

World Conference: TRIZ FUTURE, TF 2011-2014

A novel approach to Architectural Problem Space Framing using TRIZ- based Contradiction Approach

Amirabbas Najari^a, Marc Barth^b, Michel Sonntag^a

^aUniversity of Strasbourg- National Institute of Applied Sciences of Strasbourg – Design Engineering Laboratory (LGéCo)- France

^bStrasbourg Business School - HUMANIS (Humans and Management in Society)- France

Abstract

In the absence of a specific form for representation, architectural contradictions are often ignored or treated by intuition, trial-and-error, or compromise. But, the set of methods and tools of Theory of Inventive Problem Solving (TRIZ) would formulate, represent and solve these contradictions. This paper presents a part of an empirical study aimed to explore the possibility of adapting the “Contradiction” concept of TRIZ as a conceptual strategy for the early stages of an architectural design process. To obtain this objective, the architectural program of Project of Extension of INSA de Strasbourg was built in the form of Problems Graph of IDM-TRIZ (Inventive Design Method). This modeling aims to study the possibility of using the information provided by an architectural program to state architectural problems as Contradiction, and its contribution to the understanding of the problem space. Some limitations of the IDM-TRIZ for architectural applications are reported as well, and certain possible developments for Analysis of Initial Situation phase to fit better to the architectural programming process are suggested.

© 2015 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the Scientific Committee of TFC 2011, TFC 2012, TFC 2013 and TFC 2014 – GIC

Keywords: *Contradiction, Architectural Programming, Problem Framing, OTSM Network, IDM-TRIZ.*

1. Introduction

“Architecture is necessarily complex and contradictory” as R. Venturi mentioned [1]. In every aspect of an architectural project, from programming to construction, form technical equipment choice to environmental considerations, even in its aesthetic expression, the architect should deal with several contradictory elements. By analyzing several architectural masterpieces from different styles and contexts, Venturi showed how the architects

* Amirabbas.najari@insa-strasbourg.fr, marc.barth@insa-strasbourg.fr, Michel.Sonntag@insa-strasbourg.fr

apply the “phenomenon of Both-And” and the “Double-Functioning elements” to design wonderful architectural solutions for contradictory purposes. Despite such awareness in architectural practice and theory, the concept of contradiction has not been explored as a conceptual strategy.

It is widely accepted that architectural design as a specific design activity lies between art and engineering [2]. However, Goel and Pirolli [3] showed that the cognitive strategies used by designers to structure the design problem space are common to all domains of design including architecture. Thereby, exploring the possibilities of the methods currently used in other domains of design for adapting to architectural design is a promising avenue toward a multidisciplinary methodological research in architectural design [4]. In this perspective, it seems the recent developments of TRIZ constitute a pertinent alternative by offering a set of tools to build the Problem space and to systematize the use of knowledge for representing the project contradictions.

Research on the possible contributions of TRIZ in architecture has recently gained momentum [5]; [6]; [7]; [8]. These studies mainly focus on the application of the classical TRIZ and its tools, particularly the Matrix of Inventive Principles. They used the theoretical corpus of TRIZ to find new solutions for a given problem. The interesting results of these studies demonstrate the applicability of TRIZ in technical dimensions of architectural projects. Such use of TRIZ fits to the purpose of classical TRIZ when it starts with a contradiction that has already been identified. However, two main questions remain for the one who pretends to apply the TRIZ in an architectural design dealing with both technical and non technical aspects of a project: *a)* Even an ordinary architectural project involves several problems: how can one select the problem among the countless multi-criteria problems of the project to be solved by TRIZ? *b)* By considering the contradiction as a common cause of any inventive problem, Altshuller [9] proposes to reformulate such problem as a contradiction. But, is it possible to represent any architectural problem in form of contradiction?

These two questions formed a multidisciplinary research conducted in the Laboratory of Engineering Design (LGéCo) of INSA of Strasbourg. This research has two objectives: First, it tries to evaluate the possibilities and limitations of using the concept of “contradiction” as a fundamental postulate of TRIZ in architectural design which deals with a non-technical domain. Second, this research tries to examine the contribution of application of dialogical-systemic TRIZ-based approaches in the early stages of architectural design, more precisely in architectural programming. In order to ensure this, the “Analysis of Initial Situation” phase of the “Innovative Design Method” (IDM-TRIZ) is applied to the architectural program of the real project of new Extension of INSA of Strasbourg. The STEPs software is used as a supporting tool of IDM-TRIZ.

This paper reports the contributions of this research in two-folds. First, with respect to the architectural programming, we argue that the Problem Graph contributes to state the architectural problems by structuring a problem space which encompasses not only the formal and functional issues (usually represented by architectural sketches and diagrams), but also the issues related to time and economy. We show also how the nontechnical objectives, constraints and propositions presented in an architectural program can be interpreted and used to build the Problem Graph. Second, with regards to the IDM-TRIZ methodology, we underline the limitations we have faced in this architectural application. Also, we propose some possible developments for the Analysis of Initial Situation phase which are more suitable for an architectural programming process. The rest of the paper is organized as follows: Section 2 presents Architectural programming, Section 3 provides Problem statement in TRIZ approaches, Section 4 reports the Case study, Section 5 presents the Discussion about the contributions of this research and finally Section 6 concludes the paper.

2. Architectural programming

When the architecture deals with “transforming the space” [10] from an existing situation to a desired one, an architectural program describes both of them. It describes an existing situation of a project and provides some features of its future. This way, the architectural programming can be compared with the phase of Analysis of Initial Situation in engineering design. While acknowledged more and more as a crucial phase in the architectural design process affecting all succeeding phases, architectural programming is poorly exploited by architects [11]. As well, while innovation is considered as the key characteristic of recent evolution of industrial society [12], there are

limited works reporting on how an architectural program could be structured to encourage innovation. However, its potential agency in architectural design process is being increasingly taken into consideration [13].

Earlier studies about the architectural program have been basically seeking the prescriptive methods and instructions for a useful programming process, ranging from searching and analyzing the relevant information to writing out a summary of project requirements. The Information Matrix of Pena and his procedural method for “Problem Seeking” [14], the Question matrix of Durek and his Issue-based programming process [11], and the Kumlin’s checklist of Design Issues [15] all aim to provide a comprehensive framework for programming. Few comparisons have been made on the suitability of these programming methods [13]. However, several critiques pointed out the insufficiency of current programming methods in different levels and aspects [16]; [17]; [18]. To improve the architectural programming process, the recent research has been also inspired by other domains such as decision-making [19]; project management [20], knowledge management [17]; [11]; Concurrent Engineering Framework [21]; and particularly, much attention has been drawn by Value Management Techniques [22]; [23] [20]. Notwithstanding this considerable expansion in architectural programming research over the past ten years, the lack of tangible improvements remains elusive [18]. In this research, we focus on two needs that the studies on architectural programming have rarely addressed to: “Problem Statement” and “Setting up the problem space” in the architectural programming process.

2.1. Architectural program as Problem statement

Ideally an architectural program presents the mission, goals, values, selected and prioritized relevant needs and requirements, problem scope, solution scope, and resource constraints (delay and cost) of the project. In order to be comprehensive, the program should process these issues in different aspects such as functional, operational, aesthetic, social, cultural, environmental, technological, economic, etc. Furthermore, the program should identify the impediments that can prevent the transformation of an initial situation to a desired one. Pena and Parshall [14] assert that programming is a heuristic process which should lead to clearly state the problems of project. This set of problems will be acquired by an architect in the early stages of design process. Nonetheless, even Pena and Parshall [14] who considered architectural programming as a “Problem Seeking” process does not propose a particular syntax to formalize a problem. This leads to an unclear, latent or discursive representation of problems. Similarly, there is no formalism or model to represent a contradiction in the literature of architectural design. Consequently, in the absence of a specific form for representation, the architectural contradictions are ignored or treated by intuition or trial-and- error.

2.2. Architectural programming as setting up the problem space

Any architectural project contains several problems. The current programming methods don’t explain how to prioritize and choose the problems to solve; this task is conceded to architect’s intuition, his experience and preferences. As Paul Rudolph mentioned “all problems can never be solved... architects are highly selective in determining which problems they want to solve” (quoted by [1]). This way, the act of preferring one problem to another is considered as an asset leading to a marvelous result. And, “the ability to identify the most influential problem and its constructive elements” [24] is known as a significant factor of design expertise. In other words, in order to choose the problem to solve among several problems, the architect traditionally relies on his/her proper experience/expertise. Such individual approach relying on the creative ability of the architect/author is criticized [4]. The critics argue that the architectural design is a collective process in which different agents contribute to identify and define several problems of project. So, not only an individual’s cognitive capacity is insufficient to process such quantity of information, but also a process which is built on such elitism is not democratic. As a reason, the critics underline the lack of a means to set up the Problem space in a participative manner.

Structuring the Problem space of the design project constitutes also as one of the main challenges in TRIZ development. We focus on the Problem Graph of IDM-TRIZ that was conceived to respond to the same shortages. The problem addressed in the current study is to know if this Graph can also be used in architectural programming. Since this method holds a specific definition of problem, provides a particular model to represent a problem, and

presents a formal structure to build and maintain the problems space. More precisely, this study is aimed to see if the problems that are conspicuously or latently presented in an architectural program can be formulated and represented through a Problem Graph of IDM-TRIZ.

This Graph, as we will show later, has allowed us to state the problems of Project of Extension of INSA of Strasbourg, illustrate the dialectical interconnections between these problems and propositions, and consequently identify the most influential problem. Although the IDM-TRIZ proposes an axiomatic approach to structuring the Problem space and also a formal definition of the contradiction related to problems [25], but this method aims principally the industrial domain. Our empirical study shows the need for some improvements in the IDM-TRIZ to better meet the architectural design specificities. These improvements are suggested in the section of Discussion of the present paper.

Before reporting the case study, it is worth mentioning briefly the recent development in TRIZ; this is to clarify the problem statement in a complex project.

3. Problem Statement in TRIZ approaches

The necessity of methods and tools for structuring the Problems space, which aims at highlighting the Core problem and identifying the elements of contradictions causing this problem, stems from the fact that the classical TRIZ is weak in managing complex problems when the conflicts are hidden, the contradictions are multiple, and the formulation of the proper technical contradiction is difficult [25]. Altshuller has also highlighted the “systematic clarification of the problem statement” as an important direction for future development of TRIZ [26, p. 278]. This need resulted in a set of methods and tools for both analysis of initial situation and definition of the problem. The Step 0 of ARIZ-85C, Innovation Situation Questionnaire (ISQ) [5] and the Network of Problems (NoP) of OTSM-TRIZ [27] are the examples of methods developed for this purpose.

The IDM-TRIZ adapted and developed the concept of network proposed by OTSM-TRIZ to provide “a complete framework for knowledge acquisition, representation and manipulation” [28] for phase of Analysis of Initial Situation. By such adaptation, the IDM-TRIZ aims to follow the OTSM-TRIZ promise, *e.g.* to extend to multidisciplinary problematic. The Analysis of Initial Situation of IDM-TRIZ provides also a possibility to extract the inherent contradictions of a problem from its statement. A detailed methodological description of Initial Situation Analysis can be found in [28].

A reminder of IDM-TRIZ ontology let us accurately position our contribution in terms of transforming the objectives and constraints of an architectural program to a Problem Graph. Problem, Partial Solution, and Parameters constitute three basic concepts of IDM ontology. A parameter is a feature of problem subject which is involved to make it unsatisfactory (but, the parameter identification in our project is out of the scope of this paper). Concerning the problem, according to TRIZ, it can be defined as a situation where an obstacle prevents progress, an advance, or the achievement of what has to be done. The IDM-TRIZ ontology requires the problem to be reduced to a single idea and its expression reach to a maximum decomposition in a specific level of genericity. So, a problem (PB) has the following syntax:

<Subject>+<verb>+<complement>

As for the Partial Solution (PS), it states a known and verified result, progress or an improvement in problem solving. The syntax for a Partial Solution complying with IDM ontology is:

<Verb in its infinitive form > + <complement>

4. Case Study

In our study, the Problem Graph of IDM-TRIZ is applied to the architectural program of the new Extension of INSA of Strasbourg in order to transform the initial fuzzy situation of the project into a specific Problem space in which the problems are clearly stated, the Core problem is identified and its Poly- contradictions are formulated. This application allows examining the possibility of representation of the architectural problems that were identified in the phase of programming as contradiction. To acquire the needed information, three sources were used: the architectural program of Extension project of INSA of Strasbourg, our notes made during meetings with the programmer and INSA representatives, and finally the Problem Seeking method [14]. In the following, the way in

which we have transformed the information provided by the architectural program of INSA project into the Problem Graph is explained.

4.1. INSA's Extension Project

To meet its development policy, the INSA of Strasbourg has defined an extension project. This new building is to be located in the occupied site of INSA, and so to have access to other existing buildings of INSA. Such mission makes the project more complex. A professional architectural programmer was thus hired to clarify the complexity of the project.

4.2. The Architectural Program of INSA Project

The architectural program, done by the programmer, matches with the typical facility program template. It sets out the missions and goals of the project, existing situation analysis (site analysis, technical diagnostic etc.), functional needs, needed spaces, and a cost /time evaluation. It also proposes the spatial locations for activities and an operation schedule. Other information presented in the INSA program, like details of space types, surface etc., belonging to a more detailed level, were not integrated in the Problem Graph.

In this program, some problems are verbally expressed, often scattered among other subjects. They are mentioned conspicuously or latently in a complex way and with complex sentences. Neither a specific formalism nor a precise syntax is used to formulate the problems in this document. There is no link between problems and/or propositions, so the mutual influences are unknown. Furthermore, the program is set in a way that the issues don't contradict each other. Consequently, no contradiction is highlighted.

4.3. Construction of Graph

4.3.1. Extract the problems and solutions from INSA program

The program of INSA defines the features of the initial situation of INSA and the future extension by presenting the objective, constraint, and proposition; it also suggests some operations to be taken to transform the existing features into the future ones. Regarding our objective research, we are interested in the problems arising in this transformation. In other words, these issues are used to extract the problems and proposition/solutions. With regards to the problem extraction, we interpreted the objectives and constraints and state a problem when one of the following cases is observed:

- When a feature of the existing situation prevents achieving a/some objective(s);
- When a feature of the future state cannot be achieved by project resources (technical/ physical etc.);
- When the limit of budget prevents achieving a/some objective(s);
- When the target date of the project schedule prevents performing an operation.

The following examples show how the IDM-TRIZ syntax is used to define one or several problems behind an objective. The sentence of the document of Program of INSA is «the project had originated from the increasing number of students. » This phrase produced:

- PB03: INSA lacks of spaces. And
- PS02: Build a new building.

The constraints presented by the INSA program provide also the problems, as the following example shows. The sentence of the document of Program of INSA is « INSA should work during construction. » This phrase produced:

- PB07: «Flow inside the existing building is a constraint for finding a location for the new building »

The two following examples show how we formulated the problems expressed by the INSA program in an indirect way. The sentence of « an opening in Southeast side to the University Campus is sought » has allowed producing:

- PB03: Site is closed to the University Campus.

The sentence of « the project aims to improve the functionality of spaces » is translated to

- PB01: Functional distribution of spaces is not satisfactory.

Furthermore, the programmer and the INSA representative had mentioned some problems and proposition/solutions during their meetings, but these issues have not been stated in the document of the project program. We added also these issues to the Graph. They were originally reformulated in the required syntax. Like the flowing PBs and PSs.

- PS22: Build several entries. Which is linked to:
- PB53: Cost of security control for several entries is high.

In order to extract the PSs, the solution/propositions of program are considered. These propositions are produced by programmer through using some known programming concepts. A concept indicates a way either to transform an existing feature to its future state or to create a new feature. For example, the Grouping is a programmatic concept used by the programmer to propose the PS41: Group the activities of Department of Architecture in the new building. Totally, 73 PBs and 74 PSs are defined and put in the Graph. (Fig. 1)

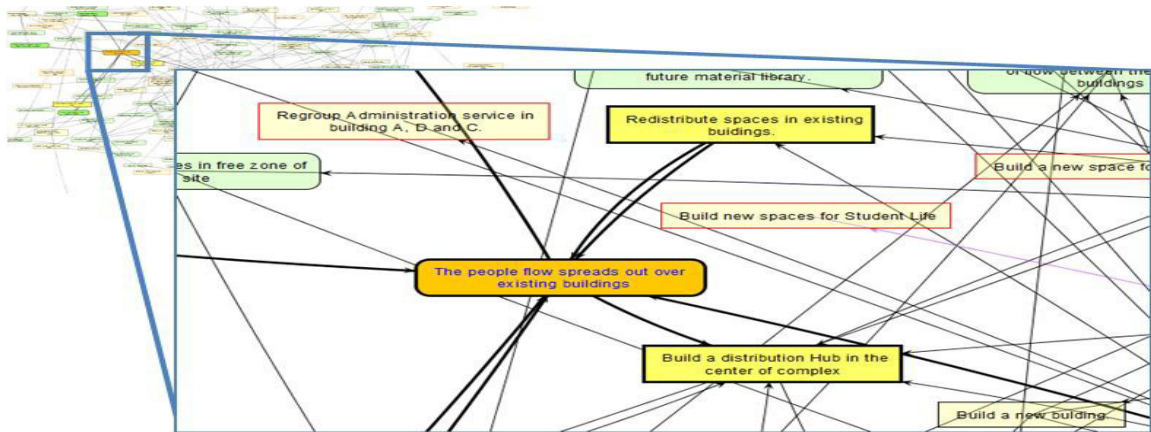


Fig. 1. The Problem Graph of architectural Program of the INSA project

Evidently, the Graph doesn't reach the saturation and hence may enhance it by placing new PBs and PSs, if one continues to ask more information. But, we aimed to use only the information presented in the program documents of INSA project.

4.3.2. Link between problems

During placing the PBs and PSs, some links were identified and created in Problem Graph. To ensure that maximum identifiable link is created, we used the Matrix of Information of Pena and Parshall [14] in which the objectives, constraints and proposition are respectively presented as goal, fact and concept, and are structured in a matrix structure.

4.3.3. Core Problem

The Problem Graph can highlight the influence of each problem on the entire Problem space. It also allows increasing the possibility of identifying a particular problem having a large influence on other PBs/PSs, called Core Problem (CP). Identifying the CP is the main objective of the Graph. The CP causes more reduction (shrinking) in the Graph; that is to say, if this problem is solved, the greatest number of problems will be eliminated.

The Problem Graph of the case study shows the PB06, (Increasing the number of staff and student need more space), as the most influential problem with an influence rate of 13 (a chain of 12 PBs is derived from this problem). But this PB, placing at a high level of genericity, doesn't provide a meaningful direction. After this problem, the PB07, (The people flow spreads out over the existing buildings), having a rate influence of 4, is chosen as CP. While the linked PS to this problem can give a significant direction to solve the CP; it is the "PS73: Build a distribution Hub in the center of the INSA complex".

4.4. From Problem Graph to contradictions

To formulate the contradictions of the CP, we applied the formal definition of contradiction proposed by [25]. This process is integrated in the IDM-TRIZ. We briefly identified the Parameters by considering the features of a PB that involve in making it subject unsatisfactory. Totally, 20 Parameters are identified and linked with PBs and PSs in Problem Graph of the case study. By declining the PBs to Action Parameters and PSs to Evaluation Parameters, the following contradiction is identified as the main cause of CP:

The Spatial location of the Distribution Hub is to be both central to satisfy the Shape-Geometry of space and non- central to satisfy the Existing building location.

In this contradiction, the Action Parameter is the spatial location of the Distribution Hub; which can be placed at the Centre or at the Corner (opposite values) of the INSA complex, and Shape-Geometry, Surface, Volume and Entries of the existing buildings constitute the Evaluation Parameters. Figure 2 shows a part of the Poly-contradiction of our case study.



Fig. 2. The Poly-contradiction of CP of the Program of INSA project

5. Discussion

The contributions of this research can be reported in two-fold, reported as follows:

First, regarding the architectural Programming, we have shown the possibility of using the concept of contradiction of TRIZ in architectural domain. We stated the objectives, constraints and proposition of an architectural problem in form of the Problem Graph. We have shown the possibility of extracting the constructive elements of the Problem Graph (PB, PS, Parameters) from a typical architectural program having a procedural model. This way, the architectural design can take advantage of recent theoretical development in engineering inventive design. Indeed, the Graph theory has been already used in architectural design [29]; but the originality of the Problem Graph of IDM-TRIZ comes from the fact that it combines two approaches of systemic and dialectic. As highlighted by [30], these two approaches constitute the principles of complex thought to act on the complexity of architecture. Furthermore, considering the programming as "a process of argumentation" [31, p. 279], the Problem Graph can help to circumvent the inability of current computer systems in the formalization of this type of reasoning in architectural programming. [32]

However, we noted that using a method that has been developed for a different domain than architecture, has an important influence on our vision of project. In fact, the linear and narrative structure of a usual architectural

program allows the architect to establish an approach to qualitative aspects of architecture. While the plain meaning of problem statement and the holistic vision offered by the Problem Graph restrict such approach.

Second, regarding the contribution of this architectural application of IDM-TRIZ, we can point out the following considerations.

- An architectural program presents different pictograms such as flowchart, diagram, plan *etc.* It is difficult to verbally represent this information in terms of PB and PS. The capacity of the Problem Graph is limited to represent the information concerning the spatial aspects representing graphically by architects
- The structure of the Problem Graph doesn't allow integrating the dynamism of the problem Space. In fact, in architectural programming, there are some propositions and solutions that extend over the time. As "mirror" initiative, in which a first building, initially unoccupied, is renovated then delivered to the functions of the second building, which is then in turn also renovated.
- When there are many problems, it is not easy to verify their possible interactions. An interface can facilitate setting up the links by inquiring systematically the links of each problem.
- The Problem space can be changed in terms of selected scenario to solve the problem. The IDM-TRIZ allows weighting the problems, but, it is useful to be able redefine the weight of problems for each scenario.

6. Conclusion and perspective

One of the main objectives of an architectural program is analyzing the problem space of the architectural project. This analysis will lead to identify the problem that presents itself as the most influential one. This empirical study shows that the method proposed by IDM-TRIZ is applicable to such an endeavor. It is possible to use the information provided by an architectural program to build a Problem Graph. The objectives and constraints presented by the architectural program are translatable in terms of PB and PS. Furthermore, these problems can be formulated as contradictions. In other words, the process is stated by identifying the Core Problem of our case study and ended by identifying the main contradiction among all of contradictions that produces this PB.

However, the next step of our research is to examine the relevance of TRIZ tools to solve this CP.

References

- [1] Venturi R., Complexity and Contradiction in Architecture. The Museum of Modern Art, 1977.
- [2] Lawson B., Design in mind. Butterworth Architecture, 1994.
- [3] Goel V. and P. Pirolli, « The structure of design problem spaces », Cogn. Sci., vol. 16, no 3, p. 395-429, juill. 1992.
- [4] Terrin J.-J., C collaborative pour r innover en a rchitectu re p rocessus, méthode, outills. Paris: l'Harmattan, 2009.
- [5] Kiatake M. and J. R. D. Petreche, « A case study on the application of the theory of inventive problem solving in architecture », Archit. Eng. Des. Manag., vol. 8, no 2, p. 90-102, 2012.
- [6] Najari A., M. Barth and M. Sonntag, 2014. Contribution de la notion de Contradiction à la Stratégie Conceptuelle pour la phase amont de la Conception Architecturale. Dans S. Kubicki et al. (ed.) Interaction(s) des maquettes numériques: PUN - Editions Universitaires de Lorraine, Nancy. pp. 113-123. ISBN : 978-8143-0171-9
- [7] Craig S., D. Harrison, A. Cripps, and D. Knott, « BioTRIZ Suggests Radiative Cooling of Buildings Can Be Done Passively by Changing the Structure of Roof Insulation to Let Longwave Infrared Pass », J. Bionic Eng., vol. 5, no 1, p. 55-66, mars 2008.
- [8] Lee Y.-C. and Y.-S. Deng, « A Design System Integrating TRIZ Method and Case-Based Reasoning Approach », Van Leeuwen JP HJP Timmermans Eds 2006 Prog. Des. Decis. Support Syst. Archit. Urban Plan. Eindh. Eindh. Univ. Technol. ISBN-10 90- 386-1756-9 ISBN-13 978-90-386-1756-5 P 387-402, 2006.
- [9] Altshuller G., Creativity As an Exact Science. CRC Press, 1984.
- [10] A. Siza, El Croquis 168/169 - Alvaro Siza. Madrid: El Croquis, 2013.
- [11] Duerk D. P., Architectural Programming: Information Management for Design, 1 edition. Wiley, 2008.
- [12] D. Cavallucci, « A research agenda for computing developments associated with innovation pipelines », Comput. Ind., vol. 62, no 4, p. 377-383, mai 2011.
- [13] Ryd N., « The design brief as carrier of client information during the construction process », Des. Stud., vol. 25, no 3, p. 231-249, mai 2004.
- [14] Peña W. M. and S. A. Parshall, Problem Seeking: An Architectural Programming Primer. John Wiley & Sons, 2001.
- [15] Kumlín R., Architectural Programming: Creative Techniques for Design Professionals. McGraw Hill Professional, 1995.

- [16] Smith J. M., R. Kenley, and R. Wyatt, « Evaluating the client briefing problem: an exploratory study », *Eng. Constr. Archit. Manag.*, vol. 5, no 4, p. 387-398, déc. 1998.
- [17] Emmitt S., *Design management for architects*. Blackwell Pub., 2007.
- [18] Blyth A. and J. Worthington, *Managing the Brief for Better Design*. Taylor & Francis, 2010.
- [19] Hershberger R., *Architectural Programming and Predesign Manager*, 1st éd. McGraw-Hill Professional, 1999.
- [20] Luo X., G. Q. Shen, S. Fan, and X. Xue, « A group decision support system for implementing value management methodology in construction briefing », *Int. J. Proj. Manag.*, vol. 29, no 8, p. 1003-1017, déc. 2011.
- [21] Kamara J. M., C. J. Anumba, and N. F. O. Evbuomwan, *Capturing Client Requirements in Construction Projects*. Thomas Telford, 2002.
- [22] Yu A. T. W., Q. Shen, J. Kelly, et K. Hunter, « An empirical study of the variables affecting construction project briefing/architectural programming », *Int. J. Proj. Manag.*, vol. 25, no 2, p. 198-212, févr. 2007.
- [23] Kelly J., R. Morledge, and S. J. Wilkinson, *Best Value in Construction*. John Wiley & Sons, 2009.
- [24] Dorst K., « Design Problems and Design Paradoxes », *Des. Issues*, vol. 22, no 3, p. 4-17, juill. 2006.
- [25] Rousselot F., C. Zanni-Merk, and D. Cavallucci, « Towards a formal definition of contradiction in inventive design », *Comput. Ind.*, vol. 63, no 3, p. 231-242, avr. 2012.
- [26] Khomenko N. and S. Yoon, « OTSM express analysis of an initial situation », in *ETRIA TRIZ Future Conference 2011 proceedings*, Dublin, Irlande, 2011, p. 277-288.
- [27] Khomenko N. and R. D. Guio, « OTSM Network of Problems for representing and analysing problem situations with computer support », in *Trends in Computer Aided Innovation*, N. León- Rovira, Éd. Springer US, 2007, p. 77-88.
- [28] Cavallucci D., F. Rousselot, and C. Zanni, « Initial situation analysis through problem graph », *CIRP J. Manuf. Sci. Technol.*, vol. 2, no 4, p. 310-317, 2010.
- [29] Loué C., René Alexander, *De la Synthèse de la forme: essai*. Paris: Dunod, 1971.
- [30] Farel A., *Architecture et complexité: le troisième labyrinthe*. Marseille: Ed. Parenthèses, 2008.
- [31] Rittel H. and D. Noble, « Issue-based information systems for design », *Univ. Calif. Berkeley Work. Pap.*, vol. 492, 1989.
- [32] Akin Ö., R. Sen, M. Donia, and Y. Zhang, « SEED-Pro: Computer- Assisted Architectural Programming in SEED », *J. Archit. Eng.*, vol. 1, no 4, p. 153-161, 1995.