A Function-Oriented Ontology Tool for Solving Inventive Problems

David Estrada-Contreras, Guillermo Cortés-Robles*, Giner Alor-Hernández and Ulises Juárez-Martínez
Postgraduate Department, Instituto Tecnológico de Orizaba, Av. Oriente 9 No. 852, 94300, Orizaba, Veracruz, Mexico.
Emails: daescon@gmail.com; {gcortes, galor, ujuarez}@itorizaba.edu.mx

Alejandro Rodríguez-González
ETS de Ingenieros Informáticos
Universidad Politécnica de Madrid
Campus de Montegancedo
Boadilla del Monte, 28660, Madrid, Spain
Email: alejandro.rodriguezg@upm.es

Every technical system provides at least one useful function. A problem-solving tool called Substance-Field Analysis – which is a TRIZ tool – uses the concepts of function and interaction to identify and describe the evolutionary changes of a system, in order to transfer successful system transformations among different technical domains. This is, in fact, a knowledge-based tool for problem-solving. Paradoxically, no knowledge management approach seems to be linked to this tool. Consequently, this work proposes a function-oriented ontology tool, which combines the advantages of ontologies with the Substance-Field Analysis approach in order to guide creative effort while solving inventive problems also called innovation problems.

Keywords: *Substance-field analysis, ontologies, theory of inventive problem solving, collaborative problem solving*

ACM Classifications: *I.2.8 Problem Solving, Control Methods, and Search; I.2.4 Knowledge Representation Formalisms and Methods; I.2.6 Learning-Analogies*

Introduction

The capacity to solve technical problems has evolved with the emergence of the world wide web. Within a challenging environment, people communicate and collaborate inside a networked world and thus, new information is generated and distributed at a rate never seen before. In this context, there is an increasing need for problem solving tools that should be capable of managing complexity, easy to be deployed in different domains and problems, but especially, they should change dynamically within a given domain (Hippe, 1996; Orloff, 2012). Despite the fact that several tools and approaches have been proposed to solve problems, there is a particular kind of problem

* Corresponding author

Manuscript received: 11 December 2012
Communicating Editor: Rafael Valencia-Garcia

that still represents a challenge: inventive problems. First, these are problems where available knowledge cannot produce an effective or satisfactory solution. Frequently, inventive problems lead to an impasse – a system demands two opposite states or mutually exclusive conditions – (Altshuller, 1999; Sushkov *et al*, 1995; Cavallucci and Eltzer, 2011). The solving process under these conditions is a complex task where creativity, knowledge gathered from different domains, and new tools are combined to produce valuable solutions. A particularly useful approach within this context is the Theory of Inventive Problem Solving, also named TRIZ theory. Indeed, TRIZ offers a set of tools useful for modeling and analyzing problems. When the nature and requirements of an inventive problem are clear, it is possible to identify a set of tools to solve it (Rousselot *et al*, 2012; Liua *et al*, 2011). One of the most effective TRIZ tools is a function-oriented tool called Substance-Field Analysis (SFA), which states that a minimal and controllable technical system that is capable of performing a function must be composed of three elements: two substances and one field (Rantanen and Domb, 2007; Fey and Rivin, 2005).

This modeling approach is based on an extent patent analysis that produces a synthesis of the most successful strategies to transform technical systems, and a method to graphically describe the interactions that take place among several subsystems or components. SFA is thus a knowledge-based tool for problem solving. Nevertheless, although knowledge is at its core, the tool does not have an explicit approach to capitalize knowledge and reuse it in a similar situation, and this produces a loss of experiences that could be very useful in other domains or to another user. The underlying hypothesis of this article could be synthesized as follows: if two different problems could be modeled with the same action affecting a function, then the solving strategy deployed in the first problem could be transferred to the second problem (Altshuller, 1984; Rahmani and Thomson, 2012). Thus, in this case, the need to capitalize knowledge becomes essential in order to increase performance when solving inventive problem, and to manage this intangible asset, the use of ontologies has proved to be an efficient strategy. Ontologies can, in fact, create the conditions for indexing and reusing knowledge, and these capabilities increase performance when solving inventive problems (Prickett and Aparicio, 2012; Zanni-Merk *et al*, 2011). Consequently, this article describes an ontology-based tool for problem solving, which uses functions in order to solve problems and capitalize knowledge.

The article has six sections. The first section describes the state of the art that reveals the feasibility of combining ontologies with the SFA. The second section describes the main approaches involved in this article with the goal to underline their complementarity: the TRIZ theory, the substance-field analysis (SFA) and ontologies. In the third section are identified the main requirements to cover the ontology. The fourth section briefly describes the integration of technologies to propose a collaborative web-based system. The fifth section describes how this tool was implemented and validated through a case study. Finally, conclusions and future work are emphasized in the sixth section.

1. State of the Art

Ontologies have been intensively applied in problem solving in several domains: software (Wongthongtham *et al*, 2006; Jung and Kim, 2011) new product development (Pandit and Zhu, 2007), as well in the medical field (de Clercq *et al*, 2001;Yu, 2006), among others. Nevertheless, there exist only a few works related to inventive problem solving or inventive design. This section describes some of the most relevant works observed from two different perspectives, inventive design and knowledge sharing, which are the main topics of this work.

Prickett and Aparicio (2012) describe a TRIZ technical system ontology, which aims to assist users in undertaking design engineering tasks by indexing and reusing knowledge extracted from patents or other knowledge sources. The ontology is also employed to support the elicitation process and facilitate the analogical reasoning through the TRIZ theory. However, it was partially developed and did not take into account a collaborative solving process. A similar direction was taken by Zanni-Merk *et al* (2011). These authors conceived a TRIZ-based ontology to formalize the main concepts concerning inventive design, which could be applied from the problem formulation step to the evaluation of potential solutions. This ontology was implemented in a solution that integrates several TRIZ tools and concepts. Nevertheless, the ontology has a prerequisite: a minimal TRIZ training is necessary due to the tool complexity. In addition, the ontology was not open to a collaborative problem solving process. Soo *et al* (2012) propose to extract knowledge from patents to assist the invention process through a cooperative multi-agent platform. This platform was conceived to retrieve and analyze patents to extract information with the aid of ontology and basic semantic web techniques. The ontology did not use the concept of function or substance-field analysis as a solving strategy. The main tool advantage is that the ontology reduces the solution space. Choi *et al* (2012) applied the Function-Oriented Search as a tool for retrieving information related to functions in patent databases. Function-Oriented Search was the mechanism to identify similarities in different domains in order to transfer technology and solutions. The authors did not use substance-field analysis, and the ontology did not support collective work.

Zanni-Merk *et al* (2009) propose an ontology to assist conceptual design. This ontology was centered on the main TRIZ concepts. The ontology is also useful to generate a framework to acquire and capitalize the knowledge revealed when solving inventive problems. This ontology was structured as a desktop application and did not support collaboration. Kitamura *et al* (2004) developed an ontological framework for capturing the associated knowledge deployed in the conceptual design phase. The objective was to facilitate the knowledge transfer among different domains. According to the authors, this framework was useful to make explicit the implicit knowledge possessed by designers, and transform it in such a way that it could be shared collectively. Yoon *et al* (2011) argue that technology reuse can reduce the time, effort, and cost in R&D activities. Nonetheless, this objective can only be achieved if a common perspective about functions is established. Thus, this work states that functions or functional characteristics in a system should be considered as a pointer to support reutilization. The authors proposed an ontological functional modeling methodology. The ontology uses only WordNet-based representations. Another work that keeps a similar orientation about reusability was published by Lasheras *et al* (2009). The authors underline that the reuse of requirements improves the quality and productivity of software process and products. The concept of functions is not considered in this article, neither is a collective dimension to carry out this kind of analysis. Park *et al* (2012), proposed a function-based patent intelligent system to extract information about functions from textual patent information using semantic analysis. The system uses the concept of function, but the ontology is not at its core and neither was a collective dimension of the process taken into account. Rahmani and Thomson (2012) conceived a piece of software that operated on port information and control of subsystem interfaces during collaboration. Thus, an ontology was conceived to define a set of ports attributes that are derived from its function and forms. This article focuses on collaborative design and also employs the concept of function, but it does not suggest any guide for problem solving. Kim *et al* (2006) proposed a new assembly design information-sharing framework and an assembly design browser for collaborative product development. The browser used an ontology with the purpose of representing engineering relations or design requirements. This browser did not consider functions as a strategy to create a homogenous comprehension of design requirements. Lau *et al* (2009) proposed a different strategy to solve problems using the Case-Based Reasoning approach. The authors proposed an ontology-based similarity measurement to retrieve the similar sub-problems that overcome the synonym problems in case retrieval. However, no resource to overcome typical Case-Based Reasoning drawbacks was offered (e.g. the condition where no similar problems are available in the case memory) or the use of functions as a vector to evaluate similarity was not provided. Table 1 summarizes the state of the art.

Author	**Functions are used**	**Ontologies are considered**	**Collaborative work**	**Capitalize knowledge**
Prickett and Aparicio, 2012	Yes	Yes	No	Yes
Soo, Lin, Yang, Lin and Cheng, 2012	No	Yes	Yes	Yes
Choi, Kang, Lim and Kim, 2012	Yes	Yes	No	Yes
Park, Kim, Choi and Yoon, 2012	Yes	No	No	Yes
Rahmani and Thomson, 2012	No	Yes	Yes	Yes
Zanni-Merk, Cavallucci and Rousselot, 2011	Yes	Yes	No	Yes
Yoon, Lim, Choi, Kim and Kim, 2011	Yes	Yes	No	Yes
Zanni-Merk, Cavallucci and Rousselot, 2009	No	Yes	No	Yes
Kitamura, Kashiwase, Fuse and Mizoguchi, 2004	No	Yes	No	Yes

Table 1: Comparison of different approaches

The review of several articles from different points of view shows that inventive design and inventive problem solving are research domains not yet thoroughly explored in depth. Nevertheless, important research guidelines come into view. Among the most relevant are: (1) Ontologies can provide a framework to assist knowledge reuse in order to improve the inventive problem solving process. (2) Collaborative problem solving needs to be considered as a capital requirement, especially because knowledge is not geographically located. Finally, (3) it is possible to integrate new tools and approaches to assist the process of innovation (i.e. TRIZ, Design for Six Sigma, Quality Function Deployment). This article describes an ontology useful to create a model of a function. Then, this model is employed as a pointer to guide problem formulation and the solving process, but also as a tool to capitalize and share the knowledge deployed in this process. The next section briefly describes a function-oriented ontology called Su-Field App Tool.

2. Function-Oriented Ontology

The concept of function is essential to develop the ontology, and it is even more important when it is regarded from the Theory of Inventive Problem Solving (TRIZ) insight. The next paragraphs describe the TRIZ perspective about functions and how the concept of function was transformed into a tool for modeling and solving problems.

The Theory of Inventive Problem Solving (TRIZ) Foundations

It is possible to state that the ability to solve problems, more specifically non-routine or inventive problems, is the cornerstone of technology. The Theory of Inventive Problem Solving (TRIZ) is an approach for solving non-routine problems based on the evolution of technical systems. TRIZ defines technical problems as the ones "*for which at least one critical step to a solution as well as the solution itself is unknown as the inventive problems*" (Savransky, 2000; Mann, 2003). TRIZ is based on this concept: engineering and scientific knowledge could be transversally applied. In order to transfer this knowledge, an essential requirement should be satisfied: a machine or an expert must recognize similar features among different domains and problems. Thus, ontologies could be developed to reveal this similarity and create a shared perspective about a problem. In order to describe the integration of ontologies and the concept of function from the TRIZ theory, it is necessary to describe what a function is.

The Concept of Function

Technical systems are built on a fundamental premise: to provide at least one useful function. From a different perspective, functions are the essence of any system; they are the element that impels technical evolution. Even though, useful functions also frequently produce harmful or undesired functions. According to TRIZ, all functions can be decomposed into three essential elements: two substances and an energy field, also frequently called object, tool and energy respectively. Thus, to exist, any function must have at least these three elements. The following corollaries derive from this statement:

- Bringing together the adequate three elements could provide any function.
- Removing the relationship among substances or between the field and one substance could eliminate any function.
- Inventive problems could be formulated when the relationship among the three basic elements produce a harmful, insufficient, excessive, uncontrollable, or even superfluous function.
- Functions usually remain, but the way in this function is produced continually evolve.
- The mechanism able to produce a function could be transferred to different technical domains.

This is the central topic of this paper: to conceive a function-oriented ontology in order to solve inventive problems that could be applied in different domains. A TRIZ tool called substance-field analysis, which is also a function-oriented problem-solving tool, is then paramount to this article. The next section concisely describes this approach and how it was applied to problem solving.

2.1 Substance Field Analysis (SFA)

The Substance-Field Analysis is a modeling approach used in TRIZ to describe the synthesis of physical structures, processes, or interactions that take place in a technological system. It is also useful to represent the condition, state, or behaviour of a system. A system is considered as a set of objects, as well as the interactions among them, that have features or properties not reducible to the features or properties of separate objects. The TRIZ perspective about functions established that the minimal technical system configuration capable of performing a function is constituted of three elements that interact together: two substances and one field. This configuration is also identified in undesired functions. Therefore, any object is a system, and the Substance-Field Analysis can deal with problems associated with its physical composition, performance, and relation to other systems. In technical aspects, the term "substance" has a wider meaning than in ordinary language: it refers not only to a variety of matter, but also to technical systems or their parts, as well as to the external environment and even living organisms. The states of substances include a huge variety of physical states: vacuum, plasma, gas, liquid, and solid, but also a large number of in-between and compound states (Savransky, 2000). The notion of "field" in Substance-Field Analysis (SFA) is also distinct from its definition in physics.

A field is an interaction characterized by a flow of energy (of any type), information, or mechanical force generated by a substance, and which potentially impacts other substances to perform an effect. Fields are typically classified as Gravitational, Mechanical, Acoustic, Thermal, Chemical, Electrical, Magnetic, Electro-Magnetic, Biological, and Nuclear field. However, SFA uses more detailed classification of fields in order to differentiate a flow of energy (i.e., mechanical field encompasses pressure, impact, vibration, and impulse, to mention a few) (Altshuller, 1984). The ontology developed to represent a function uses these fields to build a model of any given system and six basic relationships: useful, harmful, insufficient, action missing, excessive action, and superfluous action.

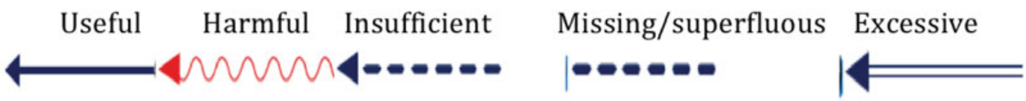


Figure 1. Basic relationships among substance and a field (Altshuller, 1999)

Once a substance and a field were defined, and the basic relations among these components were established, it is possible to create a model of a minimal technical system.

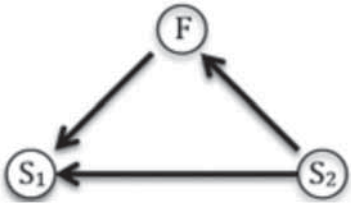


Figure 2A: Minimal Useful Technical System

Figure 2B: Incomplete Technical System

Example: Available methods for determining water content in machine oil demand several resources (time, special measuring equipment, among others) and are usually complex. A minimal system to detect water in machine oil must be proposed. The system should be as simple as possible and to provide the function "Detection". There are two substance S_1 (Oil) and S_2 (Water) and a relation between both, but the field is missing (Figure 2B). A flow of energy should be thus added in order to provide a useful function. The field has to be simple and easily available, and has to create an effect to reveal whether water is mixed in oil. The nature of the substance combined with the right field should provide the function. In this case, a thermal field can be added (Figure 3). As it was documented in Japanese Patent No. 5246837, the integration of a thermal field in the system can solve the problem if it is combined with particular properties from both substances, such as the boiling point (Salamatov, 1999).

Field (Nomenclature)	Field (Nomenclature)
Gravitational ($\mathbf{F_G}$)	Electrical ($\mathbf{F_E}$)
Mechanical ($\mathbf{F_M}$)	Magnetic ($\mathbf{F_{Mg}}$)
Acoustic ($\mathbf{F_A}$)	Electro-Magnetic ($\mathbf{F_{EM}}$)
Thermal ($\mathbf{F_T}$)	Biological ($\mathbf{F_B}$)
Chemical ($\mathbf{F_C}$)	Nuclear ($\mathbf{F_N}$)

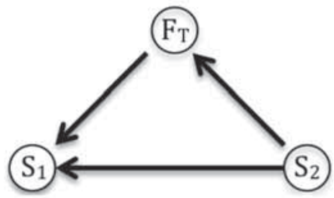


Figure 3: Field addition to complete a technical system

Fey and Rivin (2005); Rantanen and Domb (2008); Orloff (2012) and Zhang *et al* (2009), proposed another way to model a function: where such function involves two components. One of these components is an "object" that should be manipulated, transformed, or modified. The object usually does not exert any self-control. Then another component, called "tool", is required to execute an operation. If both components are present and an action is connecting them, then a function could be described. Nouns are used to describe objects and verbs to describe actions (Figure 4).

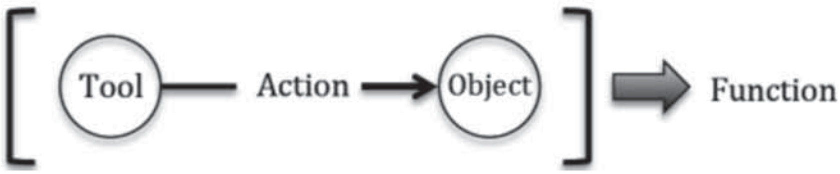


Figure 4: Field addition to complete a technical system

This way of representing functions is widely applied in value engineering (Kawakami *et al*, 1996) and software (Cortellessa *et al*, 2005), among a huge range of domains. However, although functional analysis is useful to schematize interaction in a system and identify conflicts, function analysis does not offer specific guidelines for problem solving as Substance-Field Analysis can do (Altshuller, 1984; 1999). Substance-Field Analysis (SFA) emphasizes the fact that if two problems result in identical or similar problem models, then they have identical or similar strategies of solutions. In consequence, SFA has a set of standard strategies – called inventive standards – for problem solving, which was synthesized from a huge knowledge capitalization effort over global patent databases. This set of solving strategies could be applied in a very versatile way. For instance, it can be used (1) to formulate inventive problems; (2) to solve problems where a system needs to be changed, but there is not a clear way to undertake this situation; (3) when there is a problem that demands a mechanism or way to measure or detect something within a system; and also (4) when it is necessary to identify evolution patterns in a system, to mention some of their most important applications. There are five different classes of inventive standards (Altshuller, 1999; Salamatov, 1999; Savransky, 2000):

Class 1: Composition and decomposition of Substance-Field Models System (SFMS)

Class 2: Evolution of SFMS

Class 3: Transitions to supersystem and microlevel

Class 4: Measurement and detection standards

Class 5: Helpers

Inventive standards are basically a set of "If-Then-Else" rules: If <condition A>, <condition B>…n are present, Then <Recommendation of inventive standard> (See table 2). It is important to notice that rule-based systems had revealed a limitation: its capacity to evolve rapidly involves a lot of effort because new situations do not always fit all the rules, and the system needs to be updated among other problems (Sun, 1996). An alternative to solve this problem is to create meta-rules that could minimize the set of effective rules and thus, accelerate adaptation or evolution (Cazenave, 2003). Inventive standards were conceived as meta-rules that could be applied in different fields and situations. Consequently, inventive standards could overcome the problem of evolution in rule-based systems, or at least, reduce its impact. Table 2 shows how inventive standards are composed. There are 76 standard solutions divided into five classes, and several algorithms have been proposed to use them. Terninko *et al* (1998) formulated one of the most useful. This algorithm is composed of five essential activities: (1) to identify the right elements, (2) to construct the substance-field model of the system, (3) to select one inventive standard, (4) to develop conceptual solution – also called adaptation process; and finally, (5) to evaluate the solution.

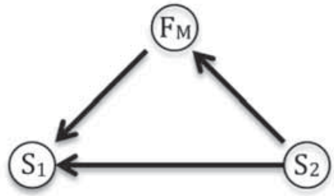
Initial condition	Recommendation	Example
Inventive standard 1.1.1 S, F  Condition: two substances available, or one substance and one field.	*If "there is an object which is not easy to change as required, and the conditions do not contain any limitations on the introduction of substances and fields, the problem is to be solved by synthesizing a SFM: the object is subjected to the action of a physical field which produces the necessary change in the object"*	To remove air (S1) from a powdered substance, the substance (S2) is subjected to centrifugal forces (F). 

Table 2: Description of one Inventive Standard (Salamatov, 1999)

In Terninko's algorithm, if the model is complete, then there are at least two substances and one field interacting, and therefore it is possible to select one inventive standard according to the class and group that best matches with the initial situation. If the system is effective, the solution should be implemented; otherwise, an inventive standard should be employed. Once this first stage has been granted, a loop starts until at least one effective solution is proposed. The process to adapt an inventive standard (which is a generic recommendation) demands a creative effort. This effort is not captured or stored with the aim of reducing effort for another person if a similar situation arrives. Consequently, Substance-Field Analysis and inventive standards are knowledge-based tools for problem solving. Nevertheless, even though these tools have proved to be effective to solve inventive problems (Zhang *et al*, 2009; Rousselot *et al*, 2012; Zanni-Merk *et al*, 2009; Zanni-Merk *et al*, 2011; Prickett and Aparicio, 2012), a formal approach to capitalize knowledge and create a shareable knowledge structure to assists its application is still under development. The next section succinctly describes the concept of ontology and the function-oriented ontology for inventive problem solving.

2.2 Ontologies, an Approach to Capitalize Knowledge

There are several definitions for ontologies and, although there is not a generalized concept, some of these definitions have a wider acceptation in computer sciences and cognitive science: "*a specification of a conceptualization*" (Gruber, 1995), "*explicit formal specifications of the terms in the domain and relations among them*" (Gruber, 1993), "*a knowledge representation of a domain or of a field, which provides conceptual resources for knowledge based systems*" (Zanni-Merk *et al*, 2011). An ontology is usually composed of concepts in a specific domain, properties, and attributes of the concept that are organized in such a way that their relationships could be employed to represent a knowledge structure. According to Noy and McGuinness (2001), ontologies provide several functions, and among the most important are:

- To share common understanding of the structure of information among people or software agents
- To enable reuse of domain knowledge
- To make domain assumptions explicit
- To separate domain knowledge from operational knowledge
- To analyze domain knowledge

All functions offered by ontologies are important in this article; yet reusing knowledge and making domain assumptions explicit are essential functions in order to increase performance when solving inventive problems. The methodology presented by Noy and McGuinness (2001) was employed to develop the ontology. A documental analysis and an assembled panel of seven experts from various domains (e.g. computer science, industrial engineering, TRIZ experts, and industrial designers) took part in this effort. The teamwork defined through a consensus the common language, which was extracted from several solved cases, and which also provided a problem solving process. Figure 5 schematizes the relationship among the elements that enables the description of problems. It is important to highlight that this process utilizes the explanations offered by users while describing, interpreting, and solving problems as a support for reasoning (Goguen *et al*, 1983).

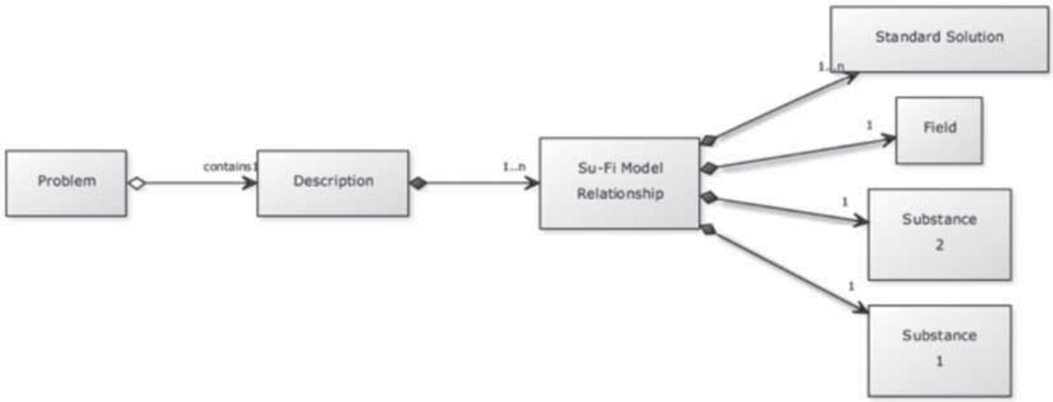


Figure 5: Elements Considered in the Ontology

The next paragraphs describe how the main features of the problem solving process were identified with the goal of transforming these features into design parameters for the ontology.

3. Identification of Essential Requirements

The resulting ontology is a collective effort to define common terms, and it specifies the representation of the different pieces of knowledge that are involved in a substance-field model. The ontology also provides a particular functionality: it can be applied in a wide range of domains due to the nature of the substance-field models. The ontology also generates the conditions for assuring long-term learning. At its core, there is a knowledge base, where generic knowledge episodes are combined with specific applications in a specific domain. To assure this functionality, the vocabulary to define a model and their intrinsic interactions were clearly defined by an expert panel. The same set of experts and the same process was deployed to identify, classify, and reutilize relationships and class attributes. As a result, the ontology creates a community of practice that has a common understanding of Substance-Field, about its concepts and inventive standards. The ontology also verifies that all defined interactions are present in order to construct an adequate system model. The expert panel defined several requirements as fundamental premises for conceiving the ontology:

1. The piece of software must help users to create a minimal functional system or Substance-field model (Code A)
2. The software must contain basic fields (Code B)
3. The software must permit the use of a huge diversity of substances (Code C)
4. The software must be flexible enough to be applied in several domains (Code D)
5. The software must guide users to select the right inventive standard (Code E)
6. The software must remember past solutions (Code F)
7. The software must be as simple as possible (Code G)

These requirements must be ranked to offer a piece of software useful and capable of satisfying the minimal user requirements. To obtain this relative importance, the Analytic Hierarchy Process (AHP) was employed. This tool is useful to transform subjective information into a clear decision framework (Vaidya and Kumar, 2006). The AHP process is employed at the initial stages of the Quality Function Deployment, which is an approach useful to transform requirements into design parameters (Carnevalli and Cauchick-Miguel, 2008). To apply the AHP it is necessary to launch a paired comparison between features, and then values are normalized, and finally conclusions are derived. A scale of nine points is typically used to run the comparison:

1. Means, equal importance
2. Between 1 and 3
3. Means moderate importance
4. Between 3 and 5
5. Means strong importance
6. Between 5 and 7
7. Means very strong importance
8. Between 7 and 9
9. Means extreme importance

Tables 3 and 4 show how the AHP was applied in this article and in the final classification of the expert panel requirements. The code inserted in each requirement is used at this stage of the article.

Interpretation: from the expert's point of view, the ontology and its implementation must in the first place provide a mechanism to remember past solution with 30.86% of the relative importance. In the second place, the system must help users select the inventive standard (27.29% of the relative importance). The third place in relative importance corresponds to the ability to assist users in

Paired comparison							
	A	B	C	D	E	F	G
A	1	5	3	5	0.33	0.14	5
B	0.2	1	1	0.2	0.2	0.25	0.33
C	0.33	1	1	0.2	0.14	0.16	3
D	0.2	5	5	1	1	0.14	3
E	3	5	7	1	1	2	7
F	7	4	6	7	0.5	1	7
G	0.2	3	0.33	0.33	0.14	0.14	1
SUM	11.93	24	23.33	14.73	3.319	3.84	26.33

Table 3: Initial Step of AHP Process-Paired Comparison

Normalization										
	A	B	C	D	E	F	G	SUM	Importance	%
A	0.08	0.21	0.13	0.34	0.10	0.04	0.19	1.09	0.1557	15.57%
B	0.02	0.04	0.04	0.01	0.06	0.07	0.01	0.25	0.0357	3.57%
C	0.03	0.04	0.04	0.01	0.04	0.04	0.11	0.33	0.0471	4.71%
D	0.02	0.21	0.21	0.07	0.30	0.04	0.11	0.96	0.1371	13.71%
E	0.25	0.21	0.30	0.07	0.30	0.52	0.27	1.91	0.2729	27.29%
F	0.59	0.17	0.26	0.48	0.15	0.26	0.27	2.16	0.3086	30.86%
G	0.02	0.13	0.01	0.02	0.04	0.04	0.04	0.30	0.0429	4.29%
SUM	1	1	1	1	1	1	1	7.00	1.00	100%

Table 4: Relative Importance of Design Parameters

creating a minimal functional system (15.57%). The last characteristic that is recommended is the possibility for applying the tool in several domains with a relative importance of 13.71%. If the system conceived offers all four functionalities, then the user satisfaction will reach at least 87.43%. The ontology should be conceived and deployed with these requirements as a target. The next paragraphs describe the process to develop the ontology.

4. Ontology-based Tool and Web Implementation

The ontology was developed based on the expert panel's expertise. The next points briefly describe this process:

- **Step 1.** Determine the domain and scope of the ontology: The domain of application is function modeling and problem solving through the substance-field analysis. The ontology will help users define a problem and identify the substances, a field, the interaction among these elements, as well as a set of recommended inventive standards to solve this problem. These inventive standards have been successfully applied in different domains and represent generic solving strategies that should be adapted to fit in specific situations.

- **Step 2.** Consider reusing existing ontologies: Some ontologies have been developed and implemented in commercial software but are not available for reutilization. There are some other ontologies/vocabularies such as SKOS (Miles and Pérez-Agüera, 2007) or Dublin Core (Weibel *et al*, 1998), which could be worth reusing due to the concepts that they use. In this context, a first version of the ontology has been designed without the use of external elements. The idea of this process is the creation of the entire ontology from scratch. Once the ontology is fully defined, an analysis of the ontology would be performed to determine which elements of SKOS and Dublin Core could be reused.
- **Step 3.** Enumerate important terms in the ontology: Definitions about substances (also called tool and object), fields, and inventive standards were listed to build the ontology. The set of inventive standards was also considered. Inventive standards will keep a numeric notation: the first number represents the class; the second digit defines the group, and the rest defines a subclass of the group (e.g. 1.1.8).
- **Step 4.** Define the classes and the class hierarchy: The taxonomy developed is presented in Figure 6. A minimal functional technical system includes substances, fields, and interactions. A problem, condition, or behaviour in this minimal functional system encompasses a substance-field model, at least one of a set of inventive standards, and at least one solution. It is important to underline that the term "functional" does not make references to something useful. It is a neutral word denoting that a positive, negative, or incomplete function is the output of a system.

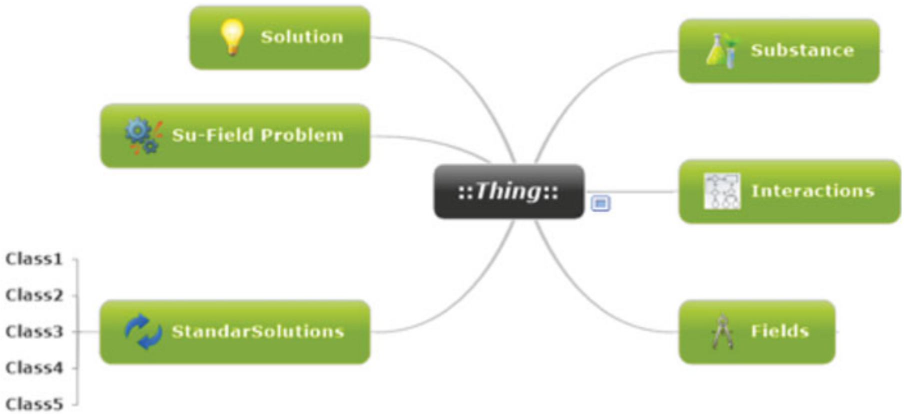


Figure 6: Basic Taxonomy

- **Step 5.** Define the properties of classes–slots: Table 5 (on the following page) describes the structure of the concepts.

  Once the classes and slots have been defined, the basic relationship among these elements is established. This relationship explains how the instance of a class is related to another.
- **Step 6.** Define the facets of the slots: Slots can have several facets that describe the type of value, the values allowed, the cardinality, and any other features of the values that a slot can take. According to the TRIZ theory, a minimal technical system or Substance-Field Model contains three elements: two substances, one field, and a set of interactions. Thus, `SFM= problem model {contains substance1; substance2; field}`. Therefore, a problem modeled as a substance-field model contains at least one substance 1, one substance 2 and one

Class	Slot	Description
Problem	Problem_name	A short identifier of the situation to be modeled
	Description	Basic information to assure that the initial situation is clear to any user
Substance	Substance_name	A short identifier for something that means not only a variety of matter, but also technical systems, the parts of this system, the external environment, or even living organisms
	Physical_state	The state of substances include a huge variety of physical states such as vacuum, plasma, gas, liquid, and solid; but it also comprises a large number of in-between and compound states
Field	Field_name	Name of the flow of energy or information which acts to perform an effect
	Description	Basic information about how this flow of energy or information is involved in the situation
Interaction	Name	A short identifier to catalogue the kind of interaction that exists between a substance and a field
Class1, Class2, Class3, Class4 Class5	Inventive_standard_ name	A numeric identifier that respresents the class, group, and subgroup of a particular solving strategy
	Description	It represents a generic strategy to transform a minimal functional technical system in order to reach an objective
Solution	Description	It explains how a desirable transformation is fulfilled through the adaptation of inventive standards

Table 5: Main Classes and Slots

field. This relationship was useful to determine the cardinality, slot value type, domain, and range of a slot.

- **Step 7.** Create instances: This is the step where specific individual instances of classes in the hierarchy are created. Several Substance-Field Models (SFM) were introduced in the hierarchy for further validation.

Once all the steps of the methodology have been covered it is necessary to develop the ontology. The Protégé-OWL editor was selected to complete the task. The Web Ontology Language (OWL) is practically the standardized ontology language. According to the OWL Web Ontology Language Guide (W3C, 2009), an OWL ontology may include descriptions of classes, properties, and their instances. If this ontology is created, then the OWL formal semantics details how to infer its logical consequences (i.e. facts not literally present in the ontology, but entailed by the semantics). Once all elements involved in the ontology have been introduced into the editor, the resulting ontology is validated with Pellet, which is a reasoner included in Protégé. Figure 7 (on the following page) shows a graphical description of the function-oriented ontology and one solved problem.

With the ontology validated, it is necessary to elaborate the mechanism to be able to capture, store, and reuse information (Table 6). To conceive a set of interfaces capable of exploiting the functionalities of the ontology, it is necessary to consider the hierarchy defined by the expert panel.

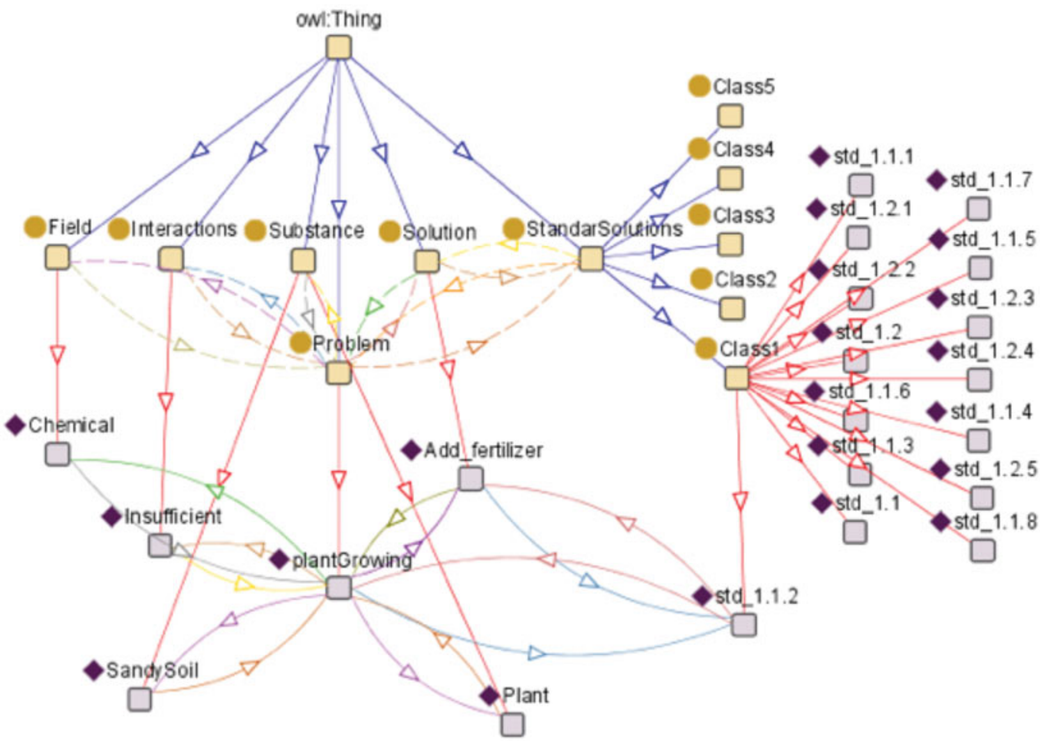


Figure 7: Function-oriented ontology

No.	Requirement	Importance	Proposed mechanism/process
1	To represent a minimal substance field model (MSFM)	15.57%	An interface capable of representing a MSFM graphically must be conceived and implemented.
2	Application in several domains	13.71%	The system must be deployed in different domains, even within no technical ones. The versatility is thus assured if users can define their substances according to their interests. Relationships among two substances and one field could be defined in advance.
3	To assist users in the selection of at least one useful solving strategy also called inventive standards	27.29%	To deploy an algorithm to select inventive standards. Savransky (2000) proposes an algorithm that was successfully tested in several domains to selected the adequate inventive standard.
4	To remember past solutions	30.86%	To create a framework that uses MSFM as a problem model and inventive standards as a structure to store solutions. Thus, the ontology must be able to store and retrieve past solved problems.

Table 6: Proposed Mechanisms

To implement these functionalities several technologies were necessary such as Jena, which is a Java-based framework for building semantic web applications. It offers several tools and Java-based libraries to develop semantic web and linked data applications, tools, and servers (Jena, 2012). A database was required, and SPARQL was selected for being an open source object-relational database system (SPARQL, 2012).

SuFiApp Architecture

SuFiApp (standing for Substance-Field Analysis Application) is the name of this ontology and its associated web service. The goal of this service is to provide an environment to model, solve, and capitalize the knowledge exposed when the Substance-Field Analysis approach is applied. The service has a layered architecture composed of three levels of abstraction: interface, service, and data access (Figure 8).

Interface layer: This layer uses Apache Jena, which is a Java-based framework to implement semantic web applications. Jena can read, process, and write RDF data, as well as handle OWL and RDFS ontologies. This interface is responsible for publishing services, and visualizing and integrating all data to represent a problem, which as previously mentioned, involves two substances, one field, the interaction among these elements, and finally, the associated solving strategy.

Service layer: This layer is where the most important processes of the system take place. It is where the core ontology and the substance field analysis modeling are located. Furthermore, a query is executed to extract similar problems, a similar substance, or similar solving strategies from the database.

Data access layer: It is the knowledge structure of the system. In this layer the search and rules engines, a SQL database and specific knowledge about the Substance-Field Analysis are stored.

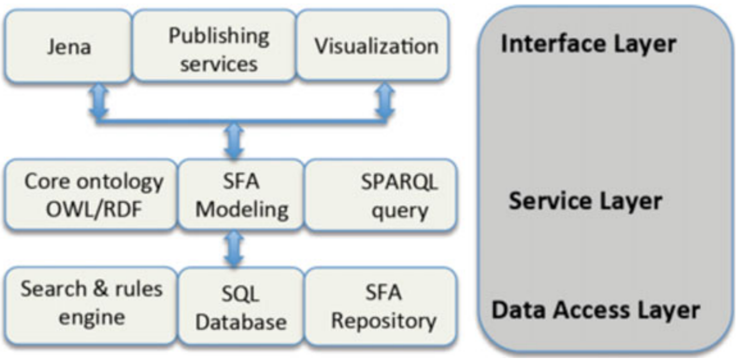


Figure 8: Substance Field Analysis Architecture

The next section presents a case study where the main features of the function-oriented ontology are described. The purpose of this case study is to demonstrate how the main design requirements were satisfied. The problem to solve concerns a big Mexican mining enterprise.

5. Case Study: The Mining Problem

The initial step of the application is useful to gather information about the problem in order to build a Substance-Field Model (SFM). The initial dialogue box demands the problem name, the selection of two substances, one field, and a relation that best describes the system's condition. Four possibilities are available in this dialogue box: (1) to create a new problem, (2) to edit problem,

(3) to edit substance, and (4) to identify an inventive standard. Also, a textual description of the problem is inserted in this window. With these elements, it is possible to create a graphical representation of the SFM. The case study is simultaneously developed as the system is described. It concerns an old problem in the mining industry.

Problem description: The problem is related to a high-risk working condition. It concerns a drilling rig specially built for underground purposes, which makes a hole where a calculated amount of explosive will be placed afterward. The purpose of this fragmentation bomb is to facilitate the extraction and transportation of a huge amount of mineral, which will be later processed to extract something valuable. The expert makes several perforations to increase productivity. Nevertheless, after the explosion these perforations cannot be used because they get filled with mineral, and the drilling rig cannot return inside the mine to dig in the holes because soil conditions are unsafe after the explosion. However, the mine manager needs to drill more holes in order to extract more minerals. How should this problem be resolved? With the information presented above, it is possible to construct a SFM:

- Substance 1: borehole
- Substance 2: explosives
- Field: Chemical
- Interaction: harmful. Explosives are necessary to break up rock, but should not be there because explosion fills up the boreholes.

It is important to underline that the user can create its definition of what a particular substance is, and then use it later in a Substance-Field Model (SFM).

Figure 9 concentrates information about the problem. With this information, the Savransky's algorithm could be launched. This algorithm is not described in this article due to its extension, but it uses a set of questions to guide the selection of the most suitable inventive standard (Savransky, 2000). At this point in the process, the problem has a model, and it is associated with at least one inventive standard or a set of standards. The next step of the process demands the

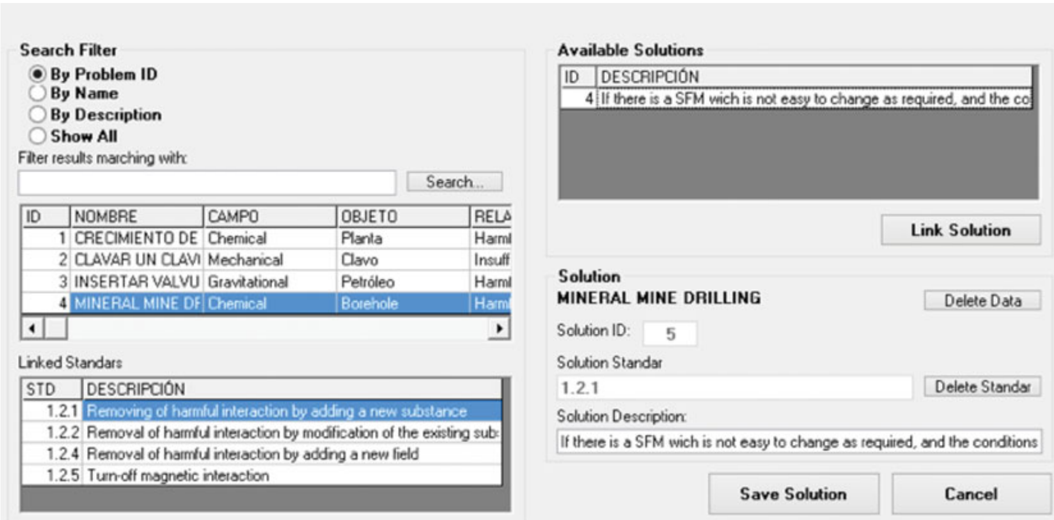


Figure 9: Inventive Standards as a Knowledge Repository

adaptation of the inventive standard to fit in a particular situation. This activity requires a creative effort because generic knowledge is transformed into valuable solutions. It is also at this step where the user will have access to similar past problems. If a similar problem was solved in the past, then this information is part of the standard, which acts as a knowledge repository that stores all similar problems. The user needs to select the experience using the Substance-Field Model and the explanation offered for other users (Figure 9).

Proposed solutions: The inventive standard 1.2.1 explains that if there is a SFM, which is not easy to change as required, and the conditions contain limitations related to the introduction of additives into the existing substances, the problem can be solved through a transition (permanent or temporary) to an external complex SFM. An association performs this transition. The association means to attach to one of these substances, another external substance. The external substance improves controllability or brings the required properties to the SFM. Thus, a third substance needs to be added to the model. This substance should be placed inside the borehole to avoid minerals to take inner space, but it should not be there because explosives need to be placed. Consequently, ice can control inner space. When ice is there, the inner space is not available, and then, no mineral will take this place. Temperature in the mine will reduce ice, and the soil will entirely absorb humidity. Water does not affect explosives or any of its characteristics (Figure 10). This solution was one among several. It was selected as a potential solution based on three criteria:

1. It is simple; hence, it can be rapidly applied to the problem
2. Its relation cost-benefit is highly positive
3. The global complexity of the system is not incremented

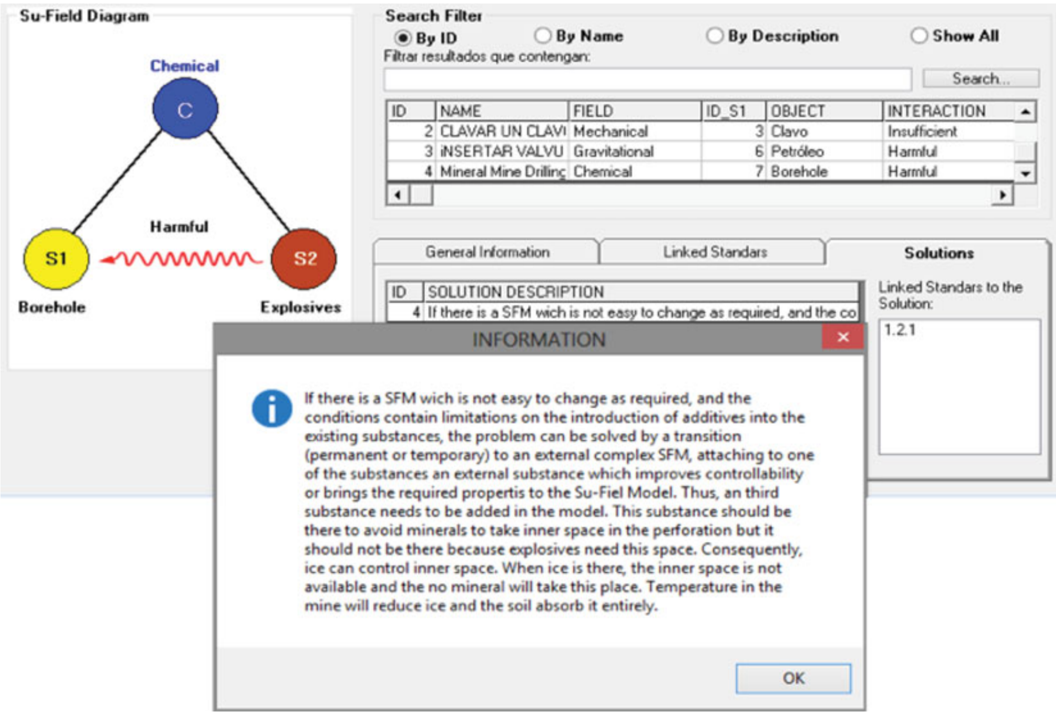


Figure 10: Recommendation for Solving the Problem

6. Discussion and Evaluation

The function-oriented ontology was conceived to provide four essential requirements:

1. To represent a Minimal Substance-Field Model (MSFM)
2. A versatile application able to be deployed in different domains
3. To assist users in the selection of at least one useful solving strategy (Standard solution)
4. To conceive and implement a mechanism to store, retrieve, and remember past solved problems using SFM as strategy to classify problems and inventive standards to store specific solutions.

Comparison and Evaluation

The function-oriented ontology was compared with three most popular inventive problem solving software: Goldfire Innovator from Invention Machine (inventionmachine.com), Innovation Suite from CREAX (creaxinnovationsuite.com) and Innovation Workbench from Ideation International (ideationtriz.com). All these commercial software were evaluated under the next criteria: (C1) The tool can store and reuse problems and past solutions; (C2) The tool uses substance-field models to index and retrieve problems; (C3) The tool offers access to other knowledge sources; (C4) The tool offers a collaborative solving problems process; (C5) The tool uses other TRIZ tools, and finally, (C6) The tool uses ontologies, even at some stage of the solving process. The evaluation considers three possible answers: ✓= the tool satisfies the criterion. X = the tool does not satisfy the criterion and also NAI: Not available information. Table 7 summarizes this qualitative evaluation:

Application	C1	C2	C3	C4	C5	C6
Goldfire	✓	X	✓	X	✓	✓
Innovation Suite	✓	X	✓	X	✓	NAI
Innovation Workbench	✓	X	✓	X	✓	X
SuFiApp	✓	✓	X	✓	X	✓

Table 7: Software Comparison

The analysis shows that in commercial software it is not possible to index, retrieve, or reuse problems and solutions using the substances or fields involved in a problem. The hypothesis of this article is that if two problems share the same SFM, then the solution of the first one could be transferred or adapted to the second one. This hypothesis also means that these problems share similar substances and fields. This capacity is not considered in commercial software. It is also evident that commercial software does not consider that inventive problem solving is a social activity, which demands interactions among several stakeholders. This reveals an interesting opportunity for further research. It is also important to notice that all available software uses past solved problems to create a memory, but they do it under different approaches. Available software (e.g. Goldfire Innovator), also offers access to different knowledge sources while SuFiApp has only one unique knowledge base: its memory and personal knowledge from stakeholders. The evaluation also reveals that all available commercial software uses several TRIZ tools, whereas SuFiApp uses only one TRIZ tool. This condition provides a new opportunity for research.

Finally, it is possible to state that ontologies are a widely accepted approach for assisting inventive problem solving. The research team, which collaborated in this article, assures that ontologies will increase its application in domains such as inventive design, conceptual design, and in innovation problem solving. The integration of substance-field analysis and ontologies creates the conditions to implement a piece of software able to guide problem solving in several domains, while ontologies are usually conceived to solve specific problems. The expert panel evaluated the following processes in order to determine the usefulness of the SuFiApp: (1) the mechanisms implemented to guarantee that the system is capable of representing graphically a SFM. (2) The mechanisms implemented to assure that the piece of software has enough versatility to be applied in several domains, and thus it offers the users the possibility to define their substances according to the problem and their technical domain. (3) A process to assist the user in selecting inventive standards. (4) Creating a framework that uses SFM as a problem model and inventive standards as a structure to capitalize knowledge. The case study shows that the system can perform the functionalities that were defined by the expert panel.

7. Conclusions and Perspectives

This article presents a function-oriented ontology that has several advantages. For instance, the piece of software developed has the ability to model problems in a very simple but effective manner: by defining two substances, one field, and their interactions in such a way that a function could be described. Consequently, the ability to model functions has a relevant technological impact: a function could be understood if a graphical representation is used. Therefore, several semantic problems could be avoided, and this is a condition that makes possible the transfer of knowledge or successful solving problem strategies among different technical domains in a different way. This is, in fact, one of the main motivations for this article. There are very limited offers of tools to assist technology transfer or mechanisms to systematize this process. The function-oriented ontology is an alternative for developing a framework useful to model functions, correlate this function with a set of inventive strategies for problem solving, and capitalize the knowledge deployed in this process.

In this article, the function-oriented ontology represents only a minimal technical system, and thus, the system needs to evolve in order to face problems of higher complexity, which demands new knowledge resources and more flexible tools. The TRIZ theory can provide these tools and the Artificial Intelligence, techniques that allow the mobilization of knowledge among several domains. The SuFiApp stores all problems and solutions. This condition will lead over time to a huge knowledge repository that would demand the integration of new techniques to retrieve and select similar problems. The Case-Based Reasoning approach has the tools and mechanisms to deal with this issue. Finally, it is necessary to mention that the concept of function and inventive standards could be employed as a meta-ontology in order to create a problem solving process useful in different domains.

Acknowledgements

This work was supported by the National Council of Science and Technology (CONACYT) and the Tecnologico Nacional de Mexico. We also recognize the support of the ROPRIN.org network.

References

ALTSHULLER, G. (1984): *Creativity as an Exact Science*. CRC Press; 1st edition.

ALTSHULLER, G. (1999): Innovation Algorithm: TRIZ, systematic innovation and technical creativity. Ed. Technical Innovation Center, 1st edition.

CARNEVALLI, J. and CAUCHICK-MIGUEL, P. (2008): Review, analysis and classification of the literature on QFD – Types of research, difficulties and benefits. *International Journal of Production Economics* 114 (2): 737–754.

CAVALLUCCI, D. and ELTZER, T. (2011): Structuring knowledge in inventive design of complex problems. *Procedia Engineering* 9: 694–701.

CAZENAVE, T. (2003) Metarules to improve tactical Go knowledge. *Information Sciences* 154 (3–4): 173–188.

CHOI, S., KANG, D., LIM, J. and KIM, K. (2012): A fact-oriented ontological approach to SAO-based function modeling of patents for implementing Function-based Technology Database. *Expert Systems with Applications* 39(10): 9129–9140.

CORTELLESSA, V., DI MARCO, A., INVERARDI, P., MANCINELLI, F. and PELLICCIONE, P. (2005): A Framework for the Integration of Functional and Non-functional Analysis of Software Architectures. *Electronic Notes in Theoretical Computer Science* 116(19): 31–44.

DE CLERCQ, P., HASMAN, A., BLOM, J. and KORSTEN, H.M. (2001): The application of ontologies and problem-solving methods for the development of shareable guidelines. *Artificial Intelligence in Medicine* 22 (1): 1–22.

FEY, V. and RIVIN, E. (2005): Innovation on Demand: New Product Development Using TRIZ. Cambridge University Press.

GOGUEN, J., WEINER, J. and LINDE, C. (1983): Reasoning and natural explanation. *International Journal of Man-Machine Studies* 19(6): 521–559.

GRUBER, T.R. (1995): Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies* 43(4–5): 907–928.

GRUBER, T.R. (1993): A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition* 5: 199–220.

HIPPE, S. (1996): Design and application of a new knowledge engineering tool for solving real-world problems. *Knowledge-Based Systems* 9(8): 509–515.

Jena Apache available at http://jena.apache.org under Documentation.

JUNG, Y. and KIM, M. (2011): Ontology-based Dynamic Role Interaction Control in Multi-Agent System. *Journal of Research and Practice in Information Technology* 43(3): 209–225.

KAWAKAMI, H., KATAI, O., SAWARAGI, T., KONISHI, T. and IWAI, S. (1996): Knowledge acquisition method for conceptual design based on value engineering and axiomatic design theory. *Artificial Intelligence in Engineering* 10(3): 187–202.

KIM, K.Y., MANLEY, D. and YANG, H. (2006): Ontology-based assembly design and information sharing for collaborative product development. *Computer-Aided Design* 38(12): 1233–1250.

KITAMURA, Y., KASHIWASE, M., FUSE, M. and MIZOGUCHI, R. (2004): Deployment of an ontological framework of functional design knowledge. *Advanced Engineering Informatics* 18(2): 115–127.

LASHERAS, J., VALENCIA-GARCIA, R., FERNANDEZ-BREIS, J. and TOVAL, A. (2009): Modelling Reusable Security Requirements based on an Ontology Framework. *Journal of Research and Practice in Information Technology* 41(2): 119–133.

LAU, A. TSUI, E. and LEE, W. (2009): An ontology-based similarity measurement for problem-based case reasoning. *Expert Systems with Applications* 36(3): 6574–6579.

LIUA, X.Y., LIA, Y., PANB, P. and LIA, W. (2011): Research on computer-aided creative design platform based on creativity model. *Expert Systems with Applications* 38(8): 9973–9990.

MANN, D. (2003): Better technology forecasting using systematic innovation methods. *Technological Forecasting and Social Change* 70(8): 779–795.

MILES, A. and PÉREZ-AGÜERA, J.R. (2007): SKOS: Simple Knowledge Organisation for the Web. *Cataloging & Classification Quarterly* 43(3–4): 69–83.

NOY, N.F. and McGUINNESS, D.L. (2001): Ontology development 101: A guide to creating your First ontology. Stanford Knowledge Systems Laboratory, Available at http://protege.stanford.edu/publications/ontology development/ontology101.pdf.

ORLOFF, M. (2012): Modern TRIZ: A Practical Course with EASyTRIZ Technology. Ed. Springer-Verlag.

PANDIT, A. and ZHU, Y. (2007): An ontology-based approach to support decision-making for the design of ETO (Engineer-To-Order) products. *Automation in Construction*, 16(6), 759–770.

PARK, H., KIM, K., CHOI, S. and YOON, J. (2013): A patent intelligence system for strategic technology planning. *Expert Systems with Applications* 40(7): 2373–2390.

PRICKETT, P. and APARICIO, I. (2012): The development of a modified TRIZ Technical System ontology. *Computers in Industry* 63(3): 252–264.

RAHMANI, K. and THOMSON, V. (2012): Ontology based interface design and control methodology for collaborative product development. *Computer-Aided Design* 44(5): 432–444.

RANTANEN, K. and DOMB, E. (2007): Simplified TRIZ: New problem solving applications for engineers and manufacturing professionals. Ed. Auerbach Publications, CRC Press.

ROUSSELOT, F., ZANNI-MERK, C. and CAVALLUCCI, D. (2012): Towards a formal definition of contradiction in inventive design. *Computers in Industry* 63(3): 231–242.
SALAMATOV, Y. (1999): Triz: The Right Solution at the Right Time: A Guide to Innovative Problem Solving. Ed. Insytec BV.
SAVRANSKY, S. (2000): Engineering of Creativity. Introduction to TRIZ Methodology of Inventive Problem Solving. Ed. CRC Press.
SOO, V.W., LIN, S.Y., YANG, Y.S., LIN, S.N. and CHENG, S.L. (2006): A cooperative multi-agent platform for invention based on patent document analysis and ontology. *Expert Systems with Applications* 31(4): 766–775.
SPARQL available at http://www.w3.org/TR/rdf-sparql-query/
SUN, R. (1996): Commonsense reasoning with rules, cases, and connectionist models: A paradigmatic comparison. *Fuzzy Sets and Systems* 82(2): 187–200.
SUSHKOV, V., MARS, N. and WOGNUM, P. (1995): Introduction to TIPS: A theory for creative design. *Artificial Intelligence in Engineering* 9(3): 177–189
TERNINKO, J., ZUSMAN, A. and ZOTLIN, B. (1998): Systematic Innovation: An Introduction to TRIZ. Ed. St. Lucie Press.
VAIDYA, O. and KUMAR, S. (2006): Analytic hierarchy process: An overview of applications. *European Journal of Operational Research* 169(1): 1–29.
W3C (2009): OWL 2 Web Ontology Language, Document Overview. Available at http://www.w3.org/TR/owl2-overview/
WEIBEL, S., KUNZE, J., LAGOZE, C. and WOLF, M. (1998): Dublin Core Metadata for Resource Discovery. *The Internet Society* available at http://www.hjp.at/doc/rfc/rfc2413.html
WONGTHONGTHAM, P., CHANG, E., DILLON, T. and SOMMERVILLE, I. (2006): Ontology-based multi-site software development methodology and tools. *Journal of Systems Architecture* 52(11): 640–653.
YOON, J., LIM, J., CHOI, S., KIM, K. and KIM, C. (2011): Ontological functional modeling of technology for reusability. *Expert Systems with Applications* 38(8): 10484–10492.
YU, A. (2006): Methods in biomedical ontology. *Journal of Biomedical Informatics* 39(3): 252–266.
ZANNI-MERK, C., CAVALLUCCI, D. and ROUSSELOT, F. (2009): An ontological basis for computer aided innovation. *Computers in Industry* 60(8): 563–574.
ZANNI-MERK, C., CAVALLUCCI, D. and ROUSSELOT, F. (2011): Use of formal ontologies as a foundation for inventive design studies. *Computers in Industry* 62(3): 323–336.
ZHANG, X., MAO, X. and ABOURIZK, S.M. (2009): Developing a knowledge management system for improved value engineering practices in the construction industry. *Automation in Construction* 18(6): 777–789.

Biographical Notes

David M. Estrada-Contreras *received an engineering management masters degree from the Instituto Tecnologico de Orizaba. He has specialized in service design and software development. He has more than five years of experience managing technological processes and projects oriented to the supply chain management.*

David M.
Estrada-Contreras

Guillermo Cortés-Robles *obtained his MSc in industrial engineering from the Instituto Tecnologico de Orizaba and a PhD in industrial systems from the Institut National Polytechnique de Toulouse in France. He is today research professor at the Instituto Tecnologico de Orizaba. His research interests are the application of TRIZ theory combined with several approaches: knowledge management, decision support systems, creativity techniques, among other approaches with the aim of accelerating the innovation process. He is also the general secretary of the Mexican TRIZ Association (AMETRIZ).*

Guillermo
Cortés-Robles

Giner Alor-Hernández *is a full-time researcher in the Division of Research and Postgraduate Studies of the Instituto Tecnologico de Orizaba. He received a MSc and a PhD in computer science from the Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV), Mexico. He has headed 10 Mexican research projects granted by CONACYT, DGEST and PROMEP. He has been guest editor of three JCR-indexed journals. He is author/co-author of around 100 papers in computer science published in referred journals and conferences. His research interests include web services, e-commerce, Semantic Web, Web 2.0, service-oriented and event-driven architectures and enterprise application integration.*

Giner Alor-Hernández

Ulises Juárez-Martínez *is a full-time researcher in the Division of Research and Postgraduate Studies of the Instituto Tecnologico de Orizaba. He received a PhD in computer science from the Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV), Mexico. He has headed three Mexican research projects granted by different Mexican organizations.*

Ulises Juárez-Martínez

Alejandro Rodríguez-González *is an associate professor at the Department of Computer Languages and Systems and Software Engineering at the Technical University of Madrid. His main research interests are the semantic web, artificial intelligence and biomedical informatics, with a particular interest in the creation of medical diagnosis systems, medical knowledge representation and the extraction of knowledge from different sources (text, social media, etc.), and the analysis of social media impact in e-Health. He has a degree in computer science, an MSc in computer science and technology in the specialty of artificial intelligence, an MSc in engineering decision systems and a PhD in computer science. He has published several papers in international journals and edited various special issues in high quality journals such as the Journal of Medical Systems, Journal of Biomedical Informatics, Computational and Mathematical Methods in Medicine or International Journal of Software Engineering and Knowledge Engineering, among others. He has also participated in the organization of several workshops in top international conferences such as the International Semantic Web Conference and Extended Semantic Web Conference.*

Alejandro Rodríguez-González