Structuring the Knowledge Domain of Office Functions with Expert Characteristics

Kees van Slooten

School of Management Studies
Twente University of Technology
PO Box 217
7500 AE Enschede
Netherlands

Abstract

Many office functions have problem solving characteristics and use heuristic knowledge. To support these functions, techniques from artificial intelligence can be used to represent knowledge. However, the process of knowledge acquisition in order to define the knowledge is not very structured. Some information analysis methods, used during the development of information systems in general, have the potential to structure the process of knowledge acquisition. However, the information analysis method which is used should have enough possibilities to define the conceptual aspects of the knowledge domain completely. Such an information analysis method has been researched in the course of a case study with positive results.

Keywords:
Office functions, knowledge acquisition, information analysis, managerial office work.

1. Introduction

The first step in automation of office functions was the automation of office devices (typewriter etc.), the second step was the automation of office procedures with routine characteristics. The next step is the automation of office procedures with non-routine characteristics. This causes a shift from automating tasks to automating functions [1]. The functions considered here have problem solving characteristics, in other words are type II office functions [2]. Techniques from artificial intelligence (AI) can be used to represent knowledge (knowledge-based systems) in order to support the automation of office functions with problem solving characteristics.

It is a well known fact that knowledge-based systems make use of different techniques to represent knowledge, for example, production rules and frames. Knowledge-based systems can be developed to support office-functions with problem solving characteristics.

The process of knowledge acquisition, in connection with the development of such knowledge-based systems in order to support office functions, is often not very structured. The purpose of this paper is to make clear that the process of knowledge acquisition can be made more structured by using information analysis techniques. Information analysis is a phase in the development of information systems which results in a complete specification of all conceptual aspects. Here we will use NIAM, the Nijssen Information Analysis Method [3]. The graphical notation of NIAM is an extension of the Entity Relationship model and approximates proposals for international standards [4].

The subject of the case study is an advisory function in the area of package holidays — a travel agency function.

In section 2 a few examples are described of structuring knowledge in this domain. In section 3 the development cycle, followed by this case-study, will be explained and compared to the development cycle generally followed during the development of information systems. A review of some conclusions of the case-study and further investigations is given in sections 4 and 5 respectively.

2. Structuring the Knowledge Domain

The NIAM information analysis method is based on three principles:

1. The conceptual principle.
The grammar of the information system describes exclusively the conceptual aspects of exchanging information between the information system (knowledge-based system) and the environment (figure 1). The grammar is a description of rules for the knowledge (information) in the information base. Information can be considered as transferable knowledge.

2. The 100% principle.
The grammar describes all conceptual aspects of exchanging information between the information system (knowledge-based system) and the environment.

3. Information can always be described by elementary deep-structure sentences of a natural language. An elementary deep-structure sentence is an elementary sentence in such a form that all knowledge necessary to understand the sentence is completely and explicitly represented by the formulation of the sentence.

The sentence: " 'Farewell to Arms' is written by Hemingway" is not a deep-structure sentence, but the sentence: "The book with title 'Farewell to Arms' is written by the author with the name Hemingway" is an elementary deep-structure sentence. An elementary sentence cannot be split up into other sentences without losing information.

A graphical notation is used to represent elementary deep-structure sentences. Entity-types are represented by closed circles, attribute-types (or label-types) by dashed circles

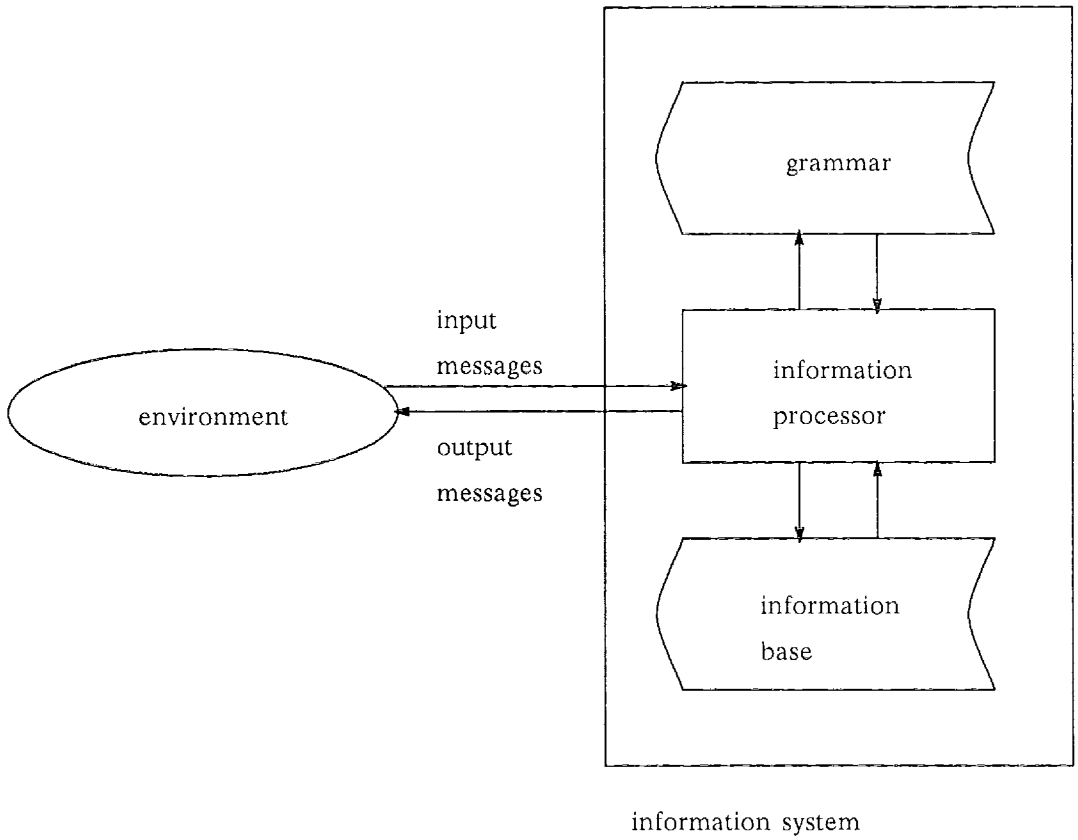

Figure 1

Basic model of an information system

(figure 2).The graphical notation can be replaced by first order predicate logic. A few examples of grammatical aspects are explained in the rest of this section.

The subset constraint (represented by an arrow between the roles "takes-place-in" and "belongs-to" in figure 2) can be formulated as follows:

Informal formulation:
For each TRIP t the following holds:
    If t "takes-place-in" a CONTINENT
    then t "belongs-to" TYPE-OF-TRIP

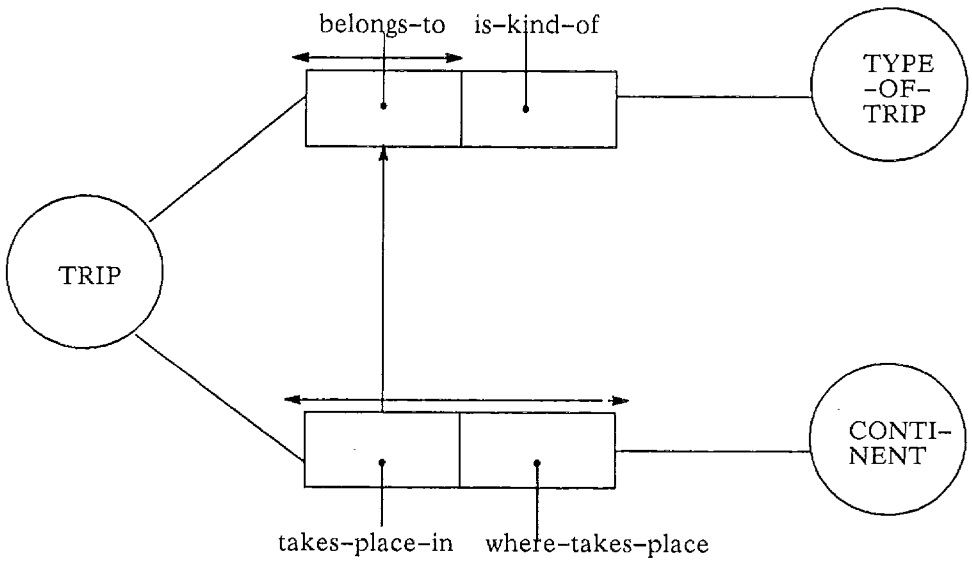

Figure 2
Subset constraint

Formal formulation:

$\forall$ t $\in$ TRIP [$\exists$ c $\in$ CONTINENT [takes-place-in (t,c)]
$\Rightarrow$ $\exists$ tot $\in$ TYPE-OF-TRIP [belongs-to(t,tot)]]

The uniqueness-constraint (represented by an arrow over the role "belongs-to") can be formulated as follows:

For each TRIP t the following holds:
The number of elements of TYPE-OF-TRIP "is kind of" TRIP t is less than or equal to 1.

A subtype-rule is a rule that indicates that an object of the supertype is also an object of the subtype (figure 3).

Informal formulation:

For each TRIP t the following holds:
t is a cruise if and only if TYPE-OF-TRIP "is-kind-of" t is a cruise.

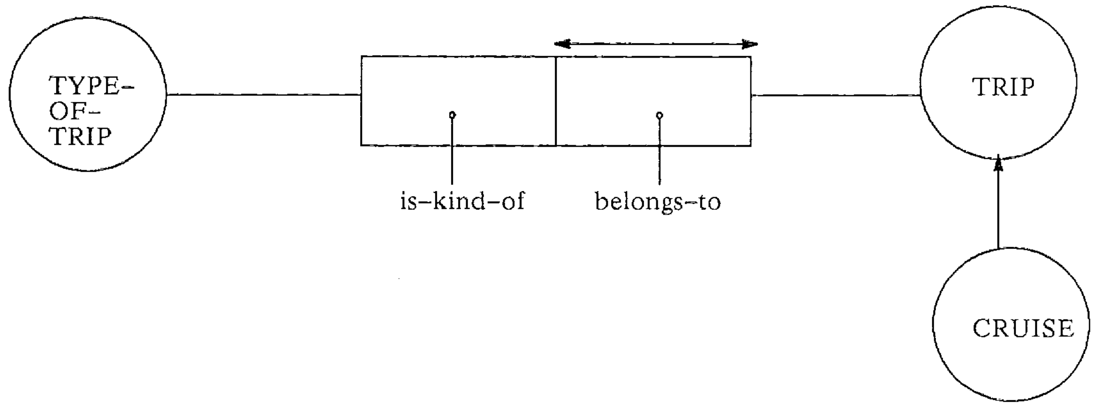

Figure 3
Subtype-rule

Formal formulation:
$\forall$ t $\in$ TRIP, cruise $\in$ TYPE-OF-TRIP
[is-a-cruise (t) $\Leftrightarrow$ belongs-to (t,cruise)]

Also derivable and inferable sentences can be included into the conceptual grammar. Derivable sentences represent redundant information. Inferable sentences are not directly derivable from the other sentences of the conceptual grammar. Such sentences are inferable by the experts. The graphical notation describes a part of the conceptual grammar. Some constraints cannot be represented graphically, but they are also elements of the conceptual grammar. For instance, a value rule can prescribe the values of an attribute-type as follows:

Informal formulation:
For each AMOUNT-OF-MONEY a, the following holds:
if a "is-budget-of" TRIP then a is greater than or equal to $500.

Formal formulation:
$\forall$ a $\in$ AMOUNT-OF-MONEY, $\forall$ t $\in$ TRIP
[is-budget-of (a,t) $\Rightarrow$ a $\geq$ 500]

In figure 4 the fact-type (sentence-type) has been drawn. Sentences between entity-types and attribute-types are called bridge-type sentences. Sentences between entity-types are called idea-type sentences.

In this way non-graphical rules can be represented, for example, cardinality rules. The knowledge domain can be structured with the help of the specifications, which have been

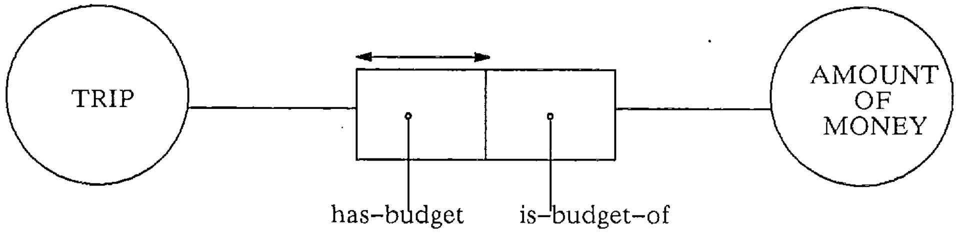

Figure 4
Bridge-type sentence

presented here. There are numerous other aspects in connection with the grammatical notation, which have not been mentioned here, but which would require a lengthy description.

3. Development Phases

In section 2 an attempt was made to support the process of knowledge acquisition with a combination of graphical and non-graphical specifications, in other words, a notation to describe the conceptual aspects of the knowledge domain formally. This conceptualization is an important step in the development process of knowledge-based systems for office functions with problem solving characteristics.

In general, the following steps can be followed [5]:

- identification
- conceptualization
- formalization
- implementation
- testing

The identification phase is called 'object systems analysis' (requirements analysis) in the development cycle of information systems.

The boundaries of the object system have to be determined. The conceptualization and formalization phases are called 'information analysis' in the development cycle of information systems. The determination of the information structure and the graphical/non-graphical rules, in other words, the determination of the grammar of the information system (in this case the knowledge-based system) can be done within the conceptualization and formalization phases.

The implementation phase and the test phase are comparable to the realization phase in the development cycle of information systems.

The identification, the conceptualization and the formalization phases are together the process of knowledge acquisition.

A case-study was done to determine whether the usual steps (object system analysis and information analysis) can be followed during the development of a knowledge-based system supporting an office function; in this case the advisory function was in the domain of package holidays.

The implementation was realized using an existing knowledge-engineering shell. There are different possibilities for knowledge representation during the implementation phase of the knowledge-based system. These possibilities include:

i) Production rules
A production rule is a combination of "condition" and "action" (if-then reasoning). An example of a production rule:

```
 IF
    type-of-trip = skiing holiday
THEN
   type-of-weather = cold
ENDIF
```

ii) Hypothesize and test
Knowledge is represented by frames. Each frame consists of descriptive statements related to the phenomena which are of interest. An example of a frame:

```
   Skiing holiday in Aspen (USA)
          [description:
                   type-of-trip = skiing holiday
                    activities = sport
                   continent = North-America
                   country-am-or = USA
                    duration ≥ 14
                   budget ≥ 4000
                   time = December / January / February]
```

iii) Bayes
For very large amounts of data, knowledge is represented by probabilities.

The choice of a knowledge representation for the implementation of an expert system is a realization aspect. The specifications of the analysis phases of the development cycle should be independent of realization aspects. In our case, production rules were chosen.

4. Some Conclusions

- The steps followed in the design of a knowledge-based system are the same as those in the design of information systems in general.

However, the information analysis method which is used should allow one to represent all conceptual aspects and the specifications should have formal characteristics.

- An information analysis method is a tool to reach completeness in conceptualizing the knowledge domain.

- At the least, indications for the knowledge representation (for example, production rules, frames) can be derived from the resulting information structure (graphical conceptual grammar).

- The information analysis method which is used should be based on natural language. This is advantageous in communicating with domain-experts.

- The information analysis method leads to correct conceptual specifications as a starting point for further knowledge representation in the knowledge-based system.

For instance, the graphical constraints can support the definition of knowledge elements in the knowledge domain and the non-graphical rules are the basis for the definition of knowledge rules for the rule-base.

- The information analysis method entails thought about each fact-type (sentence-type), this is helpful in structuring the knowledge domain.

5. Further Directions for Investigation

- Case studies of more office functions.
- The implications of the conceptual specifications on knowledge representation.
- Completeness of conceptual specifications.

6. References

1. Zisman, M. D., "Representation, Specification and Automation of Office Procedures", Dissertation, Wharton School, Univ. of Pennsylvania, 1977.

2. Woo, C., Lochovsky, F. H., "Supporting Distributed Office Problem Solving in Organizations", ACM Transactions on Office Information Systems, Vol. 4, No. 3, July 1986.

3. Verheijen, G.M.A., Bekkum, J. Van, "NIAM - An Information Analysis Method", Information Systems Design Methodologies: a comparative review", IFIP, CRIS - Conference. North-Holland, 1982.

4. Griethuijsen van, J. J. (ed.), "Concepts and Terminology for the Conceptual Schema and the Information Base", Report of ISO TC 97 / SC5 / WG5, 1982.

5. Hayes-Roth, F., *et al.*, "Building Expert Systems", Addison-Wesley, 1983.