

# Formal conceptualisation as a basis for a more procedural knowledge management

Javier Andrade , Juan Ares , Rafael García , Juan Pazos ,  
Santiago Rodríguez , Andrés Silva

*University of A Coruña. Campus de Elviña, s/n. 15071. A Coruña. Spain*  
*Technical University of Madrid. Campus de Montegancedo, s/n. 28660. Boadilla del Monte, Madrid. Spain*

## Abstract

Knowledge management at an organisational level can only be brought into practice if a corporate memory is defined. Unfortunately, at this moment there is no complete and procedural specification on how to build it.

This paper presents a complete and generic knowledge representation scheme that makes it possible to conceptualise/represent the knowledge of any domain in a systematic way, guiding the definition of a corporate memory and allowing us to reach a more procedural level in knowledge management discipline. The conclusions of our study, which follows the generic and formal definition of any conceptualisation, are illustrated by a real project.

*Keywords:* Corporate memory; Formal conceptualization; Knowledge management; Knowledge representation scheme

## 1. Introduction

The term knowledge management (KM) was coined in 1986 by Wiig and formalised in 2004 by Holsapple and Joshi in their Knowledge Management Ontology (KMO). According to Wiig, its main objectives, which are in harmony with the DKMC1 definition provided in KMO and other works are the following: (i) *to make the enterprise act as intelligently as possible to secure its viability and overall success*, and (ii) *to otherwise realize the best value of its knowledge assets*.

These objectives can only be reached by a KM programme that considers the three following key aspects: (i) the introduction of mechanisms that guarantee the total implication of the employees i.e., a company culture based on exchange and collaboration, (ii) the possibility for employees to exchange tacit knowledge, independently from their location or the moment of interaction and (iii) the definition of a corporate memory (CM) that supports the representation and interchange of the relevant corporate knowledge made explicit —cf. knowledge definition (DKMC6) in KMO. This paper focuses on the definition of this memory—or set of *knowledge artifacts* (DKR6) in KMO terminology—, which is the basis for the previous key aspects.

The CM has been initially approached through strategies from outside KM the most relevant being knowledge-based and document-based techniques. With regard to the former, ontologies were significantly exploited. It should nevertheless be noted that (i) a CM is different from a knowledge-based system and (ii) an ontology, although created to share and reuse knowledge, is only concerned with the static, not dynamic, knowledge of a particular domain. With regard to document-based corporate memories, even though paper-based or electronic documents can by themselves represent a CM, they are often considered as merely a first step in its implementation and very frequently those documents are not well-indexed or constitute a particular and reduced bibliography for each employee.

From the KM viewpoint, and taking into account how knowledge can be provided, a CM implies two different repositories (i) a corporate knowledge base, which directly provides the knowledge made explicit that it contains, and (ii) yellow pages, which allow accessing the relevant knowledge that is not made explicit in the previous repository by providing its location.

The first repository includes two elements: the knowledge itself and the lessons learned. The first element refers to the knowledge that is of a particular organisational environment or domain. With regard to the second element, several definitions have arisen since this was defined as guidelines, tips, or checklists of what went right or wrong in a particular event [38]. All these definitions however are centred on a common aspect: knowledge derived from (gained by) experience. This paper follows this approach, and the lessons learned refer therefore to the knowledge that each person possesses in the shape of experience. Hence, a lesson learned can be considered as knowledge about knowledge—meta-knowledge in an etymological sense—and, therefore, (a particular type of) knowledge.

The second repository involves the publication of the human (e.g., an expert) and non-human (e.g., a document) sources that have additional knowledge, i.e., relevant knowledge that is not made explicit in the previous repository. A knowledge management system (KMS) should not try to make explicit all the knowledge and lessons learned that exist in the organisation (corporate knowledge). Such a pretension is not viable due to the costs associated to this process and the huge amount of relevant knowledge that every organisation has, and this is why this repository is quite useful.

In recent years, many KM frameworks (e.g., the proposals of Wiig, and Liebowitz and Beckman) have

appeared for the deployment of KMSs in organisations

However, none of them meet the expectations of the KMSs developers they present a primarily prescriptive orientation, limiting themselves to indicating the steps that must be followed without describing how to approach each single step. A good example of the fact that these proposals neglect primordial procedural aspects is that they do not indicate how to develop a CM in a systematic—not *ad-hoc*—and detailed way, even though this is a widely established phase (cf. e.g., the *Explore knowledge and its adequacy* phase and the *Store* phase in the above-mentioned proposals).

The reason for this situation is that these frameworks focus on the study of the management process rather than on the object that is going to be managed i.e., the knowledge itself. Although only Wiig's proposal considers knowledge itself by means of a small set of descriptors, those descriptors are merely oriented towards describing certain knowledge characteristics (e.g., *stability* and *location*), not towards representing knowledge itself—which is the fundamental aspect in the definition (development) of a CM. This attempt recalls the work of disciplines such as Information Science on the subject of knowledge annotation—e.g., Resource Description Framework and Dublin Core Metadata Initiative—, where the focus is to merely label/annotate a knowledge asset.

This issue is of no relevance in the case of the yellow pages, because this repository does not consider knowledge about a domain, but sources that possess knowledge. In fact, we have defined in [7] a simple yellow pages system without a specific study about knowledge by means of the following descriptors: *name* of the source, its *location*, list of *topics* it knows, and *languages* in which it can provide the knowledge. The corporate knowledge base however is a repository that actually considers knowledge and knowledge—about knowledge (i.e., lessons learned). The main problem, which currently complicates the definition of a CM, is precisely to know how to represent the knowledge and/or the above-mentioned meta-knowledge about a domain. The present paper addresses this issue by providing a knowledge representation scheme and showing its application to constructing the corporate knowledge base in a KMS.

### 1.1. Conceptualisation in computing

Representing the knowledge about any domain obviously requires an appropriate previous understanding or conceptualisation of this knowledge .

For this reason, such conceptualisation should guide the definition of the corporate knowledge base for that domain. Logically, the used conceptualisation approach should be (i) complete (i.e., take into account all knowledge levels), (ii) independent of a particular domain (i.e., applicable to any domain), and (iii) not linked to the format in which knowledge can be found (e.g., a document).

Following this idea, all the conceptualisation (analysis) approaches from computer science or information systems development—where conceptualisation is used most extensively—could be initially considered. However, they are software-solution sensitive rather than domain sensitive: they take software development concerns into account and clearly reflect the behaviour of a software system within a given software development paradigm rather than an understanding of the domain.

Thus, the structured, object-oriented and agent-oriented analysis approaches conceptualise the domain at hand by means of implementation-oriented concepts, used to develop the software-solution, and not in terms of concepts proper to the domain. Although some approaches (e.g., TELOS and KAOS) have aimed for domain-sensitivity, the advances have not been effective as they do not rule out proximity to the implementation of the concepts used (e.g., the main concept in TELOS is class, bearing the same connotation as in object-orientation)

The most prominent approach in information systems development that now endeavours to explicitly account for domain-sensitivity in analysis is Soft Systems Methodology (SSM). SSM, however, generates conceptual models that are too limited (only activities and their logical dependencies) and that may be adequate for modelling the management process, but are not sufficient to model the knowledge itself. The same applies to business process modelling approaches, because all of them are only concerned with concepts directly linked to processes and their control flow (cf. tasks, inputs and outputs, etc.). This limitation implies that the rest of knowledge about a domain (i.e., tactical and declarative in our approach) that should be modelled is set aside in both cases.

The proximity to implementation concepts is also present in knowledge engineering. In fact, some authors consider the conceptual graph formalism to avoid this problem. They explicitly accept, however, that this formalism has important disadvantages for KM such as, for example, its rigidity (i.e., it is based on mathematical semantics) and the lack of a global view of the modelled domain.

This software-solution sensitivity is not adequate for KM because (i) we have to conceptualise and represent the knowledge of a given domain, not the behaviour of a software system for this domain in a given development paradigm, and (ii) the conceptualisation (and subsequently representation) should be easily understandable by the employees, who are not necessarily software specialists.

With regard to the first point, it is important to note that, in a software development project, the whole conceptualisation (and, therefore, consolidation—verification and validation) has to be redone if the paradigm is changed, although the knowledge of the considered domain remains unchanged.

With regard to the second point, if the conceptualisation is expressed using software concepts, logically a complex and costly special training is normally required. However, this training does not necessarily avoid the difficulty or even impossibility of understanding and consolidating the achieved conceptualisation. Note that in these cases, there is a use of concepts that are usually unfamiliar to the employees and not proper to the domain in which they work, which considerably complicates the already complex conceptualisation and consolidation tasks.

For all these reasons, we must find a way of conceptualising the knowledge about a domain that actually focuses on that domain, not on foreign concepts.

Section 2 presents a study that determines how to conceptualise the knowledge about any domain considering this issue. This study is based on the formal and generic definition of any conceptualisation. Section 3 shows a complete knowledge representation scheme, derived from the previous study, that allows the desired procedural level in defining a corporate knowledge base for any domain. Section 4 presents the process and the result of the application of this scheme to the development of a real corporate knowledge base, in which we have considered both the knowledge of the organisational environment and the associated lessons learned. Finally, Section 5 presents the key conclusions of this work.

## 2. Knowledge conceptualisation

In generic conceptualisation, concepts are the primary elements, because they are the most elementary and, therefore, essential units of any sort of human knowledge. Despite their evident importance, the nature of concepts generally is one of the toughest and oldest philosophical questions. However, Diez and Moulines established five hypotheses about concepts that can be used to quite effectively delimit what they are and how

they can be detected and used. These hypotheses of conceptualisation are the following

**H1.** Abstract entities. Concepts are, in principle, identifiable abstract entities to which human beings have access, providing knowledge and guidance about the real world.

**H2.** Contraposition of a system of concepts with the real world. Real objects can be identified and recognised thanks to, among other things, the available concepts. Several (real) objects are subsumed within one and the same (abstract) concept.

**H3.** Connection between a system of concepts and a system of language. The relationship of expression establishes a connection between concepts and words (expressions in general), and these (physical entities) can be used to identify concepts (abstract entities).

**H4.** Expression of concepts by non-syncategorematic terms. Practically all non-syncategorematic terms introduced by an expert in a domain express a concept.

**H5.** Need for set theory. For many purposes, the actual concepts should be substituted by the extensions of these concepts (sets of subsumed objects) to which set theory principles and operations can be applied.

If we define a concept as a mental structure that derives from the acquired knowledge, which applied to a domain clarifies it to the point of allowing its understanding, then conceptualisation would be defined as the use of concepts and relationships to deal with a domain

Indeed, a generic conceptualisation is formally defined as a triplet (Concepts, Relationships, Functions)

This triplet is an abstract representation of any conceptualisation, and includes: the presumed or hypothesised concepts, which exist in the domain—universe of discourse—; the relationships, in the formal (mathematical) sense, between concepts—relational basis set—; and, finally, the functions, also in the formal sense, defined on the concepts—functional basis set. Functions are a particular case of relationships, which, owing to their importance and specificity, are considered separately.

It would, however, be unpractical to articulate a conceptualisation (or a corporate knowledge base in a KMS) on the basis of the above three formal elements. There are three main reasons for this: (i) because concepts are abstract entities, as indicated in H1, they are difficult to access; (ii) relationships and functions are complex formal elements that are defined on the concepts, which means that, inherently, they further complicate the previous point; and (iii) people naturally express their

knowledge in natural language, not by means of concepts, relationships and functions, as indicated by H3.

Considering the preceding reasoning and H4, we propose the use of the conceptual elements resulting from the analysis of the linguistic categories of natural language. This procedure is based on the fact that there is a parallelism between natural language—the language *par excellence* for describing the knowledge about a domain—and the conceptual modelling and, equivalently, between the natural language and conceptual modelling structures

From the above-mentioned analysis, which is summarised below, we find that the identified conceptual elements can actually be matched to some of the three elements of the formal triplet ( $C, R, F$ ); that is, the generic and formal definition of a conceptualisation is not overlooked. However, ultimately, we can establish a knowledge taxonomy. This taxonomy, which comprises static and dynamic knowledge, is much more natural and practical from the viewpoint of developing a conceptualisation (or a corporate knowledge base in a KMS—cf. the explanation given in KMO related to knowledge definition (DKMC6) —), and can be used to conceptualise the knowledge about any domain without considering foreign concepts. Indeed, we have used this approach to build generic conceptual models in both knowledge engineering and software engineering and to structure the contents in e-learning systems with KM techniques which demonstrates its generality and closeness to the domain.

### 2.1. Linguistic categories considered

Based on H4, we have identified the conceptual elements that have to be considered by analysing the non-syncategorematic categories of nouns, adjectives and verbs. Moreover, we have attached importance to other linguistic categories that, although syncategorematic terms in many cases, were considered relevant because of their conceptual bearing: adverbs, locutions and other linguistic expressions.

For each category, we will indicate the elements to be considered when conceptualising the knowledge of any domain and, therefore, to which any knowledge representation scheme should attend. Finally, all these elements are classed in functional levels of knowledge to indicate their coherence, integration and correspondence with the description of the knowledge of any domain.

#### 2.1.1. Nouns

The most commonly used classification in this linguistic category establishes whether the noun is common or proper. Taking this into account, we notice a

parallelism between nouns and elements that are handled in any conceptualisation: common nouns can lead to concepts or properties and proper nouns can lead to particular objects or property values. We consider these four elements in the following subsections.

*2.1.1.1. Concepts.* This term, following the above-mentioned definition, is used here in the sense of anything real or otherwise that is relevant in the considered domain and about which something is to be expressed. In this way, relationships, actions and many other elements are actually concepts. However, only the concepts proper to and significant in the domain are considered here; the others will be addressed shortly hereafter. Thus, concept here means anything that is strictly speaking proper to the considered domain, which may refer to concrete or abstract, real or imaginary, or elementary or compound things. In this manner, the concepts are included within the universe of discourse of the triplet  $(C, R, F)$  and, considering H2, a concept subsumes a set of objects (the particular instances of this concept).

*2.1.1.2. Properties.* Properties are features of concepts and of relationships, because a relationship can be considered a compound concept (due to the concepts that it relates). In fact, this consideration means that it is possible to model relationships where a related element is, at the same time, a relationship. Relationships are conceptual elements that we consider later.

The set of all the properties of a concept/relationship describes the characteristics of the instances subsumed by this concept/relationship, and each of these can be considered as functions or relationships in the triplet  $(C, R, F)$  depending on how the domain is to be conceptualised. To appreciate this possibility, let us consider the example of the well-known Towers of Hanoi puzzle domain and its particular conceptualisation  $(\{D1, D2, D3, D4, D5\}, \{\text{blue, black, white}\}, \{\})$ .

In this conceptualisation there are one concept representing each disc and three unary relationships, each with respect to a different colour, and there are no functions. Note that, although the names of the concepts, relationships and functions are written, the conceptualisation involves the actual concepts, relationships and functions. This conceptualisation considers the property disc colour, but not the properties of these colours. If we want to consider the properties of the properties, the colours have to be viewed as concepts and a function, *colour*, has to be included to ascertain the colour of each disc:  $(\{D1, D2, D3, D4, D5, \text{blue, black, white}\}, \{\text{dark, light}\}, \{\text{colour}\})$ . As the colours

are now concepts in the universe of discourse, relationships can be added to indicate colour properties: unary relationships *dark* and *light* in the latter conceptualisation.

As shown by this example, a property can be considered within  $R$  or  $F$  in the formal triplet, depending on how the domain is conceptualised. Finally, it is worth mentioning that a property does not necessarily have only one value for one instance. That is, if the property were defined as a function, applying the property to an instance can result in a set of values.

*2.1.1.3. Particular objects.* A particular object is an occurrence of a concept or of a relationship for which the properties have specific values, and differs from other particular objects thanks to the value of such properties. Particular objects do not appear in conceptualisations, but in a level of instantiation of them

*2.1.1.4. Property values.* The value of a property, which is selected from a range of values (the set of possible values that can be assigned to the property in the domain), does not necessarily have to exist and, if it does, it does not necessarily have to be the unique one, as was previously discussed.

As shown in the above example of the Towers of Hanoi, if a property is conceptualised as a function, its values are within the universe of discourse. If it is conceptualised as a relationship, the actual property really disappears and unary relationships are considered for each value instead.

## *2.1.2. Adjectives*

There are two types in this linguistic category: determiners—used to determine the extent of the meaning of the noun—and modifiers—used to qualify the noun.

The adjectival determiners do not have to be considered in a conceptualisation, since they always accompany a noun and do not have any bearing altering the semantics of the noun phrase.

The adjectival modifiers can be descriptive or relational. The former refer to a property of the noun that they qualify. Accordingly, for example, the expression *big toy* presupposes the property size of the noun *toy*, whose assigned value is *big*. These adjectives are classed into different types according to their semantic trait—size, type, etc. Only the type-related classification can lead to a new interpretation to be conceptualised: the relationship of generalisation/specialisation, which is described in the next section. This is

the case of the expression *dynamic knowledge*, where *dynamic* indicates a particular type of *knowledge*. Finally, the relational adjectival modifiers allude to the scope or domain of the noun. In the expression *journalistic text*, for instance, the domain of *text* is *journalistic*. The same logic can therefore be applied to these adjectives.

Obviously, we can conceptualise *dynamic* or *journalistic* in the above examples as a property value. The way in which they are conceptualised depends on the relevance of the classification in the domain. However, the interesting point is that there is another possible interpretation to be conceptualised—the relationship of generalisation/specialisation—which needs to be considered.

### 2.1.3. Verbs

The linguistic theory that we use to analyse this category is the Case Grammar Theory which describes a natural language sentence in terms of a predicate and a series of arguments (cases). These cases (e.g., agent and object) are predefined by the verb in question.

We have selected this theory because it is a universal approach, irrespective of the language in question and provides a semantic description of the verbs, which is the essential aspect to be considered here. It is precisely the semantic relationship between the verb and its cases that is interpreted and modelled when conceptualising. This theory interprets the relationship between concepts: the verb alludes to the relationship and the nuclei of the phrases—cases—to the related concepts. This obviously is in accordance with the relational basis set of the  $(C, R, F)$  triplet.

The semantics and, therefore, the conceptualisation of the relationship differ depending on the type of verb that expresses the knowledge. Case grammar theories establish a verb classification depending on the cases that accompany the verb. Because this classification is not unique, as it depends on the semantic nuances and the cases considered, we use the classification by Martínez which places special emphasis on justifications based on natural language. Taking into account this classification, we have detected different types of relationships:

- ⌞ Generalisation/specialisation. It represents the semantics *is a type of* in natural language, indicating the inclusion of a concept in another more general one. In this type, what is true for a concept is also true for its subconcepts, although other properties or relationships for these subconcepts can appear.

- ⌞ Aggregation. It represents the semantics *part of* in natural language, making it possible to form a concept from other concepts of which the former is composed. An aggregation occurs when the whole has at least one emerging property that cannot be deduced from the properties of its parts.

- ⌞ Instantiation. It represents the semantics *x is a y* in natural language, and is the semantics that there is between a concept, or a relationship, and each of its instances at a level of instantiation.

- ⌞ Defined by the meaning of the verb (or domain defined). These relationships are particular to each domain and usually vary from one domain to another. A classification can therefore not be established as for the above relationships, where the meaning remains unchanged in any domain.

Because a conceptualisation (and a KMS) should represent knowledge as closely as possible and gather most of the semantics, it should consider the different types of relationships specified above. This distinction makes it possible to immediately assimilate all the underlying meanings for each type.

### 2.1.4. Other linguistic categories

We have not yet considered some linguistic structures that actually do introduce aspects to be conceptualised. These structures are adverbs, locutions, and other linguistic expressions that are really frequent in the description of any domain, which introduce the following conceptual elements: (i) constraints (e.g., no greater than, like minimum, etc.), (ii) inferences and calculations (e.g., deduce, if... then..., calculate, etc.), and (iii) sequence of actions (e.g., firstly, finally, after, etc.).

**2.1.4.1. Constraints.** A constraint can be defined as a predicate whose values are true or false for a set of elements. It can be seen as a function in the triplet  $(C, R, F)$  that constrains the possible values of these elements.

Constraints affect elements already identified: (i) constraints on properties, which demand that the properties have a value or a single value for each of the instances of the element that has the considered property, (ii) constraints on property values, which restrict the possible property values in the instances and (iii) constraints on relationships, which constrain the instances of a relationship.

**2.1.4.2. Inferences and calculations.** These elements allude to the manipulation of known facts to output new facts (e.g., property values used to obtain new property values). Inferences indicate what to conclude and what

to use for this purpose, and calculations indicate how to obtain an element using a mathematical or algorithmic expression. These elements can be therefore placed within the functional basis set in  $(C, R, F)$ .

*2.1.4.3. Sequence of actions.* This aspect indicates what actions are taken and when to consider/solve certain aspects of the domain. We should therefore consider the representation of two fundamental conceptual elements:

- ↳ Decomposition into steps. Human beings typically solve problems by breaking them down into steps. The last—non-decomposed—steps of the decomposition should indicate exactly what function they have and how they are carried out (inferences and calculations involved).
- ↳ Step execution order. The order in which the steps are taken is also essential, because, generally, it is not the same; for example, to take first one step and then another as the other way round.

The steps can be considered within the universe of discourse, and their decomposition and order as relationships within the relational basis set in the triplet  $(C, R, F)$ .

## *2.2. Knowledge taxonomy: functional knowledge levels*

Depending on the function that they fulfil in the domain, all the conceptual elements identified in the above sections can be grouped into two basic levels: (i) static, which constitutes the structural or descriptive domain knowledge—concepts, properties, relationships and constraints—, and (ii) dynamic, which constitutes the behaviour that takes place in the domain—inferences, calculations and step sequence. This last level can be divided into two sublevels: strategic sublevel, which includes what to do, when and in what order—decomposed and non-decomposed steps, and step sequence—and tactical sublevel, which specifies how to obtain new declarative knowledge—inferences and calculations.

These levels are interrelated as follows: the strategic level manages the tactical level, as it has to specify what inferences and calculations have to be made for each non-decomposed step; and the declarative level is managed by the other two levels, as it specifies what is used to decide on the alternatives of execution in the step sequence (bifurcation points) and on what basis the inferences and calculations are made.

With regard to the above-mentioned lessons learned, they are knowledge about knowledge (i.e., meta-

knowledge) and, therefore, knowledge. By this reason, the proposed approach is also applicable to lessons learned. In other words, a lesson learned can be conceptualised (i.e., represented) on the basis of the three knowledge levels presented here, as can be seen in the last section.

## **3. Knowledge representation scheme**

In order to facilitate the knowledge representation based on the functional levels identified in the previous section, we have segmented the representation scheme in as many subschemes as identified levels. Thus, there is a static subscheme—which consists of the declarative level—and a dynamic subscheme—which consists of the strategic and the tactical levels.

Each of these subschemes consists of the descriptors that are presented hereafter. Although we have identified three different subschemes, we also established a set of common descriptors that do not depend on the considered functional level: they label/annotate (cf. Section 1) any knowledge asset. These descriptors do not represent the knowledge itself, but describe certain characteristics of the knowledge.

Finally, it should be noted that the contents of the presented descriptors are assigned by the knowledge management team or by the individual workers (producing agents), depending on whether the knowledge incorporation is active or passive

### *3.1. Common descriptors*

As can be seen in Table 1, the common descriptors are divided into terminological, qualificatory, and relational descriptors. The first group allows us to clarify the nomenclature that is used to reference a knowledge asset in the organisation. The second group characterises the knowledge asset from the point of view of its utility. Finally, the third group allows us to go into the details of the knowledge asset, if such were necessary, either through an additional knowledge source (yellow pages) or through an existing link between the knowledge and the associated lessons learned.

### *3.2. Strategic subscheme*

The strategic subscheme allows us to represent the step sequence that is necessary for a certain organisational function: it allows us to define what to do, when, and in what order.

Because a KMS should provide knowledge to various access profiles (coordinators and executors),

Table 1  
Common descriptors

Type	Descriptor	Description
Terminological	Name	The main term by which the asset is known in the organisation
	Synonyms	Other terms by which the asset may be known
	Abbreviations	Abbreviations that are used for the asset
	Observations	Any type of clarification concerning the terminology used in the KMS
Qualificatory	Topic	Theme to which the asset refers
	Reliability	Level of trust that is given to the asset
	Impact	Relevance of the asset in the domain of interest
	Validity	Time frame in which the asset is considered valid
	Availability	Time frame in which the asset is available
	Security	Organisational roles that have access to the asset
	Language	Language of the asset
Relational	Support sources	Sources that have additional knowledge. This descriptor allows the integration of a yellow pages system. A more detailed description of a yellow pages system can be found in
	Associate knowledge/ lessons learned	If we are considering a lesson learned, this descriptor indicates the knowledge that is being refined. If the asset is a “basic” knowledge asset, this descriptor indicates the related lessons learned in the KMS
	Producing agent	Agent that provides the knowledge asset. This descriptor allows us to put into operation the mechanisms that recognise and guarantee the knowledge exchange (i.e., a culture of exchange and collaboration)

we consider the complete decomposition of a function into its constitutive steps, considering both the non-terminal (decomposed) steps and the steps of the last level (non-decomposed). The former provide the possibility to supervise parts of the function, whereas the latter guide the executor of the step. The descriptors of both the decomposed and non-decomposed steps are shown in Table 2.

### 3.3. Tactical subscheme

This subscheme allows us to represent the knowledge that indicates the way in which a certain task should be carried out, and involves the descriptors presented in Table 3. This knowledge is connected to the non-decomposed steps—the task it is associated with—that are defined in the strategic subscheme.

Table 2  
Descriptors for the strategic subscheme

Type	Descriptor	Description
Decomposed steps	Description	Definition/description of the strategic step in terms of the coordinator
	Decomposition structure	Description of the decomposition associated to this step, considering the substeps into which it is decomposed
	Execution structure	Description of the execution order of the substeps into which the step is decomposed
	Preconditions	Previous requisites that are necessary to carry out the step
	Postconditions	Output requisites that are necessary to finalise the step
	Inputs	Elements that are necessary for the execution of the step
	Outputs	Elements that are generated as a consequence of the step
	Control elements	Elements implied in the bifurcation points of the <i>Execution structure</i> (cf. interrelationships between knowledge levels in Section 2.2)
	Coordinator	The role that is responsible for the coordination of the step
	Observations	Any related observation (e.g. the justification of the need of the step)
Non-decomposed steps	Description	Definition/description of the step in terms of the executor
	Preconditions	Necessary previous requisites
	Postconditions	Requisites to finalise the step
	Inputs	Elements that are necessary to execute the step (determined by the tactical knowledge of the descriptor <i>Operational mode</i> )
	Outputs	Elements that are generated by the execution of the step (determined by the tactical knowledge of the descriptor <i>Operational mode</i> )
	Operational mode	Reference to the tactical knowledge that obtains the <i>Outputs</i> from the <i>Inputs</i> (cf. interrelationships between knowledge levels in Section 2.2)
	Executor	Entrepreneurial role charged with the execution of the step
	Observations	Any related observation (e.g. the final purpose of the step)



Table 3  
Descriptors for the tactical subscheme

Descriptor	Description
Description	Definition/description of the considered tactical knowledge
Basic elements	Elements—concepts, relationships and properties—that are required to proceed with the <i>Definition</i> (cf. interrelationships between knowledge levels in Section 2.2)
Conclusion elements	Elements—concepts, relationships and properties—that are obtained by the <i>Definition</i> (cf. interrelationships between knowledge levels in Section 2.2)
Definition	Algorithm, mathematical expression (cf. H5 in Section 2), inference or procedure that describes the considered tactical knowledge
Observations	Any related observation (e.g., limitations in the application of the <i>Definition</i> )

### 3.4. Declarative subscheme

Table 4 presents the descriptors defined to allow the representation of the conceptual elements considered under the declarative level: concepts, relationships, and properties. The restrictions have been incorporated in the group of the element they affect (i.e., in the properties and the relationships).

### 3.5. Overview of the knowledge representation scheme

Table 5 summarises the proposed knowledge representation scheme, reflecting in which subscheme the conceptual elements identified in Section 2 are considered. Each subscheme comprises the descriptors presented in its respective section, and these were defined through the following steps:

- ↳ Consideration of all the conceptual elements that were identified through the study of the natural language (e.g., the descriptor *constraints* for relationships is derived from the analysis of adverbs, locutions, and other linguistic expressions). Therefore, this step allows us to identify the descriptors oriented towards conceptualising/representing the knowledge itself.
- ↳ Inclusion of those descriptors (derived specifically from the KM domain) oriented towards labelling/annotating the conceptualised/represented knowledge, and describing as such some of its characteristics. Some of these new descriptors affect only one conceptual element (e.g., the descriptor *coordinator* for the conceptual element decomposed steps) and some others (the common descriptors) affect all the knowledge representation subschemes (e.g., the descriptor *language*). These new descriptors are

established mainly by considering the previously mentioned Wiig’s proposal [43]. This step has therefore allowed us to identify the descriptors oriented towards labelling/annotating the knowledge, complementing as such the previous step.

We do not claim that our set of descriptors is the unique possible solution. Other sets are obviously possible, although they must comprise, in one form or another, all the aspects that we have considered. For example, other authors could follow Wiig’s proposal, where our two descriptors *support sources* and *language* are joined in a unique descriptor called *form*.

## 4. Application of the representation scheme

This section presents the pilot project that was carried out in COFAGA ([www.cofaga.org](http://www.cofaga.org)), a Spanish pharmaceuticals distribution company to which the proposed representation scheme was applied so as to construct the

Table 4  
Descriptors for the declarative subscheme

Type	Descriptor	Description
Concepts	Description	Definition/description of the concept
	Properties	Characteristics that describe the concept
	Relationships	Associations maintained with other concepts
Relationships	Observations	Any related observation (e.g., the justification of the concept)
	Description	Definition/description of the identified relationship
	Elements	Concepts that take part in this relationship
	Type	The type of the relationship (see Subsection Verbs in last section)
	Properties	The characteristics that describe the relationship
	Constraints	Restrictions that are applicable to the relationship
	Observations	Any related observation (e.g., the relevance of the relationship)
Properties	Description	Definition/description of the considered property
	Element	Concept or relationship to which the property is linked
	Data category	Type of values of the property and, if necessary, the measure and precision units that are required
	Range	Allowable values taken by the property
	Constraints	Limitations of the property (e.g., compulsoriness or optionality)
	Source	The source that provides the values assigned to the property
	Observations	Any related observation (e.g., a value by default)

Table 5  
Summary of the suggested scheme

Knowledge representation scheme			Conceptual elements identified in Section 2			
Dynamic subscheme	Strategic subscheme (Table 2)	Non-decomposed steps Decomposed steps	Non-decomposed steps Decomposed steps Step execution order	Decomposition into steps	Strategic level	Dynamic level
	Tactical subscheme (Table 3)		Inferences and calculations		Tactical level	
Static subscheme	Declarative subscheme (Table 4)	Concepts Relationships Properties	Concepts Relationships and their constraints Properties and their values and constraints		Declarative level	Static level
	Common descriptors (Table 1)	Terminological Qualificatory Relational	These descriptors do not depend on the considered functional knowledge level, and describe certain characteristics of the knowledge			

corporate knowledge base in a KMS. We start by briefly describing the company and more specifically those aspects that influence the organisational area of interest: orders to suppliers. We then present the process of conceptualising and representing the knowledge associated with this area, the result of this process, and an example of a lesson learned.

#### 4.1. Description of the business environment

The pharmaceuticals distribution sector is characterised by the need (i) to deliver all the products that are ordered by the clients (pharmacies) and (ii) to do this in a lapse of time typically between 1 and 5 h. The second condition has made the companies of this sector adopt a distributed organisational structure. COFAGA consists of three separate warehouses at a distance of approximately 100 km, each of which covers a specific geographical area. These warehouses operate independently from each other, handling their own clients, suppliers and products; i.e., they operate as *knowledge islands*. Their only point of cooperation is the control of narcotics and psychotropic drugs, which require a strict control under Spanish legislation. This requirement, and the fact that the mentioned products exist only in small quantities, have led the company to centralise their management in one single warehouse and distribute them internally through inter-warehouse distribution routes. In this case, the warehouses simply receive the order from their clients, redirect it to the warehouse in charge and distribute the products when they arrive.

Due to the fact that each warehouse must be able to deliver any product to the clients of its geographical area of influence, it inevitably manages a large number of suppliers and products (more than 800 suppliers and approximately 25,000 products). Also, each warehouse is obliged to maintain an adequate stock for each pro-

duct, because it is bound by the previously mentioned response time. As a matter of fact, each warehouse divides its suppliers into those that are contacted for an order (*fired* in the pharmaceuticals distribution jargon) on a weekly, fortnightly or monthly basis, depending on endogenous factors—sales volume for the geographical area of influence—and exogenous factors—the response time of the suppliers, their location or costs. On average, 90% of the suppliers are *fired* weekly, 8% fortnightly and 2% monthly, which implies that each warehouse places an average of 40,000 orders each year.

On the one hand, it is important to note that not all the products undergo the same sales behaviour. The rotation of the products varies between *very high* (hundreds of thousands of sales each year) and *very low* (0, 1 or 2 units each year). In fact, approximately 20% of the products cover 80% of the sales. On the other hand, most products, even those with a high rotation level, are season bound and their sales vary significantly according to the time of the year (e.g., antihistamines in the spring and sun creams in the summer). All these aspects are taken into account to determine the stock for each product.

Finally, we note that in this particular business environment (i) the suppliers very often impose minimal orders for certain products, which forces the buyer to pay for a minimal number of units and, or, sets of units, and (ii) the suppliers usually offer discounts and special tariffs according to the order volume. These aspects obviously influence the restocking decisions of each warehouse and, as such, the placement of their orders.

#### 4.2. Representation of the corporate knowledge

The pilot project was carried out by a working group of two persons with notions on KM and on the presented representation scheme, and three persons from the Purchases Department (the Head of Purchases and two

staff members), all belonging to the warehouse with the largest sales and purchases volumes.

The project focused on a single warehouse because (i) the restocking process, according to the company's High Direction, is very similar for all three warehouses, and (ii) the pilot project should interfere as little as possible with the daily business routine. Nevertheless, even though the process focused on the operations and staff of one warehouse, the other warehouses participated in the consolidation and final approval of the established knowledge representation.

The project started with an introduction and training session of 2 h for all the personnel involved in the warehouses' restocking process. The purpose of this session

was to explain the project, its aims, its phases and techniques, and to promote the interest of the personnel. It was followed by an acquisition–conceptualisation process in order to represent the corporate knowledge according to the proposed scheme.

This process was based on the Task Environment Analysis and modelling (TEA) technique and complemented with the Verbal Protocol Analysis (VPA) technique. As stipulated by the TEA technique, a first level of interviews was to give a general vision of the restocking task. Each of the three selected members of the Purchases Department was interviewed during approximately 1 h, which allowed us to conceptualise and represent the decomposition structure of

Table 6  
Decomposed strategic step *Restocking*

Description	This step determines the restocking orders that will be placed to cover the current restocking needs
Decomposition structure	
Execution structure	
Preconditions	The Head of Purchases has approved the planning that indicates when the orders must be placed with each supplier (Restocking planning)
Postconditions	The purchases personnel has sent all the orders with Situation = "Waiting to be sent"
Inputs	Product, Supplier, Order, Order line, Be supplied(Product, Supplier), Have(Order, Order line), Be ordered(Product, Order line), Be sent(Order, Supplier)
Outputs	Order, Order line, Have(Order, Order line), Be ordered(Product, Order line), Be sent(Order, Supplier)
Control elements	Product
Coordinator	Head of Purchases
Observations	Initially, the information system quantifies the units to be ordered according to the needs of the products (i) whose supplier is planned for consideration and (ii) whose current stock is inferior to the minimum. In this process, the system already considers the orders with Situation = "Waiting for reception" (already sent but not yet received). Finally, the result will be adjusted/supervised by the purchases personnel

the *Restocking management*. This task was decomposed into two main steps: *Restocking planning* and *Restocking*. The *Restocking* step consists of the restocking that follows the established planning, and the restocking that results from the minimum level reached by the stock of certain products. Both restockings are quantified automatically and subsequently adjusted by hand before placing the order. The rest of this section focuses on the knowledge that is related to the *Restocking* step.

In order to avoid significant interferences with the daily routine, the project continued with the VPA technique, which was applied 1 h a day for a week. This technique allowed us to progressively acquire and understand (conceptualise) more knowledge, and synthesise (represent) it according to the proposed scheme. The result of this synthesis was consolidated (verified and validated) as much as possible by means of the mind maps technique. The detected inconsistencies, misunderstandings and questions were often resolved in the next application of the VPA technique by acquiring and understanding more knowledge. The problems that remained unresolved were directly tackled in the second level of interviews of the TEA technique, which lasted approximately 1 h for each member of the Purchases Department. We thus follow the iterative acquisition–conceptualisation cycle that is proper to any modelling process.

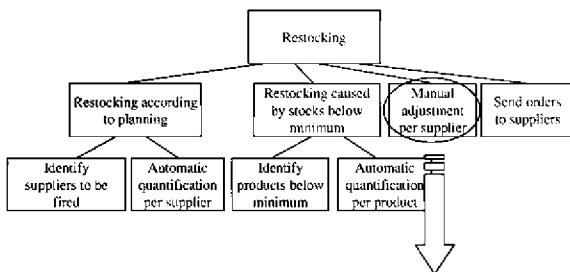
Finally, the obtained knowledge representation was revised and approved by the purchases personnel of the other warehouses. The mind maps technique that was applied during this revision only suggested small changes, which were mostly related to the level of detail provided by the below-mentioned tactical knowledge *Manual restocking adjustment*.

As an example, Table 6 presents the final revised and approved knowledge representation that corresponds to the aforementioned decomposed strategic step *Restocking*. It can be seen that all the knowledge is synthesised on the basis of the descriptors that were defined in the previous section.

With respect to the non-decomposed strategic step *Manual adjustment per supplier*, Fig. 1 shows the content of the descriptors of this strategic knowledge asset and of the associated tactical knowledge.

Fig. 2 shows the representation of the following elements: the concept *Product*—one of the most relevant concepts of the domain, its property *Minimal quantity that must be ordered*, and the relationship of this concept with the concept *Supplier*.

The representation scheme proposes, as it should, a generic and exhaustive set of descriptors. These characteristics nevertheless imply the necessary adaptation of the scheme to each particular situation. In this specific project we have adapted the terminological, qualifying,



**Non-decomposed strategic step *Manual adjustment per supplier***

Description	The purchases personnel adjust the restocking suggested by the information system in order to optimise the restocking process and the associate costs
Preconditions	The information system has quantified the orders to cover the current restocking needs
Postconditions	All the orders with the property Situation="Waiting to be adjusted" were adjusted manually by the purchases personnel
Inputs	Product, Supplier, Order (with Situation="Waiting to be adjusted"), Order line, Have(Order, Order line), Be ordered(Product, Order line), Be sent(Order, Supplier)
Outputs	Order (with Situation="Waiting to be sent"), Order line, Have(Order, Order line), Be ordered(Product, Order line), Be sent(Order, Supplier)
Operational mode	Manual restocking adjustment
Executor	Purchases personnel
Observations	This step serves (i) to adapt the orders to the changes in the sales trends, and (ii) to avoid additional costs for not considering the benefits of the tariffication sections or the discounts per purchase volume offered by suppliers

**Tactical knowledge *Manual restocking adjustment***

Description	This procedure manually adjusts the orders that are generated by the information system before sending them to the supplier
Basic elements	Product, Supplier, Order (with Situation="Waiting to be adjusted"), Order line, Have(Order, Order line), Be ordered(Product, Order line), Be sent(Order, Supplier)
Conclusion elements	Order (with Situation="Waiting to be sent"), Order line, Have(Order, Order line), Be ordered(Product, Order line), Be sent(Order, Supplier)
Definition	For each order with property Situation="Waiting to be adjusted": 1. If the order was generated because the stock is below minimum: a. Decide if we cancel the order according to the future (planned) firing date of that supplier and to the real need of the products b. Consider the possibility of suggesting the modification of the restocking strategy to the Head of Purchases (firing period for that supplier and/or the values of the properties of Maximal stock and Minimal stock of the ordered products) 2. For each product of the considered order: a. If we notice a significant change in the sales trends or in the context (e.g., advancing of the summer season with the ensuing impact on the sun creams sales), consider the adjustment of the quantity to be ordered b. If there exist any special offers for this product, consider the adjustment of the quantity to be ordered c. If there exists a beneficial tariffication section, consider the adjustment of the ordered quantity to that section 3. If there exist any special offers per purchase volume offered by that supplier, consider the adjustment of the global order 4. If there exists a significant change in the sales trends of that supplier's products, consider the possibility of suggesting the modification of the restocking strategy to the Head of Purchases (firing period for that supplier and/or the values of the properties of Maximal stock and Minimal stock of the ordered products) 5. Establish the property Situation of the order="Waiting to be sent"
Observations	The recommendations to the Head of Purchases concerning the firing period of a supplier should not originate from punctual changes in the sales trends

Fig. 1. "Navigation" through the knowledge *Manual adjustment per supplier*.

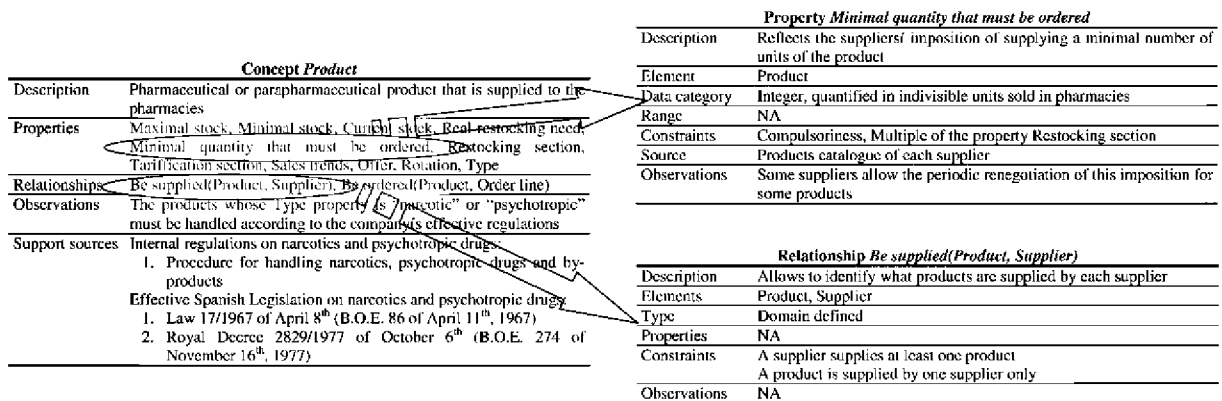


Fig. 2. "Navigation" through the concept *Product*.

and relational descriptors; in order to avoid unnecessary detail in the process and in the result, we have used only those descriptors that were required at a given moment. For example, the descriptor *Support sources*, which is annexed to the concept *Product* in Fig. 2, was used to indicate the sources that contain the directives concerning the storage and distribution of products classified as narcotics and psychotropic drugs.

#### 4.3. Representation of lessons learned

Whereas the previous section describes the knowledge of the organisational environment, this section

focuses on the representation of the lessons learned that were incorporated into the KMS in the course of the pilot project.

The incorporation process started at the company's already existing *Suggestions box*, which was opened for proposals, comments, and contributions related to the *Restocking management*. The established work team filtered the received suggestions. Those that seemed interesting underwent the same acquisition–conceptualisation process as the one described above, taking into account the person(s) who put them forward. If the suggestion was eventually considered interesting, it was represented through the proposed scheme and

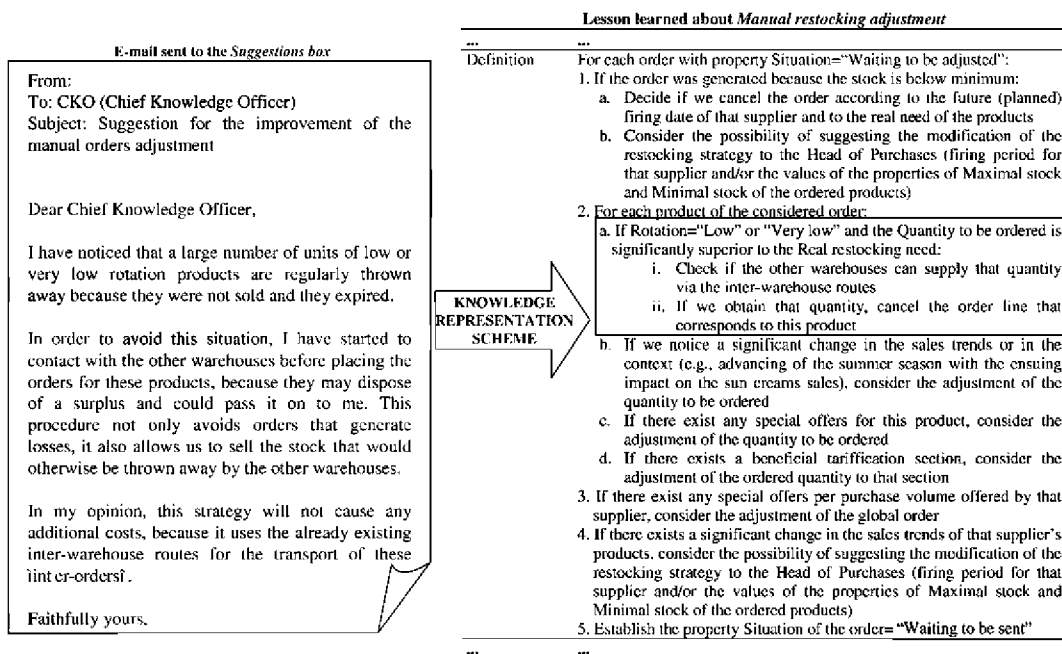


Fig. 3. Representation of a lesson learned.

incorporated into the CM as a lesson learned. This lesson was then communicated to all the personnel involved in the process.

Fig. 3 shows one of the proposals that were sent to the *Suggestions box*, the lesson learned derived from it, and its impact on the already represented knowledge. This suggestion specifically refers to the previously presented tactical knowledge. The descriptor *Associate knowledge/Lessons learned* was incorporated into the lesson learned representation so as to link it to the knowledge it refines. Moreover, the descriptor *Producing agent* was incorporated to carry out the reward policy based on the contributions to the KMS. Similarly, the tactical knowledge *Manual restocking adjustment* was associated with this lesson through its descriptor *Associate knowledge/Lessons learned*.

Although the above lesson learned is one of the simplest lessons that surged from the *Suggestions box*, it has generated important economic and competitive advantages: reduction of the standstill of products with a *low* or *very low* rotation—with the ensuing financing of purchases, optimisation of the storage capacity—with the ensuing optimisation of the stock and of handled products, and improvement of the service to the clients—with the ensuing increase in market share. It is for this reason that, at present, the company has adopted this lesson learned as knowledge: from meta-knowledge, it has evolved into an actual *modus operandi*. The homogeneity of the proposed scheme (Section 2.2) in treating knowledge itself and lessons learned allows such a direct change.

## 5. Conclusions

Current KM frameworks are primarily prescriptive. However, KM practice and KMS development necessarily require approaches that are also procedural, allowing us to detail how the prescribed activities must be carried out.

This paper has presented a knowledge representation scheme that facilitates the required procedural detail by directly approaching the object that is to be managed: knowledge. Starting from the generic and formal definition of any conceptualisation, we have obtained this representation scheme by considering the knowledge types (functional taxonomy) and its defining characteristics (descriptors).

We advocate this scheme to articulate the management process established by the current KM frameworks in a procedural way. This paper has focused on the definition of the most relevant element of a KMS: the CM and, specifically, the corporate knowledge base repository. The application of the defined scheme has

allowed us to reach a generic, complete, and systematic definition of such a repository, illustrated with a real project. This situation avoids the current *ad-hoc* practice by the developers of KMSs.

Finally, the presented scheme is not only useful for the definition of a CM, it also serves to describe and direct the activities of a KM framework. In fact, this scheme has allowed us to articulate the most relevant phases of a procedural KM methodological framework, whose first version is described, applied, and evaluated.

The phases that most benefit from this scheme are *knowledge acquisition*, *knowledge assimilation* (conceptualisation and representation) and *knowledge consolidation*, as can be concluded from the presented example. Interestingly, these phases are the ones that underlie most of the current KM frameworks,

## Acknowledgements

We would like to thank COFAGA, for accepting our proposal; La Caixa, for funding project P021005056; the Dirección Xeral de Investigación e Desenvolvemento da Xunta de Galicia (Autonomous Government of Galicia), for funding project PGIDIT03PXIA10501PR; the University of A Coruña, for its economical support; and Salomé García, for acting as intermediary between the two Universities. Our thanks also go to Valérie Bruynseraede for her help in translating this paper.

## References

- A. Abecker, A. Bernardi, K. Hinkelmann, O. Kuhn, M. Sintek, Towards a well-founded technology for organisational memories, in: B. Gaines (Ed.), Proceedings of AAAI Spring Symposium on Artificial Intelligence in Knowledge Management, AAAI, Stanford, 1997.
- J. Andrade, J. Ares, R. García, J. Pazos, S. Rodríguez, A. Silva, A methodological framework for viewpoint-oriented conceptual modeling, *IEEE Transactions on Software Engineering* 30 (5) (2004).
- J. Andrade, J. Ares, R. García, J. Pazos, S. Rodríguez, A. Silva, A methodological framework for generic conceptualisation: problem-sensitivity in software engineering, *Information and Software Technology* 46 (10) (2004).
- J. Andrade, J. Ares, R. García, J. Pazos, S. Rodríguez, A. Silva, Definition of a problem-sensitive conceptual modelling language: foundations and application to software engineering, *Information and Software Technology* 48 (7) (2006).
- J. Andrade, J. Ares, R. García, S. Rodríguez, Conceptual modeling in requirements engineering: weaknesses and alternatives, in: J.L. Maté, A. Silva (Eds.), *Requirements Engineering for Socio-technical Systems*, Information Science Publishing, Hershey, 2005.
- J. Andrade, J. Ares, R. García, S. Rodríguez, M. Seoane, S. Suárez, The knowledge management as an e-learning tool, in: C.

- Ghaoui (Ed.), *Encyclopaedia of Human Computer Interaction*, Idea Group, Inc., Hershey, 2005.
- J. Andrade, J. Ares, R. García, S. Rodríguez, S. Suárez, Lessons learned for the knowledge management systems development, in: W.W. Smari (Ed.), *Proceedings of the 2003 IEEE International Conference on Information Reuse and Integration*, IEEE SMC, Las Vegas, 2003.
- J. Ares, J. Pazos, Conceptual modelling: an essential pillar for quality software development, *Knowledge-Based Systems* 11 (2) (1998).
- M. Boman, Jr.J.A. Bubenko, P. Johannesson, B. Wangler, *Conceptual Modelling*, Prentice-Hall Series in Computer Science, London, 1997.
- B.G. Buchanan, D. Barstow, R. Bechtel, J. Bennett, W. Clancey, C. Kulikowski, T. Mitchell, D.A. Waterman, Constructing an expert system, in: F. Hayes-Roth, D.A. Waterman, D.B. Lenat (Eds.), *Building Expert Systems*, Addison-Wesley, Reading, 1983.
- T. Buzan, B. Buzan, *The Mind Map Book: How to Use Radiant Thinking to Maximize Your Brain's Untapped Potential*, Plume, New York, 1996.
- P.B. Checkland, *Systems Thinking, Systems Practice*, John Wiley and Sons, London, 1981.
- W.A. Cook, *Case Grammar Theory*, Georgetown University Press, Washington D.C., 1989.
- T.H. Davenport, L. Prusak, *Working Knowledge: How Organizations Manage What They Know*, Harvard Business School Press, Boston, 2000.
- R. Dieng, O. Corby, A. Giboin, M. Ribi re, Methods and tools for corporate knowledge management, *International Journal of Human-Computer Studies* 51 (3) (1999).
- J.A. Diez, C.U. Moulines, *Fundamentos de Filosof a de la Ciencia*, Ariel S.A., Barcelona, 1997.
- M. Genesereth, N. Nilsson, *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann Publishers Inc., California, 1986.
- R.L. Glass, I. Vessey, V. Ramesh, Research in software engineering: an analysis of the literature, *Information and Software Technology* 44 (8) (2002).
- A. G mez-P rez, M. Fern ndez-L pez, O. Corcho, *Ontological Engineering with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*, Springer-Verlag, London, 2004.
- C.W. Holsapple, K.D. Joshi, A formal knowledge management ontology: conduct, activities, resources, and influences, *Journal of the American Society for Information Science and Technology* 55 (7) (2004).
- C.W. Holsapple, Knowledge management support of decision making, *Decision Support Systems* 31 (1) (2001).
- J. Hoppenbrouwers, B. van der Vos, S. Hoppenbrouwers, NL structures and conceptual modelling: grammalizing for KISS, *Data & Knowledge Engineering* 23 (1) (1997).
- O. K hn, A. Abecker, Corporate memories for knowledge management in industrial practice: prospects and challenges, *Journal of Universal Computer Science* 3 (8) (1997).
- J. Liebowitz, *Knowledge Management Handbook*, CRC Press, Florida, 1999.
- D. Liu, I. Wu, Collaborative relevance assessment for task-based knowledge support, *Decision Support Systems* 44 (2) (2008).
- P. Mart nez, Una Propuesta de Estructuraci n del Conocimiento para el An lisis de Textos: una Aplicaci n a la Adquisici n de Esquemas Conceptuales de Bases de Datos, Ph.D. Thesis, Department of Artificial Intelligence, Technical University of Madrid (1998).
- J. Mendling, G. Neumann, M. N ttgens, A comparison of XML interchange formats for business process modeling, in: F. Feltz, A. Oberweis, B. Otjacques (Eds.), *Proceedings of EMISA Information Systems in E-Business and E-Government*, GI, Bonn, 2004.
- G.A. Miller, WordNet: a lexical database for English, *Communications of the ACM* 38 (11) (1995).
- G.W. Mineau, R. Missaoui, R. Godinx, Conceptual modeling for data and knowledge management, *Data & Knowledge Engineering* 33 (2) (2000).
- D. Nevo, Y.E. Chan, A temporal approach to expectations and desires from knowledge management systems, *Decision Support Systems* 44 (1) (2007).
- B. Nuseibeh, S. Easterbrook, Requirements engineering: a roadmap, in: A. Finkelstein (Ed.), *The Future of Software Engineering*, ACM Press, New York, 2000.
- S. Rodr guez, Un Marco Metodol gico para la gesti n del Conocimiento y su Aplicaci n a la Ingenier a de Requisitos Orientada a Perspectivas, Ph.D. Thesis, Department of Information and Communications Technologies, University of A Coru a (2002).
- B. Rubenstein-Montano, J. Liebowitz, J. Buchwalter, D. McCaw, B. Newman, K. Rebeck, A systems thinking framework for knowledge management, *Decision Support Systems* 31 (1) (2001).
- G. Simon, Knowledge acquisition and modeling for corporate memory: lessons learnt from experience, in: B. Gaines, M. Musen (Eds.), *Proceedings of Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop*, AAAI, Banff, 1996.
- M. Uschold, M. Gr ninger, Ontologies: principles, methods and applications, *Knowledge Engineering Review* 11 (2) (1996).
- G. van Heijst, R. van der Spek, E. Kruijzinga, Corporate memories as a tool for knowledge management, *Expert Systems with Applications* 13 (1) (1997).
- Y. Wand, D.E. Monarchi, J. Parsons, C.C. Woo, Theoretical foundations for conceptual modelling in information systems development, *Decision Support Systems* 15 (4) (1995).
- R. Weber, D.W. Aha, I. Becerra-Fern ndez, Intelligent lessons learned systems, *Expert Systems with Applications* 20 (1) (2001).
- B. Wielinga, J. Sandberg, G. Schreiber, Methods and techniques for knowledge management: what has knowledge engineering to offer? *Expert Systems with Applications* 13 (1) (1997).
- K. Wiig, Knowledge management foundations: thinking about thinking, *How People and Organizations Create, Represent and Use Knowledge*, Schema Press, Texas, 1993.
- K. Wiig, *Knowledge Management Methods: Practical Approaches to Managing Knowledge*, Schema Press, Texas, 1995.
- K. Wiig, Knowledge management: where did it come from and where will it go? *Expert Systems with Applications* 13 (1) (1997).
- K. Wiig, R. de Hoog, R. van der Spek, Supporting knowledge management: a selection of methods and techniques, *Expert Systems with Applications* 13 (1) (1997).