A FUNCTIONAL APPROACH TO REALIZING DECISION SUPPORT SYSTEMS IN TECHNICAL REGULATION MANAGEMENT FOR DESIGN AND CONSTRUCTION

G.L. LUCARDIE
TNO Building and Construction Research
P.O. Box 49
2600 AA Delft
The Netherlands

ABSTRACT. Technical building standards defining the quality of buildings, building products, building materials and building processes aim to provide acceptable levels of safety, health, usefulness and energy consumption. However, the logical consistency between these goals and the set of regulations produced to achieve them is often hard to identify. Not only the large quantities of highly complex and frequently changing building regulations to be met, but also the variety of user demands and the steadily increasing technical information on (new) materials, products and buildings have produced a very complex set of knowledge and data that should be taken into account when handling technical building regulations. Integrating knowledge technology and database technology is an important step towards managing the complexity of technical regulations. Generally, two strategies can be followed to integrate knowledge and database technology. The main emphasis of the first strategy is on transferring data structures and processing techniques from one field of research to another. The second approach is concerned exclusively with the semantic structure of what is contained in the data-based or knowledge-based system. The aim of this paper is to show that the second or knowledge-level approach, in particular the theory of functional classifications, is more fundamental and more fruitful. It permits a goal-directed rationalization strategy towards analysis, use and application of regulations. Therefore, it enables the reconstruction of (deep) models of regulations, objects and of users accounting for the flexibility and dynamics that are responsible for the complexity of technical regulations. Finally, at the systems level, the theory supports an effective development of a new class of rational Decision Support Systems (DSS), which should reduce the complexity of technical regulations and restore the logical consistency between the goals of technical regulations and the technical regulations themselves.

1. Introduction

Technical building regulations range from housing law to national building codes and standards issued by professional bodies. These regulations aim at securing acceptable levels of safety, health, usefulness and energy consumption for future occupants. But the logical consistency between these goals and the means to achieve them is hardly tractable. One problem is the large quantity of highly complex and frequently changing building regulations. Another is the variety of user demands; and yet another the steadily increasing body of technical information on (new) materials, products and buildings. Together, these developments have produced a very complex set of knowledge and data, which must be taken into account when dealing with technical building regulations.

As building regulations grow in complexity, scope and importance, the need for sophisticated regulation management systems becomes urgent. One of the entry points to realize such systems is the use of insights developed in information technology (IT). It is widely recognized that knowledge and database technology are important components of IT for improving the management of knowledge and data in regulations. In particular, the integration of knowledge

and database technology seems to provide a wide range of possibilities (see for instance McCulloch 1991; Koster and Stott Parker 1990). In general, two strategies of integrating knowledge and database technology can be distinguished. The first emphasizes the transfer of data structures and processing techniques from one field of research to another. In contrast, the second approach is concerned exclusively with the semantic structure of what is contained in the knowledge-based or data-based system (Newell 1981; Levesque 1984; Brachman and Levesque 1986; Twine 1989).

The aim of this paper is to show that the second approach, the **knowledge-level strategy**, in particular the theory of functional classifications, is more fundamental and more fruitful than the first **symbol-level strategy**. The fire-safety code, one of the most complex parts of the building regulations in the Netherlands, will serve to illustrate the main issues. To state them clearly, Section 2 first reviews the notions of the knowledge level and the symbol level. Section 3 explains the two strategies of integrating knowledge and database technology. Next, the knowledge-level strategy is discussed following a functional view. Finally, some implications of following a functional approach when developing Decision Support Systems (DSS) in the field of building regulations will be dealt with.

### 2. The Knowledge Level versus the Symbol Level

In his presidential address to the American Association of Artificial Intelligence (AAAI), Newell proposed a new computer-systems level, which he called the knowledge level (Newell 1981). The introduction of the knowledge level is primarily intended as a separate level within which knowledge can be defined. The global definition of knowledge that is often employed views knowledge as competence for selecting actions to realize goals. This competence is accomplished through knowledge of goals, knowledge of actions and knowledge relating goals to actions. From this notion of knowledge, it follows that relations between goals and actions are the basic components of knowledge. The **principle of rationality** structures these relations. More precisely, according to the principle of rationality, if the knowledge is present that an action will lead to a goal, that action will be selected. Under certain simple conditions the principle of rationality will be effective, but in many situations it is not suited for determining behaviour. Some of these situations can be covered by auxiliary principles. In case multiple actions are involved, the principle of rationality is extended with the auxiliary principle of **equipoise of acceptable actions**. The latter asserts that every action leading to a goal is equally acceptable from the viewpoint of the goal itself. A second auxiliary principle, covering situations in which actions are connected to multiple goals, is the **preference of joint goal satisfaction**. "For given knowledge, if goal G1 has the set of selected actions \([A1,1]\) and goal G2 has the set of selected actions \([A2,1]\), then the effective set of selected actions is the intersection of \([A1,1]\) and \([A2,1]\)" (Newell 1981, p. 9).

Notwithstanding the utility of these extensions, on many occasions, they fail to provide effective guidelines for predicting the system's behaviour. Even after the introduction of other auxiliary principles accounting for goal preferences, risk and uncertainty, the elementary extensions of the central principle of rationality are not sufficient to cover all situations. Knowledge-level models describing the environment do not contain encompassing principles which state that multiple goals need to be compatible or which solve incompatibility. The failure to determine behaviour uniquely, the probabilistic elements and the incapability to describe the entire range of behaviour all indicate that knowledge-level models are approximations of reality.

Knowledge used in the service of goals includes semantic issues (about the functionality of a system). But it completely excludes user-interface aspects (how to present the functionality of a system to users) as well as implementation aspects (how to encode the functionality). At the
knowledge level, attention is explicitly focused upon what knowledge is present. An essential part of Newell's proposal is the existence of a symbol level. Knowledge - from a knowledge-level perspective viewed as competence to select goal-related actions - is reduced at the symbol level to structures and processes. At the symbol level, data structures with particular properties and associated processes carry out problem solving to realize a goal-oriented functionality. The data structures contain knowledge and the processes provide access to this captured knowledge. This is represented by the following well-known equation:

\[ \text{Representation} = \text{Knowledge} + \text{Access} \]

Newell (1981, p.14) explains this equation as follows: "The representation consists of a system for providing access to a body of knowledge, i.e., to the knowledge in a form that can be used to make selections of actions in the service of goals.

This paper adopts the presumption that a clear conception of knowledge should logically occur prior to that of representation. This implies that work should be done at the knowledge level. But what is to be gained from this knowledge-level perspective? First, a major advantage of knowledge-level research is the provision of an adequate entry point for the analysis of the nature of knowledge. Such an analysis permits one to understand or predict behaviour without the need to construct an operational model of the implied process. A second advantage is the spin-off of knowledge-level research to conceptual modelling.\(^1\) A knowledge-level analysis explores the fundamental principles of conceptual models which will lead to conceptual models, free of inadvertent implementation biases (Brachman and Levesque 1986, p.77; Twine 1989, p.125).

Levesque (1984) formulates this as follows: "In terms of system design, the main reason for distinguishing between the knowledge level and the symbol level, is to allow the functionality of a system to be treated independently of its symbolic implementation. In particular, it allows us to consider new operations on a knowledge base (that can be explained in terms of existing ones) without necessarily committing ourselves to any particular implementation style" (Levesque 1984, p.206). Furthermore, knowledge-level models can be effectively used for validation and design purposes. According to Steels (1990), a knowledge-level perspective permits the development of deep conceptual models enabling the realization of robust systems. These systems are capable of providing - when compared with explanation facilities in traditional systems, which are simple replays of used data structures - more subtle and sophisticated justifications of conclusions. Thirdly, the knowledge level can serve as a vantage point to compare and examine critically the properties of representation formalisms. Those properties will continue to hold, no matter what symbol-level decisions are made. These comparisons and examinations should relieve information-systems developers of the burden of having to mould each problem to suit their implementation tools.

Though Newell's ideas have had an ever-increasing impact on IT, there is still great confusion between the knowledge level and the symbol level. Brachman (1985), for instance, points to the problems encountered by Fahlin, Touretzky and Van Reggen (1981) when they tried to ascertain the meaning of their inheritance mechanism and could not find a consistent interpretation. Conversely, semantic issues are often mixed up with symbol-level or implementation concerns. When discussing the functionality of a system, notions of inheritance mechanisms, frames and the like often crop up.

---

\(^1\) Conceptual modelling refers to a fundamental phase in the design of information systems in which knowledge and data about the application domain are collected, modelled and documented. It is also used to denote the methodology applied in this phase for collecting, modelling and documenting knowledge and data (see Di Blasi et al. 1989, p. 245). The results of the conceptual modelling phase are documented in conceptual models or conceptual schemata consisting of concepts.
3. Integrating Knowledge Technology and Database Technology: Two Strategies

Relationships between knowledge and database technology are central issues in computer science. Some researchers stress the similarities of knowledge technology and database technology, such as the common logic basis and the appearance of triggers. Other researchers point to differences such as the deductive proof-theoretic inferencing of knowledge-based systems versus model-theoretic query evaluation in data-based systems (Brooke and Jarke 1986) or emphasize the complementary nature of knowledge and data-based systems (Risch et al. 1988).

In spite of these efforts, the state of affairs with respect to the distinction between the knowledge level and the symbol level is no different from the general situation in IT. The main thrust of the research directed at the integration of knowledge and database technology is concentrated on transferring data structures and processing techniques from knowledge technology to database technology or vice versa. The use of records to store production rules (Herwijnen, Van Houten, Houtuma and Ronkema 1990), the use of frames to store relational data (Chow 1987), or the addition of rule-processing algorithms to databases (Stonebraker 1984) are all examples of research conducted at what we have called the symbol level.

It cannot be denied that knowledge and database technology can help each other at the symbol level. But integration at this level without understanding fundamental issues at the knowledge level leads to serious difficulties. Too much concern with representation mechanisms at the expense of knowing what function a system is computing has led to disadvantageous implementation biases. All too often, discussions about the utility of representation techniques like inheritance taxonomies, production rules and connection graphs lack correct understanding of what, if anything, these symbol structures indicate (Brachman 1983; Etherington and Reiner 1983; Berg-Cross and Price 1989).

A more fundamental and fruitful view of the relationship between knowledge and database technology is concerned only with the semantic structure of what is contained in the knowledge or data-based system. At the knowledge level, the distinction between these systems is non-existent. In terms of what knowledge is represented, as opposed to the data structures and processing techniques to encode the knowledge, both systems are simply collections of facts serving representational ends (Brachman and Levesque 1986, p.72; Twine 1989, p.125). The similarity between knowledge-based and data-based systems can be illustrated by applying some basic mathematical logic to describe the components of knowledge. At the knowledge level, a component of knowledge can be viewed as a set of functions. A function is defined as follows (De Brock 1989, p.6):

**Definition 1**

F is a function $\mathcal{F}$ is a set of ordered pairs and

$$\forall (x,y) \in \mathcal{F} \forall (x',y') \in \mathcal{F} : \text{if } x = x' \text{ then } y = y'$$

If $F$ is a set of ordered pairs (i.e. the ranking of the ordered pairs is not relevant and there are no duplicates), $\text{dom}(F) = \{ x | (x,y) \in \mathcal{F} \}$ and $\forall x \in \text{dom}(F)$ exactly one $y$ exists (i.e. the same $x$ cannot have different $y$’s, but the same $y'$ could be connected to several $x$’s), then $F$ is a function.

To illustrate the definition of a function, we revert to an example taken from the fire-safety code in the Netherlands. More precisely, we will look at the fire-safety regulations which attempt to limit the spread of fire by imposing demands on the fire-resistance of walls. For the sake of simplicity, we assume that two properties are important for determining the fire-resistance of
walls: thermal insulation and irradiance. The following sets of ordered pairs contain the knowledge describing the values of these properties of two walls:

\[ T_1 = \{ (\text{identifier} \{\text{we123}\}, \text{thermal insulation} \{25\}), (\text{irradiance} \{35\}) \} \]
\[ T_2 = \{ (\text{identifier} \{\text{we124}\}, \text{thermal insulation} \{35\}), (\text{irradiance} \{45\}) \} \]

\( T_1 \) and \( T_2 \) are functions consisting of three ordered pairs. Every domain element of a set is linked to exactly one value. Such functions are grouped into components, to make them more amenable to survey techniques. The definition of a component \( C \) is as follows:

**Definition 2**

\( C \) is a component over the set \( A \). \( \not\in \) \( C \) is a set and
\[ \forall c \in C : c \text{ is a function over } A \]

taking into account the following definition:

**Definition 3**

\( F \) is a function over the set \( A \). \( \not\in \) \( F \) is a function and
\[ \text{dom}(F) = A \]

If \( C = \{T_1,T_2\} \) then \( C \) complies with the definition of a component. \( C \) is an example of a component occurring in regulations for construction and industry. Those regulations comprise a set of functions with the explicit purpose selecting fire-resistant walls to limit the spread of fire. \( C \) is an element of an allowed set of components. The specification of this set is as follows (see De Breek 1989, p.26):

**Definition 4**

\[ WW = \{ C | C \subseteq \Pi(FW) \} \]
\[ \forall x \in WW : \forall y / x \in WW : \text{if } x \neq y \text{ then } t(\text{identifier}) \neq t(\text{identifier}) \]

taking into account that:

**Definition 5**

\[ FW = \{ (\text{identifier} \{\text{we120; we121;...}\}), \text{thermal insulation} \{N\}, \text{irradiance} \{N\}) \} \]

**Definition 6**

\[ \Pi(FW) = \{ f | f \text{ is a function over } \text{dom}(FW) \text{ and } \forall x \in \text{dom}(f): \]
\[ f(x) \in FW(x) \} \]

\[ N \] is the set of integers (including 0).
From a symbol-level perspective, component $C$ could be represented as a table of a database over $A$ with exactly four elements. Each element of $C$ corresponds to a tuple of a table. As every element of $C$ is a function, a tuple can be viewed as a function over the relevant set of field names (Figure 1 and Definition 3). A second option is the implementation of the elements of $C$ as production rules. Then $C$ forms a set of production rules (Figure 1). The two types of representation structures, namely records and production rules, employ search in quite distinct ways. The underlying search mechanism for extracting knowledge from tables is a form of query evaluation. The fundamental access mechanism of a knowledge base consisting of production rules boils down to some sort of deductive inferencing (Brodie and Jarke 1986, p.191; Smith 1986, pp. 8-12).

### Walls: a record-based database

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Thermal Insulation (in minutes)</th>
<th>Irradiance (in minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>w123</td>
<td>25</td>
<td>35</td>
</tr>
<tr>
<td>w214</td>
<td>35</td>
<td>45</td>
</tr>
</tbody>
</table>

### Walls: a rule-based knowledge base

Domain = \{identifier, thermal insulation, irradiance\}

- IF identifier = w123 THEN thermal insulation = 25 and irradiance = 35
- IF identifier = w214 THEN thermal insulation = 35 and irradiance = 45

**Figure 1: Two symbol-level representation structures**

The example shows that the distinction between both kinds of systems is non-existent at the knowledge level. A component described at the knowledge level has several representation options at the symbol level. But this does not imply that choice of a representation structure should take place at the symbol level. On the contrary, as we will argue, only at the knowledge level can well-founded choices of representation structures be made. One of the advantages of working at the knowledge level, mentioned in the previous section, is the availability of a separate, implementation-independent level from which representation structures and access mechanisms can be analyzed. An example of this is the view of databases from the knowledge level as described by Brachman and Levesque (1986). They state that it is possible to assess class membership after some theorem proving that one of two conditions is true without saying which one (using disjunction) or to state that an object satisfies a certain condition without saying what that phenomenon is (using existential quantifiers). A solution to this intractability problem is to limit the uncertainty expressible in an implementation language. Because record-based data bases are restricted in precisely this way, they can be considered as knowledge-based systems of a limited form. A data base is comparable to the knowledge representation component of a knowledge-based system. To illustrate a view of databases from the knowledge level we increase the number of functions of the previous example to four.

---

3 Note only physical objects are meant by the term object. It also refers to phenomena, processes etc. that are of interest.

4 "A knowledge-based system is any system that uses an explicit knowledge base in some capacity. The knowledge representation component is the part of the overall system that manages the knowledge base" (Brachman and Levesque 1986, p.72).
This produces the following table:

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Thermal insulation (in minutes)</th>
<th>Irradiance (in minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>we123</td>
<td>25</td>
<td>35</td>
</tr>
<tr>
<td>w214</td>
<td>35</td>
<td>45</td>
</tr>
<tr>
<td>w329</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>w491</td>
<td>25</td>
<td>35</td>
</tr>
</tbody>
</table>

Figure 3. A table on the fire resistance of walls

The range of uncertainty in these functions is quite limited. Disjunctions, negations or existential quantifiers are not present. A consequence of these limitations in expressiveness is a lack of knowledge. This becomes clear when we rephrase the question as: How many walls meet the fire-safety requirements for limiting the spread of fire? It appears that the knowledge expressed by the logic functions is not sufficient to answer this question. For example, we also need to know in what way the two heat-insulation attributes of walls are related to fire resistance.

Assume that we possess the knowledge that a wall meets the minimal fire resistance requirements if the heat-insulation properties both exceed 30 minutes. Then, the formal query to successfully answer the question should be:

\[ \lambda C \in \text{WW}: (C < \text{thermal insulation} > 30 \text{ and } C < \text{radiance} > 30) \]

But it is still not certain that the query yields the right answer. For it is possible that the list of functions is incomplete or that we123 and w491 are different identifiers of the same wall. To reinterpret the symbol-level question as the question originally posed, additional knowledge is needed:

\[ c_i \neq c_j \]

(each constant represents a unique individual)

\[ \forall \text{wall}(x) \supset x = \text{we123} \lor x = \text{w214} \lor x = \text{w329} \lor x = \text{w491} \]

(universal quantifier naming explicitly the positive instances)

Because of the extra knowledge, no reasoning is needed to find out how many walls are present. Inference is simply reduced to calculation. All the system has to do is count how many

---

5 If \( l \) is a set, then \(|l|\) is the number of elements of \( l \). The notation \( \lambda C \in \text{WW}: U \), in which \( U \) is an expression in \( l \), is a shorthand for \( \{(C U) : C \in \text{WW}\} \).
appropriate tuples appear in the wall relation. It does not have to reason by cases or contradiction. But suppose that the following knowledge about the relations between various types of walls and the minimal required fire resistance is captured in the system:

\[
\{(\text{wall-type:exterior}), (\text{minimal required irradiance:30 minutes})\}
\]

\[
\{(\text{identifier:swel23}), (\text{wall-type:exterior})\}
\]

The knowledge in the first function states that the minimal required irradiance for exterior walls is 30 minutes. The second function asserts that swel23 is an exterior wall. Implicitly, the knowledge tells us that the fire-resistance requirements for exterior walls are less stringent than for interior walls. The attribute thermal insulation is not relevant for exterior walls and, under certain circumstances, this might yield a higher fire resistance for exterior walls. Now, the previous conclusion that there is one fire-resistant wall should be replaced by the inference that two walls are fire-resistant. The example can be made more realistic by incorporating more attributes of walls and by introducing concepts defining conditional relationships between fire-resistance requirements and different types of spaces adjoining the walls represented in the database. Once again, other conclusions will be reached!

This example illustrates that a view of data bases from a knowledge-level perspective permits the assessment of the semantic consequences of adding knowledge. These assessments enable us to make decisions about the symbol-level trade-offs tractability/expressiveness. Actually, the example denotes a more universal problem caused by a symbol-level approach: the difficulties occurring while querying a database (see for instance Remmen 1985; De Jonge, Bruining, Schoemaker and Otten 1988; Van de Riet 1990). In the next section, a specific approach to knowledge-level research will be discussed.

4. Functional Object-Types as Knowledge-Level Models

A knowledge-level model should contain adequate knowledge of its environment. Since concepts play an important role as classificatory and storage mechanisms for the organization of knowledge, they can be regarded as basic epistemological components of a knowledge-level model. A concept has an intension and an extension. The intension of a concept is a set of sufficient and necessary conditions which should be satisfied by an object in order to belong to the class covered by the concept. The extension of a concept is the set objects complying with the intension. The intension relates to an object-type.

There are several basic views on how to reconstruct the object-type of a concept, one of which is the functional view (Hendriks 1986; Lucardie 1989; Reitma 1990; Van der Smagt and Lucardie 1991). According to the functional view, an object-type of a concept is established by a goal-oriented reconstruction process wherein a disjunction of conjunct sets is modelled. Such a disjunction is depicted in the treelike schema of Figure 3. This disjunction consists of three conjunct sets, all leading to goal 1:

\[
\{(1 \land \lambda \iota 2-1)\},
\{(1 \land \lambda \iota 2-\sigma)\} \text{ and } \{(1 \land \lambda \iota 2-\sigma, \lambda 3-\sigma)\}
\]

An element of a conjunct set is an INUS-condition: an Insufficient but Necessary part of the conjunct set which is Unnecessary but Sufficient for the result. Within a conjunct set, an INUS-
condition is indispensable for achieving a goal, but the conjunct set itself, to which the INUS-condition belongs, is replaceable by other conjunct sets.\textsuperscript{6} As an object must satisfy one of the conjunct sets of an object-type in order to belong to the extension of a concept. The example, though greatly simplified, shows some interesting features of the functional approach. For instance, it is possible for objects, which at first sight are different, to be identical in the context of a goal. Or, in other words, objects having different attributes, but matching with a conjunct set of an object-type, are equivalent. An example of two 'different' but functionally equivalent objects is formed by an object 1 characterized by the attributes \{((j,b),(2,v))\} and an object 2 having the attributes \{((j,b),(1,v),(3,y))\}. In the context of goal 1 both objects are similar. Here, the notion of functional equivalence is essential: objects are identical, fall in the same concept or are similar if they possess - even quite different - attributes to perform the same function. Three mechanisms are responsible for the fact that a goal or a function is attainable by quite different strategies.

![Diagram](image)

**Figure 3.** A disjunction of three conjunct sets

The first strategy is the mechanism by which, under certain conditions, other attributes (descriptors) may become important for determining class membership. In the third conjunct set, \{(2,v)\} becomes a descriptor if \{((j,b),(2,v))\}. This mechanism is effective in the fire-safety regulations limiting the extension of fire. These fire-safety requirements for walls are influenced by the conceptualization of rooms adjacent to the walls. Fire compartments, bathrooms, lavatories and traffic rooms all are examples of types of rooms which have different fire-resistance requirements. Therefore, it is important to have adequate definitions of these rooms at one's disposal. If we look at the conceptualization of a fire compartment, we can see that quite 'different' rooms can be classified as a fire compartment: both a heating room and a technical room can function as fire compartments. But the technical room should have a user surface

\textsuperscript{6} The definition of the INUS-condition is not free from discussion. Mackie defines an INUS-condition as follows: "A is an INUS-condition of a result \(F\) if and only if, for some \(X\) and some \(Y\) (\(A\) and \(X\) or \(Y\) is a necessary and sufficient condition of \(P\), but \(A\) is not a sufficient condition of \(P\) and \(X\) is not a sufficient condition of \(P\)." (Mackie 1965, p.246). The definition of Mackie is amended by Dennie (1984) as follows: "A is an INUS-condition of a result \(F\) if and only if, for some \(X\) and some \(Y\), \((A\) and \(X\) or \(Y\)) is a necessary and sufficient condition of \(P\), and \(A\) is necessary condition of \(P\) and \(X\) is not a sufficient condition of \(P\) and \(X\) is not a sufficient condition of \(P\)." (Dennie 1984, p.30).
exceeding 50 m². The user-surface attribute is virtually a new descriptor, which is only useful for the conceptualization of fire compartments in cases where we are dealing with a technical room.

The second strategy is that categorizations of attributes of objects influence each other. This phenomenon is called conceptual interaction. In Figure 3, conceptual interaction manifests itself in the mutual influence of the categorizations of the first and second attributes. If \( ((l_1 a)) \), the classification of the second attribute is \( (k_2) \), if, on the other hand, \( ((l_2 b)) \) the classification of second attribute is \( (r_2) \). Referring again to the conceptualization of fire compartments, conceptual interaction is present between the type of room and the user surface. If we are dealing with an enclosed room, the adequate categorization of user surface is \( (< 500 \text{ m}^2 \text{ or } > 500 \text{ m}^2) \), because the user surface of an enclosed room should not exceed 500 m². However, if we are dealing with a technical room, the categorization of user surface is \( (< 50 \text{ m}^2 \text{ or } > 50 \text{ m}^2) \). To be classified as a fire compartment, the user surface of a technical room should exceed 50 m².

The third strategy refers to the situation that objects may have different attribute values, but that this variation is limited to, or falls within, a goal-constructed category. Two objects characterized by the same \( ((l_1 a)) \), but with different values of \( ((l_2 b)) \) - object 1 and object 2 respectively have the values \( k_1 \) and \( k_2 \) with \( k_1 \) and \( k_2 \) both falling in the category \( k \) of \( ((l_2 b)) \) - are functionally equivalent. A technical room with a user surface of 55 m² and a technical room with a surface of 60 m² are functionally equivalent in the context of defining a fire compartment.

Functional equivalence shows that no a priori defined concepts are allowed to describe a knowledge-level model; only goal-constructed concepts are. From a functional viewpoint, it is a prerequisite that a knowledge-level model should avoid describing its application domain by a priori fixed categorizations. Rather, a knowledge-level model should avoid these reifications by accounting for goal-based dynamic, flexible categorizations of the environment. Under the influence of (changing) goals, continuously different descriptors will be needed to assess class membership. Not the object properties in their own right are relevant, but the functionally required properties. Functional equivalence stresses the structural heterogeneity of objects; in many situations, the presumption that objects are describable by fairly stable characteristics will be a misconception.

We have seen that one of the advantages of working at the knowledge level is the possibility to analyze representation formalisms at the symbol level. When we examine the basic properties of record-based information models from a functional viewpoint, we gain some interesting insights. If the essential configuration of the conjunct sets leading to the same goal is characterized by identical attributes, each attribute having the same kinds of values, records are excellent representation and processing tools. If, on the contrary, these conjunct sets are characterized by heterogeneity, which is caused by having to classify objects in a goal-oriented fashion accounting for different descriptors and new conceptual interactions, records are not appropriate (Kent 1979, Maier and Johnson 1990).

The more a domain deviates from homogeneity, the less appropriate the record configuration is. Referring again to our example, it is easy to see that many attributes play a role in limiting fire extension: external walls, internal walls, thermal insulation, irradiance, fire compartments, technical rooms, enclosed rooms, traffic rooms and so on. Furthermore, the definitions of different types of rooms yield other attributes. For instance, the definition of a technical room states that it is an enclosed room for the installation of equipment necessary for the functioning of a building. This definition yields at least two extra attributes.

There are certain techniques for accommodating the variability in the description of an object-type in a record-based database design. We mention the design of multi-field tables, multi-meaning fields and multi-table databases. Unfortunately, each of these designs has its drawbacks. A database design in which all attributes are represented in multiple fields of one table is likely to have null-values in many fields. A simple illustration is the fact that if we are dealing with an external wall, the attribute thermal insulation is not relevant. This delivers a null-value in
the corresponding record and field. A more serious problem is that no system facility will enforce a correct input, because this limited relevance is not defined to the system. So the multi-field format will introduce a data-integrity hazard.

A database design in which the same field is allowed to have different meanings is inefficient because every field value has the same meaning for the system. In this design, the system is unable to navigate through tables by matching key values. Only in the buried logic of an application program can the significance of field values be made clear. Also, a database design in which different multi-field tables represent a conjunct set of an object-type is not a workable solution. A user has to know the number and names of tables representing an object-type and be able to query them. Validation is still a problem. Again, data-integrity hazards endanger the suitability of this representation formalism. All these record-based solutions to cope with heterogeneous but functionally equivalent objects remain makeshift. For a more elaborate discussion on the limited capabilities of record-based information models, see Kent 1979.

5. Conclusions

This paper has explored the possibilities of integrating knowledge and database technology to manage the complexity of regulations for design and construction. In studying the integration of knowledge technology and database technology, the assumption is adopted that a clear conception of knowledge should logically precede that of representation. Knowledge-level research completely excludes user-interface and implementation aspects, but it includes semantic issues which can be effectively used for modeling, validation and design purposes. We have asserted that the theory of functional classifications is consistent with Newell’s idea that “knowledge is to be characterized entirely functionally, in terms of what it does, not structurally in terms of physical objects with particular properties and relations” (Newell 1981, p.10).

At the knowledge level, the functional approach accounts for a rational goal-directed strategy by stressing the flexibility and dynamics responsible for the complexity in building regulations about which we wish to communicate knowledge and data. It enables the reconstruction of (deep) models of regulations, of complex objects and of the different users of the regulations.

At the symbol level, the functional view supports the examination and comparison of representation formalisms. For instance, the theory of functional concepts can explain why record-based information models are limited. In situations where quite different attributes or conceptualizations of objects can fulfill the same function, and thus are functionally equivalent, the assumption of vertical and horizontal homogeneity of record-based information models is not matched (Kent 1979).

At the systems level, the introduction of a knowledge-level perspective, in particular the theory of functional concepts, permits an effective development of a new class of DSS. A DSS constructed according to a functional approach could be of use for the design and modification of building regulations and for compliance-checking by applying these regulations to complex objects. For each of these modalities, a DSS could be used to check standards on completeness (regulations should be applicable in every situation within their scope), correctness (the meaning of the regulations is intended by the code designers), and consistency (regulations do not contradict each other). In particular, the phenomenon of different actors, each with their own goals and needs having to work with building regulations, requires extended capabilities for representing and inferring complex knowledge and data.

Although research on functional object-types has evolved from a critical evaluation of the ‘causal modeling approach’ in quantitative scientific research, the principles of the functional approach can serve as a paradigm to reconstruct knowledge-level models and to design and implement DSS. Knowledge-level research conducted according to a functional viewpoint of the
goals and the technical regulations to achieve them should reduce the complexity of the regulations and restore the logical consistency between the goals and the technical regulations.

References


Brachman, R.J. (1985) ""I Lied about the Trees" Or, Defaults and Definitions in Knowledge Representation", AI Magazine 6, 80-93.


