA.

36. F. Rolli, C. Parretti, P. Citti, M. Rinaldi, (2017) *Functional case test design to optimize the software development in Italian tax processes (Part I): Methodology definition*, The 11th International Conference on Axiomatic Design (ICAD 2017).

37. E. Benavides, (2011) *Advanced Engineering Design,* Woodhead Publishing in Mechanical Engineering.

38. G. Booch, J. Rumbaugh, I. Jacobson, (1999) *UML User Guide*, Addison-Wesley.

39. M. Fowler, (2010) *UML Distilled*, Addison-Wesley.