Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 53 (2016) 56 – 62

www.elsevier.com/locate/procedia

The 10th International Conference on Axiomatic Design, ICAD 2016

Improvement of the compilation process of the Italian income certifications: a methodology based on the evaluation of the information content (Part 1)

Fernando Rolli*, Alessandro Giorgetti, Paolo Citti, Massimo Rinaldi

Department of Innovation and Information Engineering, Guglielmo Marconi University, Via Plinio 44 - 00193 Rome, Italy

* Corresponding author. Tel.: +39- 06- 377251; fax: +39-06 -377-25 647. *E-mail address:* rollifernando@gmail.com

Abstract

In recent years the Italian tax system has been affected by significant changes. Important legislative reforms have revolutionized the relationship between citizens and public administration. These changes tried to simplify the relationship between citizens and public administration, reducing the burden of bureaucracy on the lives of citizens and businesses. To simplify the completion of income statement taxpayers, the Italian Revenue Agency started to use precompiled statement from 2015. However, many manual actions are still necessary to correct or complete the statement. This paper focuses on the improvement of compilation and control of income certifications in order to reduce non-compliance poured into precompiled statements. The proposed method aims to improve the process by introducing a robust procedure based on Axiomatic Design that is able to quantify the information content of a software project by measuring the information content through the Function Point technique. The developed approach will be able to prevent the generation of non-compliances. In this way, to identify possible critical situations in a proactive way and to avoid classes of non-conformities, it will be possible to optimize the data compilation process to verify compliance with the technical specifications of the Italian Revenue Agency. The goal of the proposed approach is to simplify the collection of fiscal data and create a clearer path for taxpayers.

Keywords: Non-Conformities Reduction; Information axiom; Tax Services; Software Engineering; Function Point Counting

1. Background

The process of computerization in the Italian public administration is progressively reducing the gap between citizens and public administration. This improvement is particularly noticeable in the field of taxation. In 2015, with the "Stability Law", the Italian Government has already introduced pre-filled tax returns, which are made available by the Revenue Agency to taxpayers with an income from employment and pensions [1]. This is a historic breakthrough for the Italian Revenue Agency: from a fiscal system where taxpayers were obliged to submit an annual tax return form (based on the income received during the previous year and certified by their employer), it now has switched to a system where the Revenue Agency provides taxpayers with a pre-filled form. The Italian Revenue Agency acquires information about the citizens' income, the deductible expenses, and those expenses that can be deducted directly by the employers. Moreover, on April 15, Tax Authorities are releasing a pre-populated form over the internet using the information received. Taxpayers will be able to access their form using a personal PIN code, which will also allow access to the services provided by the Agency. Taxpayers can accept the pre-filled statement, thus avoiding any subsequent tax audit. The Revenue Agency will forward the debt or credit payment schedule to the employers, who will act as "withholding agents". If taxpayers contest the pre-populated form, they will be able to change it online or seek for advice at any chartered office for tax assistance.

2. Introduction

This article focuses on improving the process for preparation and control of income tax certifications in order to reduce non-conformities, which still persist and prevent the full automation of the entire system. For this reason, we propose the adoption of a software design methodology that leverages a decomposition approach of functional requirements based on the Axiomatic Design. This paper pays particular attention to the application of the axiom of information that is usually a critical issue in the development of industrial products [2-6] and transactional processes [7] but which is often overlooked by software designers [8]. This use is facilitated by adopting the estimate in function points of the information content of the decomposition to be evaluated. Thus, the designer may also select the solution, among the different functional substantial decompositions, requiring the least effort in terms of resources and development time. This approach is the same as choosing the functional decomposition with the lowest information content, meeting the same functional requirements at the same time. The present article describes the theoretical aspects of the proposed approach. A second article (Part II), on the other hand, illustrates the application of this approach on a real case example.

3. Operational environment

Despite progress in recent years, there are still some areas of inefficiency, mainly due to an incomplete computerization process in some sectors. Very often, the existing computer systems have already been designed to fulfill tax obligations other than those provided for by recent standards. Therefore, the reconversion of IT procedures faces a certain rigidity as well as operational limits, which need to be taken into account. Redesigning the management procedures of service delivery to citizens (salary transfers, pensions, insurance benefit management, etc.) is extremely onerous and therefore inapplicable. Thus, the most viable solution to the problem consists in implementing the system of compilation of income certifications in a contextualized management, in its operational environment. This contextualization results in producing income certifications that must meet the requirements as per the instructions of tax compilation produced by the Revenue Agency, taking into account, at the same time, the intrinsic characteristics of the operating environment. These intrinsic characteristics are formalized in this paper in clusters of non-conformities and design constraints. In this way, the process of compilation of income certifications can prevent spreading problems related to a particular operating environment, proceeding towards the precompiled tax declaration that is produced for the citizens by the Revenue Agency. Therefore, the production system of income certifications may not be a simple Data Entry provided by the conferring procedures. A design must be set for the compilation process of these certifications, so that it might have a proactive function of non-conformity resolution for those management procedures that may be generated, since they were designed according to tax rules that are no longer in force. For this purpose, the present article aims at presenting a methodology for implementation and maintenance of tax information systems, which can eliminate the non-conformity historical process and ensure conformity with the time constraints of data transmission to the Revenue Agency; on the other hand, it allows meeting the Agency's compilation rules. For this reason, we propose the adoption of a design methodology that makes use of a decomposition approach of functional requirements based on the verification of the axioms of independence and information. In this case, the functional requirements for the compilation process are defined taking into account both the instructions for certifications and the operating environment’s context features, by assessing its overall consistency.

4. General outline of development

In this article, we will refer to the Axiomatic Design of Object-oriented Software Systems (ADo-oSS) methodology [9]. This methodology allows the integration in a simplified manner of AD design with the implementation of software systems adopting the Object-Oriented Programming (OOP). AD is a top-down type of design approach. It narrows down starting from the general down to the particular. The Functional Requirements of the project (FR) are defined starting from the Customer Attributes (CA). At this stage, the Design Parameters (DP) are identified. If the level of analysis is too abstract, we proceed with a decomposition activity (FR). At this point, the corresponding (DP)s are identified. The process continues down to the level of detail needed for design purposes [9, 10]. The last stage of AD designing is actually the identification of a decomposition between FR and DP, which corresponds to the conceptual design of the system to be implemented. Therefore, the output of the AD design stage is the input for the system programmers. In fact, the OOP programming is Bottom Up: starting from the particular, it proceeds upwards to the general. The logic drawing of the system is the first phase of the software’s life cycle.

5. Building a robust process

Axiomatic Design is based on the verifications of the axioms of independence and information. The axiom of independence guarantees that consistent design solutions are those whose relationships between FR and DP are uncoupled or decoupled. On the contrary, the information axiom allows us to select, if various different substantial solutions are present, the most robust solution - which corresponds to the project with the lower information value. The concept of information is also understood as more likely to meet the Functional Requirements (FR) via the Design Parameters (DP). However, in the software design field, Axiomatic Design is often applied using only the verification of the axiom of independence. The solutions that are thus identified are consistent from a designing point of view, but are not robust in more conceptual sense. Scientific literature on this subject often justifies the non-application of the axiom of information by stating the difficulty in quantifying the information content of a software project [9]. This paper suggests an application method for the axiom of information by measuring the information content between FR and DP relations through the Function Point technique [11-13]. Thus, the design process of populating a system of income tax certifications turns out to be not only consistent from a logical point of view, but also robust. In this specific case, the robustness of the process is equivalent to implementing a reduced-complexity system, while ensuring the fulfillment of the functional requirements and the design restrictions (deadlines).

5.1. Integration with the payment system

The key prerequisite for the implementation of a robust system of compiling income certifications is that the elementary items regarding pay slips are all identified by univocal codes. Moreover, the certifications production system should have direct access to payment procedures. Information about payments and withholding taxes are meant to be stored on a monthly basis. These data are necessary in order to receive the process information in a short time. Furthermore, the input data must be elementary in order to enable a rapid re-aggregation in the case of law changes. Integration can be structured as interoperability between different systems within the same structure. In this case, the various information systems share objects / data (components). The interoperability permits to have access to procedures of conferring data in real time, as well as providing synchronous data concerning the assessment of duties to payment systems. The selection of software objects to be shared between the various systems may use a Component-Oriented Software Development methodology based on AD [14]. Such selection allows for the reuse of already existing applications, the assessment of compatibility and the reduction of the adjustment time of the procedures.

5.2. Management of non-conformity history

As far as complex organizations are concerned, recording Non-Conformity compilations of income certifications (NC) in specific structured databases can be a useful tool. The post-production service can analyze and categorize the non-conformities reported by users in a multilevel database, according to a HNCR approach (Holistic Non-Conformity Reduction) [15-18].

6. Designing a robust system of compiling income tax returns

The production process of income certifications can be summarized as in Fig.1. The user requirements (CA) are the instructions released by the Revenue Agency for completing the income certifications [15]. Likewise, the design Constraints (C) are the restrictions imposed by regulations, such as deadlines and the terms of electronic submitting of returns. NCs are the clusters of non-conformity defined during cataloguing and reclassification of historical non-conformities registered by the help desk. They have a proactive value. They are warnings for specific areas of concern and must be taken into account when preparing the annual income tax returns. The Functional Requirements of the system (FR) establish the project formalization of the compilation instructions, under the conditions imposed by design constraints, and follow the indications given by the analysis of non-conformity clusters. Design Parameters (DP) are the income and tax data to be processed and completed. The Process Variables (PV) are the implementation tools of this system, consisting of subroutines, compilers and other software components. Furthermore, the proposed decomposition adopts the classic zigzagging process. Thus, the decomposition of a complex problem starts with determination of the most general functional requirements. The FRs, DPs, and PVs must be decomposed down to a level of detail necessary to enable the implementation of the project by programmers. These hierarchies of FRs, DPs, PVs, and the corresponding matrices compose the system architecture. This decomposition cannot be done by remaining in a single domain. The zigzagging process yields a parallel decomposition of all three domains [9]. In this article, for simplicity we focus on the first two domains (FR, DP).

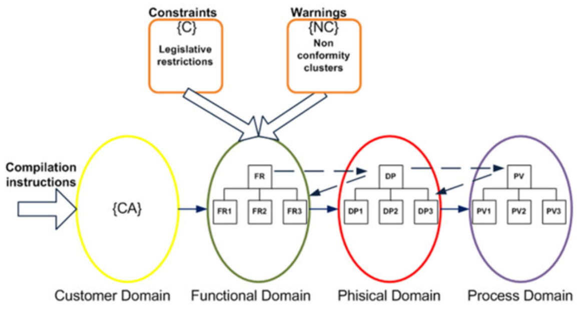


Fig. 1. General Process.

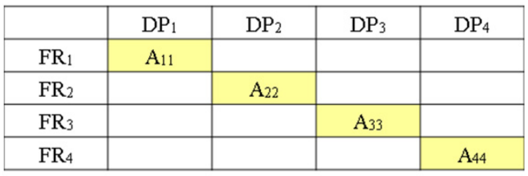
7. Methodology of functional requirements decomposition with AD application

This paper proposes to design a robust system for compiling income tax certifications, with the use of a decomposition approach of functional requirements, based on the verification of the axioms of independence and information [9]. The independence axiom ensures the consistency of the project. The axiom of information enables the selection of the most robust solution, in presence of more substantial decompositions. The robustness of a software project implementation results in the development of a software solution consisting of logically consistent modules. It must meet the client’s requirements, with a minimal information content value. However, in the field of software engineering, the use of the axiom of information is often overlooked due to a difficulty in quantifying the information systems’ content. The axiomatic design is limited to the verification of the decomposition consistency. Next, the choice for the most suitable solution remains limited to the designer’s experience and creativity. Nevertheless, this approach can be dangerous from an engineering point of view, because the deadlines of a system delivery to the client risk not being respected, or procedures with anomalies could be deployed. All these risks entail important repercussions on the production process of income certifications. This article will show, however, that the application of the axiom of

information in the field of software allows both to plan the system development phases and to respect the deadlines.
With the proposed approach, designers will be able to evaluate - among different functional consistent decompositions - the solution that requires the minor effort in terms of resources and development time. This approach is equivalent to choosing the functional decomposition with the lowest information content. For this reason, we propose to measure the information content of the system through the counting Function Point technique [12, 13]. In fact, the measure of the information level in terms of FP of the selected decomposition enables us to define the software implementation productivity. In addition, algorithms such as COCOMO enhance the estimation of deadlines and resources needed for the project, according to the function points of the selected decomposition [19]. Therefore, the concept of robustness in terms of AD is to select the designing solution which, on the same level of functional requirements, involves less human resources.

7.1. *Function point estimate of functional decomposition*

At each level of decomposition of the functional requirements (FR), an estimate can be made in terms of function points. If the level of detail of the decomposition is too abstract, estimation techniques can be used, such as the Early and Quick Point function. The estimate approximation always depends on the level of knowledge of the system operation. In general, there is a certain correspondence between functional requirements (FR), the design parameters (DP) and the constituent elements of the calculation in the function point. To illustrate the correspondence, we must consider that each functional decomposition can be represented with a module-junction structure diagram. For instance, let us consider the functional decomposition represented by the correlation matrix between FR_s and DP_s, as seen in Fig. 2. The Module-junction structure diagram (MJS) outlines this decomposition. The MJS diagram enables us to represent the FR-DP relations in a single block. Each module of the MJS diagram includes property details and functionalities associated with them [9, 10, 20]. Therefore, the matrix above can be represented in Fig.3.



	DP_1	DP_2	DP_3	DP_4
FR_1	A_{11}			
FR_2		A_{22}		
FR_3			A_{33}	
FR_4				A_{44}

Fig. 2. First level Matrix.

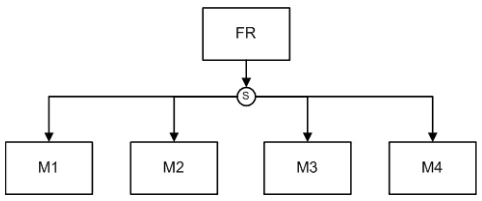


Fig. 3. Module-junction structure diagram.

In addition, each module (M) is an object that holds the data structure (DP) and the functions that refer to that structure (A_{ij}), as seen in Fig.4. Therefore, a generic module (M) is defined as the row of design matrix (Fig. 2), that yields the FR of the row when it is multiplied by the corresponding DP (i.e., data) [21]. Essentially, a module is an object which contains the data and the methods that allow to implement the functional requirement FR. In terms of Function Points (DP), the data structures correspond to the data files (EIF and ILF), while (A_{ij}) are the transactional functions (methods) that will be defined below.

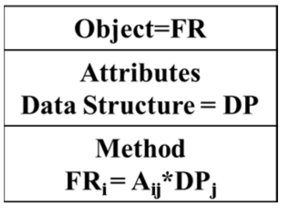


Fig. 4. Module M representation. Source [8].

These findings can provide a measure in terms of function points for each F_i functional requirement. The ILFs (Internal Logical Files) correspond to the data structures maintained (modified) by the system. The EIFs (External Interface Files) 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 Fig.5.

The complexity is determined according to the following parameters [12, 22, 23]:

- 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.

The matrixes, shown in Fig.6, in Fig.7 and in Fig.8, assess the complexity of transactional functions and of data files according to IFPUG 4.1 [12].

	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

Fig.5. Count FP matrix.

EI	1-4 DET	5-15 DET	> 15 DET
0-1 FTR	Low	Low	Average
2 FTR	Low	Average	Large
3 or more FTR	Average	Large	Large

Fig. 6. Evaluation matrix of EI complexity.

EI	1-4 DET	5-15 DET	16 o più DET
0-1 FTR	Low	Low	Average
2 FTR	Low	Average	Large
3 or more FTR	Average	Large	Large

Fig. 7. Evaluation matrix of EO / EQ complexity.

ILF/EIF	1-19 DET	20-50 DET	51 o più DET
1 RET	Low	Low	Average
2-5 RET	Low	Average	Large
6 or more RET	Average	Large	Large

Fig. 8. Evaluation matrix of ILF / EIF complexity.

7.2. Function point analysis

Estimating the information content of a system to be designed using function points (FP) or equivalent instruments such as the Early & Quick function Point (EFP) provides another relevant advantage for the functional decomposition mechanism proposed. As observed, each measure in function points has, as its main objective, the identification of elementary functions. Each estimate equals to decomposing the system into smaller parts, which are assigned a numerical value following the counting rules. In the case of Early & Quick Point function, the estimate is carried out on macro functions at a non-elementary conceptual level, as no detailed knowledge is available yet on how the system is functioning [13]. The measuring method through the function points, however, takes place with the identification of elementary functions and data structures - whose descriptions already provide software designers with all the necessary elements to start with the implementation phase. In this case, the axiomatic approach can select the most appropriate functional decomposition for our objectives, i.e. the robust decomposition. However, prior identification of the basic components of the system enhances a simplification of the functional decomposition mechanism. The action of decomposition becomes a linear process.

7.3. VAF (Value Adjustment Factor)

The measurement techniques of the information content proposed in this article do not take into account the adjustment factor. This is a corrective device of the function points size, compared to 14 General System Characteristics, which cannot be deducted from the system user's point of view. The 14 characteristics of the system that are taken into consideration are: Data communication, processing Distribution, Performance, extensive configuration use, frequency of transactions, interactive data entry, End-user efficiency, interactive updating, computing complexity, reusability, easy installation, easy operation management, multiplicity of sites and easy editing. Their influence is evaluated on the application of a scale of values: 0 not present / with no effect, 1 secondary influence, 2 moderate influence, 3 average influence, 4 significant influence, 5 strong generalized influence [12, 24, 25].
The adjustment factor can extend the potential use of the function point method, thus enabling us to consider elements that are not related to the end user’s point of view. For example, the function points metrics does not measure the level of complexity of the calculations performed by the system. The EO transactional functions are counted according to the DET (i.e. the number of fields they are referred to) and to the FTR (the accessed data structures). The complexity of the algorithm calculation is neglected. With the VAF, it is possible to adjust the counting, taking into account the level of complexity of the calculations performed by the procedure. This measure adjustment of the information content of a system can find its use in the presence of specific design constraints.

8. Formalization of functional decomposition method

The considerations made in the paragraphs above allow us to propose a decomposition mechanism of the functional requirements for the compilation of the income tax returns process, which takes into account both the axiom of independence and the axiom of information. In addition, the axiomatic approach also enables us to take into consideration, during the analysis phase, the adherence to the design constraints and warnings derived from the analysis of non-conformity clusters that are catalogued during the assistance to users phase during the past years. To summarize, the compilation of the income tax certification process may follow a functional decomposition approach which is divided into the following phases:

a) Identification of functional requirements FRs, design parameters DPs and related reports;
b) Identification of the design constraints;
c) Cautionary adjustment during the system design stage, based on non-conformity clusters NCs classified by the assistance to users;
d) Verification of the validity of the axiom of independence by the correlation matrix between FRs and DPs;

e) Verification of the axiom of information validity, based on the measurement of the running data system content using the Function Point technique or any similar estimation approaches such as the Early and Quick Function Points [11, 12].

9. Selecting a robust decomposition

Operations outlined above in a), b), c), d) and e) must be performed in an iterative manner. An application of the above five points at the same time enables us to locate the robust decomposition on the referring level. In addition, each decomposition can be expressed in an equivalent manner by using special diagrams (Module-junction structure diagram, Flowchart diagram, Class diagram for the design matrix, UML diagram). These diagrams help us get the technical documentation ready for each level of decomposition. This activity is aimed at planning the implementation activities and to provide guidance to programmers. This decomposition makes use of zigzagging process between two adjacent domains [9, 21]. The process ends when a sufficient level of detail is reached, in order for programmers to obtain all the information needed to start the implementation phase of the system. The entire process is summarized in Fig. 9.

10. Conclusions

The proposed top down decomposition approach of the functional requirements and robust configuration selection by levels of analysis allows to get multiple key-benefits, in particular it is possible to:

- Design software solutions deemed suitable the specific operating environment;
- Develop the minimum amount of software features for system purposes, avoiding unnecessary implementations;
- Have an accurate planning (resources, costs, scheduling);
- Implement any preventive resolution policies of entire non-conformity clusters, avoiding many recurring problems reported through the previous annual experiences, even under different forms;
- Reduce the system software development time;
- Enhance any routine or evolving maintenance of the system;
- Produce technical documentation of the project at all levels of the development process;
- Estimate the software size of the system in terms of Function Points if the level of decomposition of the project is well detailed, or, vice versa, in terms of Early Function Points;
- Measure to balance the size of the software project developed in terms of Function Points (Baseline);
- Update the baseline system rapidly, to account for any maintenance actions;
- Facilitate the reporting of technical information between designers and programmers.

This results can help in the development of robust process able to prevent the generation of non-compliances.

References

[1] Law December, 23, 2014, n. 190. Provisions for the formulation of the annual and multiannual budget of the State (2015 Stability Law). (14G00203) (GU General Series n.300 of 12-29-2014 - Ordinary supplement n. 99).

[2] Monti C., Giorgetti A., Girgenti A. An Axiomatic Design Approach for a Motorcycle Steering Damper. Procedia CIRP 2015;34:150-55.

[3] Vezzù S, Cavallini C, Rech S, Vedelago E, Giorgetti A. Development of high strength, high thermal conductivity cold sprayed coatings to improve thermal management in hybrid motorcycles. SAE International Journal of Materials and Manufacturing 2015; 8(1):180-6.

[4] Giorgetti A, Baldanzini N, Biasiotto M, Citti P. Design and testing of a MRF rotational damper for vehicle applications. Smart Materials and Structures 2010;19 (6):art. no. 065006.

[5] Cavallini C, Giorgetti A, Citti P, Nicolaie F. Integral aided method for material selection based on quality function deployment and comprehensive VIKOR algorithm. Materials and Design 2013; 47: 27-34.

[6] Giorgetti A, Girgenti A, Citti P, Delogu M. A Novel Approach for Axiomatic-Based Design for the Environment. Axiomatic Design in Large Systems, Ed. Springer 2016: Chapter 5 131-148 2016.

[7] Arcidiacono G, Giorgetti A, Pugliese M. Axiomatic Design to improve PRM airport assistance.Procedia CIRP 2015;34: 106-11.

[8] Girgenti A, Giorgetti A, Citti P, Romanelli M. Development of a Custom Software for Processing the Stress Corrosion Experimental Data through Axiomatic Design. Procedia CIRP 2015; 34: 250-5.

[9] Suh NP. Axiomatic Design - Advances and Applications. Ed. Oxford University Press, New York. Chapter 5; 2001.

[10] Clapis PJ, Hinterstiener JD. Enhancing object-oriented software development through axiomatic design. Proceedings of ICAD 2000, Oxford, Massachusetts.

[11] Albrecht AJ. Measuring Application Development Productivity. Proceedings SHARE/GUIDE IBM Applications Development Symposium, October 1979.

[12] IFPUG - Function Point Counting Practices Manual, Release 4.1 - Westerville - Ohio, 1999.

[13] Meli R, Santillo L. Function point estimation methods: a comparative overview., IFPUG Conference - September 15-19, 1997 - Scottsdale, Arizona USA.

[14] Togay C, Dogru AH, Tanik JU. Systematic component-oriented development with axiomatic design. Journal of Systems and Software 2008; 81(11):1803-15.

[15] Instructions for completion CU 2016. Italian Revenue Agency, 04/02/2016.

[16] Cavallini C, Giorgetti A, Citti P, Meneghin A. Sviluppo di un approccio olistico per l’analisi e la risoluzione delle non conformità. Sei Sigma & Qualità, 2012, 3 (1).

[17] Giorgetti A, Cavallini C, Meneghin A, Arcidiacono G. Development of a holistic model for the proactive reduction of non-conformities, International Journal of Production Research, Forthcoming.

[18] Rolli F, Giorgetti S, Citti P. Integration of Holistic Non-Conformities Management and Axiomatic Design: a case study in Italian Income Tax Returns Management. Procedia CIRP 2015; 34: 256-62.

[19] Rollo AL Functional Size measurement and COCOMO – A synergistic Approach. Proc. of Software Measurement European Forum 2006: 259–67.

[20] EL-Haik BS, Shaout A. Software Design for Six Sigma: A Roadmap for Excellence. Ed. John Wiley & Son, Inc., 2010.

[21] Do SH, Suh NP. Object-oriented software design with axiomatic design. Proceedings of ICAD2000, Cambridge, MA, 278-84.

[22] Desharnais JM, ST.Pierre D, Maya M, Abran A. Full Function Points: Counting Practices Manual - Procedures and Counting Rules, SERML/SELAM, November 1997.

[23] Abran A, Robillard PN. Function Points Analysis: An Empirical Study of Its Measurement Processes. IEEE Transactions on Software Engineering 1996;22 (12).

[24] Lokan CJ. An Empirical Analysis of Function Point Adjustment Factors. Information and Software Technology 2000;42 (9): 649–59.

[25] Srivastava JS, Singh G. Optimized GSCs in Function Point Analysis - A Modified Approach. International Journal of Research and Reviews in Applied Sciences 2013;17.

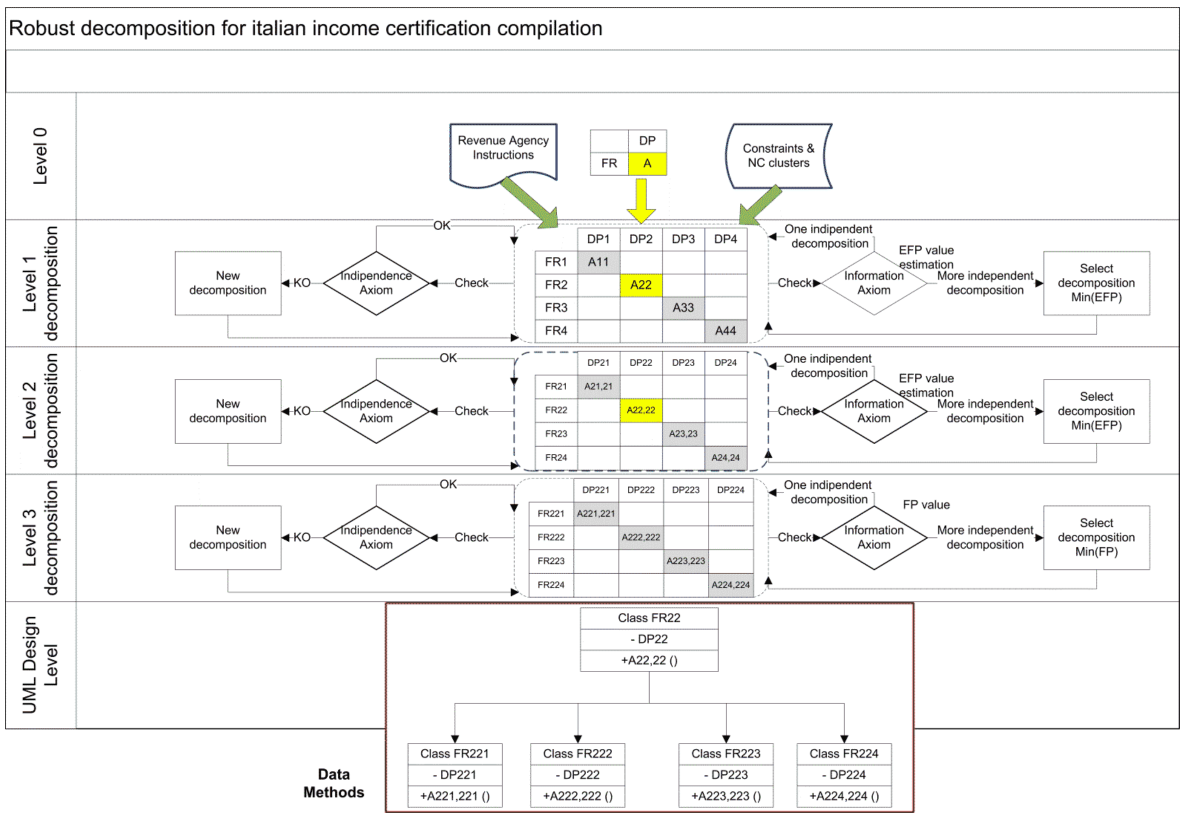


Fig. 9. Robust decomposition for software design.