8-16-1996

Advancing CASE Productivity by Using Natural Language Processing and Computerized Ontologies: The ACAPULCO system

Alexander Hars
Department of IOM, University of Southern California, ahars@sba.usa.usc.edu

Advancing CASE Productivity by Using Natural Language Processing and Computerized Ontologies: The ACAPULCO system

Alexander Hars
University of Southern California, Dept. IOM
Los Angeles, CA 90089-1421
ahars@sba.usc.edu

Abstract

We present a new approach to software engineering which reduces the knowledge gap between user and development methodology by explicitly supporting concepts expressed in natural language. The tool uses a natural language description of a business process as input and transforms it into a process model. The system recognizes actors, objects, locations, relationships etc. referred to in the description and distinguishes different types of actions and conditions. The system uses multi-pass parsing and disambiguation NLP techniques and relies upon a custom-built dictionary of 23.000 English root words. The dictionary includes information about syntactic (e.g. noun, verb...) and semantic categories as well as word frequency. Currently 15 different semantic categories such as 'tangible object', 'person', 'event', etc. are distinguished. The ACAPULCO prototype, which runs on a standard PC under Windows 3.1 with 16 Mbytes of RAM, demonstrates a) that natural language processing for software engineering is feasible, b) that this approach has potential of redefining the interaction and relationships between users, analysts and developers and c) that this approach is a powerful extension to traditional methods because it uses explicit knowledge about real-world business concepts.

Introduction

One of the key problems in systems analysis and design is to translate the description of a business problem into a formal representation which is more suitable for developing technical solutions. It has long been recognized that a semantic gap exists between users and developers which remains difficult to bridge (Kaiser and Bostrom 1982). While the user tends to take a broad organizational view to solve specific problems, developers use graphical specification languages such as the entity relationship model or process models to systematically describe business requirements. Conventional CASE-tools usually begin with these diagrammatic languages and do not offer direct support for capturing the user perspective or transforming a users problem description into more structured forms. The result is frequent miscommunication between users and developers, often leading to reduced user involvement and project failure. The lack of support for the user perspective may also be one of the key reasons why CASE tools have not been able to live up to their initial promises.

In this paper we introduce a prototype system which demonstrates a natural-language based approach to improve the productivity of CASE tools. The prototype uses problem descriptions in natural language - the language of the user - and transforms it into some of the diagrams used by the developers. By using a dictionary of natural language concepts with an explicit representation of semantics, the prototype also incorporates real-world knowledge which can be used to evaluate the description, detect relationships between different concepts and modeling views and guide the user through the description of the business problem. By using a simple ontology, the prototype overcomes one of the key limitations of traditional CASE tools - the scarcity of constructs between which a tool is able to differentiate.

Limitations of current CASE tools

Current CASE tools use a small number of constructs to represent the elements and relationships which are relevant for an information system. Although different constructs are utilized for business and technical perspectives of an information system (Sowa and Zachmann 1992, Scheer 1994), the total number of constructs differentiated in CASE tools is extremely small compared to the complexity of a business domains. The range of things that may be represented as "OBJECT" in an object-oriented methodology, is

very broad; it includes tangible objects such as "cars, buildings", abstract objects such as "account", events such as "order", persons such as "customer, sales representative, employee" even activities such as a "project". As a result, there are almost no conclusions which can be drawn out of the fact that something has been modeled as an object. As Boulding (1956) stated so eloquently:

"We always pay for generality by sacrificing content and all we can say about practically everything is almost nothing".

Natural language processing allows to bridge this gap and reintroduce content into systems analysis and design methodologies. This approach underlies the development of the tool presented. It is a radical departure from traditional and object-oriented development paradigms. It returns the focus to the user or business owner of an information system, and uses his/her language as the basis for communication and reference. This is different from the traditional approaches where an artificial - usually diagram based language - that is deemed most appropriate for technical development is used and the user is expected to communicate and reason in that language. Using a natural language-based tool allows to represent often neglected critical business concepts such as goals, locations or even some temporal relationships and to actively use these concepts as part of the development methodology.

System overview

The ACAPULCO tool was built to show that development tools can be built on this concept and to use the interaction with business persons to learn about factors that they regard as vital but that are neglected in current methodologies. The approach taken for ACAPULCO was to begin with process descriptions and expand the approach subsequently to include additional types of statements (e.g. structural statements). To transform textual process descriptions into a diagram, two objectives must be achieved: The tool must be capable of identifying the condition-action structures used in natural language, for example in the form of 'if'..'then' statements and their variations. A prerequisite for this is that the tool is capable of identifying action structures within sentences. The sentence "check available rooms and determine room rate", for example, consists of two actions. Secondly, the tool must be capable of identifying the key concepts that are involved in each action or condition and determine their semantic category. From the business process shown in figure 1, for example, the system automatically recognizes the following types of PERSONS involved "customer", "sales representative", "manager", "front desk manager" in different actions. The tool must be capable of distinguishing these concepts from locations such as "room" or tangible objects such as "room key".

The tool achieves these objectives by using multi-pass parsing and disambiguation techniques based on a custom-built dictionary with semantic, syntactic and word frequency information. The dictionary currently contains approximately 23.000 English root words and includes information about word type (noun, transitive verb, etc.) as well as information about concept category. Currently 15 different categories are distinguished (e.g. tangible object, event, person etc.). A word may map to several categories at the same time. The dictionary only contains single words. Word combinations such as "sales representative" or "shop floor" are not contained in the dictionary. Word categories are used to identify key elements in a business process and can be used as the basis of plausibility checks. Depending upon the type of activity, for example, an action may require tangible objects or information. If these concepts are missing, then the tool can alert the user and ask him to fill in the missing information. The approach of using basic semantic information is a powerful concept that could also be applied in traditional methodologies and improve, for example, consistency checking, identification of synonyms and schema integration activities.

Customer arrives at front desk

If the customer has no reservation, then sales representative checks available rooms and determines room rates.

Front desk manager apologizes to the customer if no rooms are available.

Representative assigns room to customer depending on his preferences and register the customer.

If the customer is a frequent customer and manager approves then upgrade customer to business room.

Sales representative makes a copy of the customers credit card.

Give room keys to customer.

Figure 1: Textual description of a hotel check-in business process

Example: Check-in process

Figure 1 shows a description of a hotel check-in process. The tool recognizes the sequence of business actions and condition-action structures and converts them into a process diagram an excerpt of which is shown in figure 2 which can be the basis for developing the corresponding IS application. The tool mimics some of the practices used by the analyst. For example, the subjects of most actions are omitted from the label to shorten the text to be displayed. Instead of displaying "customer arrives at front desk" the system displays "arrive at front desk". In straightforward cases, the system also recognizes and eliminates negations in conditions. For example instead of the condition "no rooms are available", its negation "rooms are available" is used. The current version of the tool does not do much more than generating a business process diagram and determining the concepts and categories involved in each action and condition. The process diagram, however, could be directly used as input to a lower CASE-tool which then would generate a program outline.

The primary objective of the tool, however, is not to automatic generation. Rather, it is the interactive elicitation, refinement and validation of business rules and business processes. The generation of a diagram from an initial textual description provides immediate feedback to the user and encourages reflection about critical or missing parts of the process. The tool itself can be used to ask quasi-intelligent questions to resolve potential problems. Such questions can be rooted in the categories of concepts that are mentioned and their relationships. For example, the tool could generate a question whether the "manager" referred to in one action and the "front desk manager" referred to in another question are the same entities. They could also be rooted in structural properties of the process. The next versions of the tool will focus on such interactive improvement of processes and thereby lead not only to correct process transformations but also to significant improvements in model quality.

Conclusion

This research has shown that natural language support for systems analysis and design is feasible. The ACAPULCO tool returns the focus in systems analysis to the way that the business user perceives and reasons about his domain. Instead of forcing the business to think in terms of objects, inheritance relationships and methods, this tool forces the development methodology to support those concepts that matter for business. The challenge for development methodologies is to incorporate more knowledge about key business concepts, their relationships and implications. The dictionary developed as part of this project and the tool are a first step to significantly transforming and improving development methodologies.

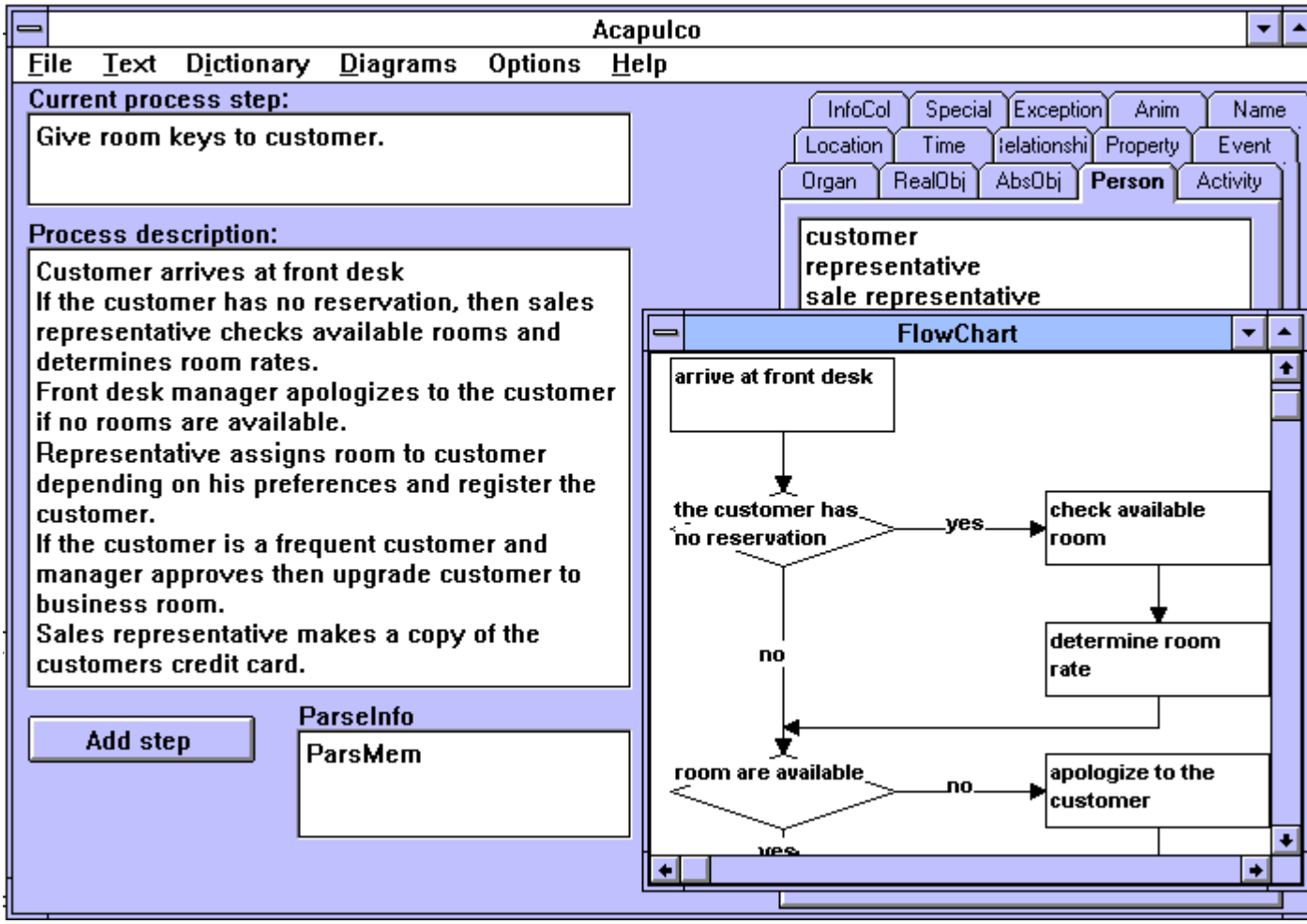

Figure 2: Hotel check-in process description and process diagram generated by the Acapulco tool

References:

Boulding, K.E. (1956). "General systems theory - the skeleton of science", Management Science vol 2 no 3.

Kaiser, K.M. and Bostrom R.P., (1992). "Personality characteristics of MIS project teams: an empirical study and action research design", MIS Quarterly, December, 43-59.

Sowa J.F., Zachmann J.A. (1992) "Extending and formalizing the framework for information systems architecture", IBM Systems Journal, Vol. 31, No. 3, 590-616.

Scheer, A.W. (1994). Architecture of integrated information systems, 2nd ed., Berlin.