An intelligent digester for interactive text processings

K. Hanakata
University of Stuttgart
Institute for Informatics

F.R.Germany

Abstract: This paper outlines a practical approach to our project to design an intelligent system that supports an expert user for his understanding and representing news reports on a special world of the japanese computer industry. With the extensive help of the intelligent system the expert user purposefully assimilates new information in a knowledge data base of a frame-like representation through a multiple window interface.

1. Introduction

Recent computer application to the so-called office automation is characterized by the increasing use of intelligent software systems. These systems are offen required to interface with textual information via users who are more or less trained or expertized to deal with the kinds of information to be processed. The ideal goal of such a text processing system is to transfer all experts'processing tasks to the system. It must be well recognized, however, that even with regard to the most advanced knowledge-based natural language processing systems such as SOPHIE(1), SAM(2), FRUMP(3), GUS(4), EXPLUS(5) this ideal goal seems to be still far from the present state of the natural language (NL) processing techniques, though they are very promissing. There are many reasons why existing NL systems can hardly be applied to the above goal. Two of them may be formulated as follows:

(a) The present relatively robust NL systems have been basically designed to deal with each separate sentence and not with a set of semantically related sentences. There are many research efforts aiming at this point (cf. eg. approaches of the Yale NL school).

(b) Because the NL understanding systems are generally concerned with deep understandings of sentences, they often take care too much of general relations and (common sense) reasonings in every detail, to which readers don't pay attentions so much in understanding them. Human readers skillfully control depth and details of understandings being conscious of their potential usefulness. In short, the present NL systems are not enough explicitly designed to extract purposeful meaning of sentences (cf.(4)).

In this developing state of NL processing and knowledge representation techniques we set a practical goal of our project to design an **interactive system** which digests news and reports on the **foreign computer industry** to support experts by a reactive knowledge data base. At the moment such a knowledge data base is urgently needed for experts to do their tasks such as writing reports to be submitted to some governmental institutions, industrial credit bureaus or agencies and the like.

2. Interactive assimilation and abstraction

Typical experts' tasks in an industrial credit bureau above mentioned are, for instance , to collect information about a special world such as the news world of computer industry. They collect clippings of news papers or journals, make memos from other news media and classify them according to their topics or themes. Their understanding of such information means that they not only understand the meaning (surface semantics) of each article literally , but also assimilate it into their related knowledge base (deep semantics) in such a way that it can be effectively used in some future contexts. That is, they identify relations of the new information with

This research was supported in part by Ostasien-Institut,e.V., Bonn

some parts of their knowledge and recognize its potential usefulness for their new conceptualization to advocate, to emphasize or build up their opinions, judgements or beliefs. The way of summing-ups by abstraction and assimilation of successive information to construct a special knowledge-base strongly depends on the purposes, as it is generally stated in empirical epistemology(15,16). In our practical case this means that the experts purposefully 'understand' the news to write a comprehensive report on the computer industry in which they draw from it conclusions such as future tendencies or proposals for some industrial policies, etc.

Taking both available techniques and reality of problems into account, we decided on the whole not to directly apply NL processing approaches to our expert support system. Instead, we extensively use NL techniques for case by case. Starting with the lowest intelligent level of interaction, we hope that we could continuously elevate the intelligent level of the machine processings in the recycling course of design and experiences (6).

The interface of the machine-expert interaction is based on the multiple window display communication (s.sect.5) and the mechanical representation of experts knowledge is founded on the frame-like formalism (s.sect.3 and 4).

In the interactive text analysis and meaning representation an expert reads a news article from his special domain of computer industry and has various levels of understandings typically reflecting his special domain dependent knowledge and his evaluation of news values. Some part of his special knowledge is activated, new information from the article is eventually added on it, a part of the existing knowledge is acknowledged, refined, amplified or reversed, etc.

In contrast to this explicit flow of information from texts to the expert user there is a implicit counter flow of predictive, expective or focal information which is feedbacked to the texts. In this sort of "backed-up" interaction between text source and expert's knowledge, there are roughly two classes of information to be represented:

(i) Conceptual information

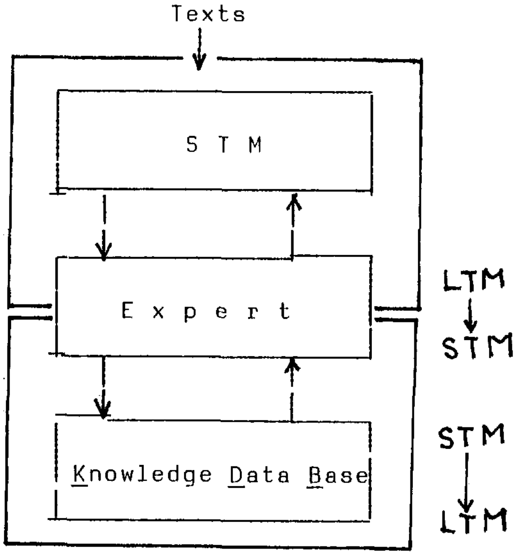
Generally speaking, the content of a news article is expressed in text sentenses, ie a restricted set of natural language in which they are written. However, for the expert of a special news world texts are a set of key concepts mostly well defined and structured within his knowledge, though they are organized by general syntactic rules of the language on a surface level. These key concepts indicate for him how to use them. The methods of using his knowledge, are described in procedures and structurally inherent in the concepts. The expert's responses to a key concept scanned from a given text are that

(a) related concepts around a given key concept are stimulated
(b) some concepts are expected in the subsequent scanning of the texts.
(c) These stimulated or expected concepts interact with each other to constrain and filter their semantic roles.

(ii) Control information

The interactive flow of conceptual information is controlled by control information which is partly invoked by the former. The experts responses (a) (b) (c) can be viewed as the evaluation results of the control information caused by understanding the conceptual information. this means that conceputal information implies the control information which specifies how the former should be processed or interpreted under which conditions and what should be done when the former is to be perceived. For (a) the control information specifies the activation of all concepts that are related to the key concepts in some way, for (b) it specifies the search of a term which satifies a given condition and for (c) it specifies the context to which all activated concepts in terms of a key concept must be subject.

The cognitive interaction between texts and expert can be actively supported by a system in that both sorts of information are directly or indirectly transferred to the system in the course of data base organization process. This triadic relation between texts, expert and system can be modelled as a **double production system** (Fig.1), in which an expert user plays a double role of production rules in LTM against the scanning texts and the stored knowledgte data base as well. However, with increasing amount of information transfer from the expert user to the system, this role of the expert user as LTM against the knowledge data base system is gradually changed to the role of STM. Then the user's control function of interactive process is taken over by the system. As a result the expert user plays an intermediate role for making decisions on long term purposeful representations of textual meanings following advices and suggestions from the system with the help of its control information.The transition degree of theuser's role from LTM to STM is continuous and goes down to the extent that the expert user only takes care of such tasks as selection of key terms or deep abstractions with regard to the future applications, those are too hard and time concuming for the system though not impossible.



(Fig 1) A double production system
Texts/User User/KDB

3.Frame-like Date base

There are many reasons why a frame-like structured data base seems to be appropriete for our purpose. They are partly identical to those which are explicitly stated by the proponent of frame theory (7) or the designers of frame-like representation languages. Following reasons are more or less specific for our case:

(1) Expert's knowledge about the computer industry and its developments are organized around the conceptual units which are more or less standardized.
E.g. "**Main-frame-makers**" is a concept for the expert that is used in conjunction with the computer-policy-making in Japan.

(2) Roles of conceptual objects are relatively clear and not much flexible in the target information space.
Eg. MBI is a rival of a national **main-frame-makers** and the national industrial Ministry **ITIM**, which specifies market strategic policies.

(3) Because of the reason (2) control information associated with each role of a conceptual unit is well formulated. E.g. If **MBI** announces a new product of **H-Series**, the expert expects the concept of **M-Series** of **main-frame-makers**.

(4) Those news and reports from a special world which are written for interested readers are partially .

(5) The frame-like representation is directly formulated to the form displayed in windows (s. sect.5) where the expert user interactively fills slots with key words or modifies descriptions.

Other reasons are based on our pragmatic attitude that we could eventually find an existing frame-like representation system which fits our special target world and approach; if not completely, we could use it by minor modification, or in the worst case we could design a new one by using existing system as a design guide line.

Our home-made frame-like representation language is called OBJTALK (8) which is characterized by the object-oriented idea of SMALLTALK (9). Frames called **objects** are organized into knowledge structures by message passings which activate procedural properties of the recipient object. In generalization heterarchies of objects, slot properties as well as procedural properties of super objects are inherited by subordinate objects. Following example illustrates instantiated concepts.

```
(concept main-frame-makers
      (individuates national-computer-industry
        (generic-properties
          produce: Compuer-main-frame
          members:Fujitsu,Hitachi,Mitsubishi,
                  NEC,Toshiba,Oki
          rival  : foreign-computer-industry
          supported-by: ITIM
        (method
       subvention: ? => (ask ITIM PD-budget: $!CPU)))

(concept MBI-JPN
    (instanciates fereign-computer-industry
    (generic-properties
      a-part-of  : MBI-WTC
      products   : ((CPU (E-series H-series))
                    (periphery (....))
                    (FS ))))
    (method
     (preis  : ? => (ask E-series preis : ?))
     (product: ? =>(ask window write: (ask self ??)))
```

OBJTALK offers a simple massage language of the form

```
(ask <object> <message>)
```

Any objects or the user can send a <message> to <object>. The above example of the instance **MBI-JPN** is created by sending to the object **foreign-computer-industry**.

```
(ask foreign-computer-industry
     make MBI-JPN with:
                <slot>.....
                <slot>....)

(concept freign-computer-industry
   (individuates computer-industry
   (generic-properties
     name : (calss string)
     . . .         . . . . . ))
```

By means of sending messages to a object we can create a new object, change it, activate actions of it, ask its properties or find an object which satisfies a given quantified condition.

It is interesting to note that, practically the expert user always tend to directly define concrete instances or individuals representing his knowledge, though frame-like languages such as OBJTALK are all designed to be used in a top-down way, ie. before he describes **MBI-JPN** as an instance of a concept, he is supposed to define the data type of **MBI-JPN**, ie. **foreign-computer-industry**. This top-down definition requested from the language structure doesn't necessarily correspond with the natural verbal behaviour of users, as it is stated in a general learning theory of languages. For bottom-up definitions of higher level concepts (super concepts) we need some methods for **generalization** and **abstraction** of individual or instance objects **intuitively** defined in terms of real data.

```
(ask <default class-name> generalize <instance-name>)
(ask <default class-name> abstract <individual>)
```

By generalizing a instance or abstracting an individual we can create a **class** of instances or individuals according to their perspectives, such that it prescribes property conditions to be satisfied by sample objects. This generalized class or abstract class must be modified by those other instances or individuals that belong to the same class.

Theoretical backgrounds for this bottom-up generalization or abstraction process are given by **learning by examples** (10) and **grammatical inference** (11).

Another practical problem we have in direct application of our object oriented language is that given a message of some properties it is a very time consuming task to identify an object with them in its inheritance hierarchy. Particularly, it is hard to find some procedural property in it, which is triggered by matching a pattern described as a prerequisite condition in it. At the moment a pattern matcher of OBJTALK does this task. Starting from the recipient object of a message the matcher searches through a inheritance hierarchy of the object a procedure called **method** which can be matched with the message body.

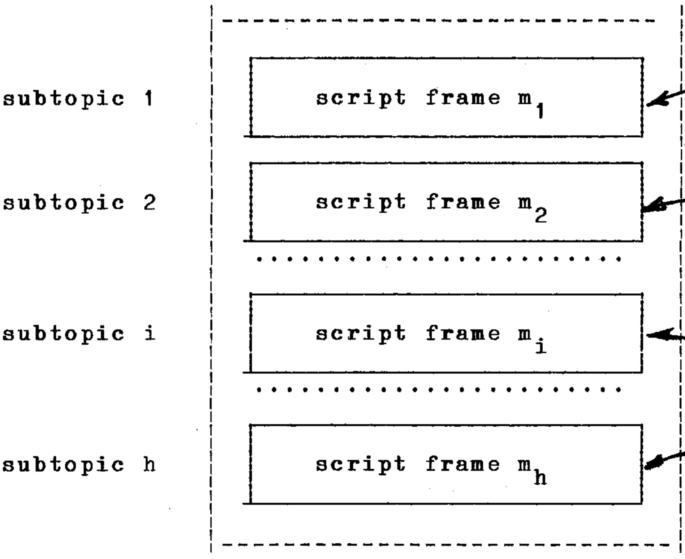
For simple desciptions of passive properties SRL (semantic epresentation language) of EXPLUS (5) avoided this time consuming problem of property inheritance by making an external bit-table which represents a herarchical property relations called semantic feature systems. For more complicated property descriptions such as procedural patterns we need a sort of global associative bit-map memory which mappes the procedural property-space onto the object space so that, given a procedural property by a message, those objects associated with this property can be retrieved very fast without any searches. This associative bit-map memory must dynamically store any new relations between objects and properties. Such a global associative memory contradicts the fram-like object-oriented representation principle which dictates the distribution of properties among objects. The philosophical justification of a pr perty/object associations are founded on the general theory of object-predicate reciprocity (12).

4.Hierarchical script frames for stereotypicality

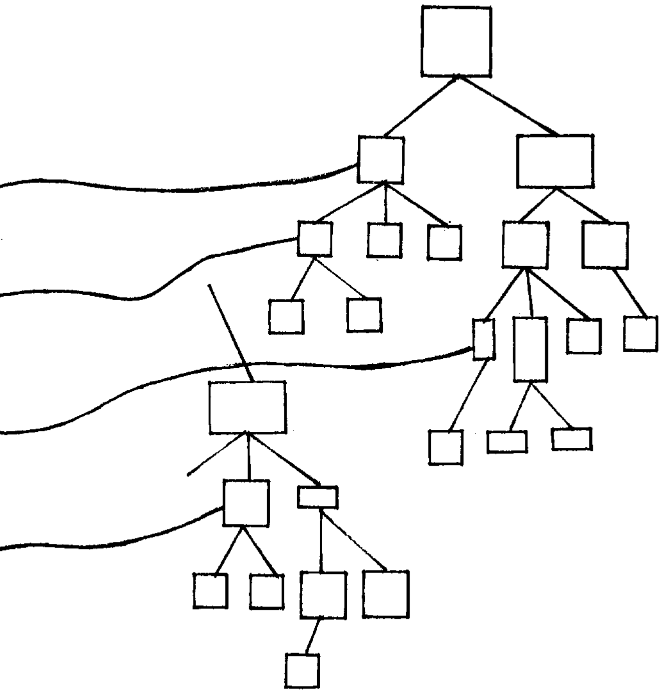
Our human experts roughly classify collected news and reports on the computer industry according to their topics,eg, **ITIM vs. MBI**, **ITIM** policy for information industry promotion, new product announcements from a computer industry, etc.

A main topic is assigned to each class which may be characterized by a set of predicates to be satisfied by members of the class. On this global class level our news world is stereotypical. However, if we try to represent many reports of the same class in a stereotypical form, we get a large sparse table but items are locally filled with various levels of details.(Fig.2). However, if we look into the contents of each report of a class under the surface of its main topic, we find on local levels some other subtopics which are not always consistently subordinate to the main topic.

This means first of all, there is no clear cut between classes, and each class cannot be completely formulated in a stereotypical form. Secondly, if we go down to the lower level of main topic descriptions ie. subtopics, we find that the stereotpicality is stronger within a subtopic. Each subtopic is described in different levels of details. We can come up with this situation by using hierarchical script frames (cf.(13)), where a subordinate script frames specify more detailed forms of the superordinate script frame. Those are connected to the latter in part-of or isa relations (Fig.3). By instantiating appropriate script frames the user can interactively organize them into a frame structure which fits a given report on the whole. The instantiation process can be supported by the system receiving message patters or key words from the user.



(Fig.2) Script frames of different details

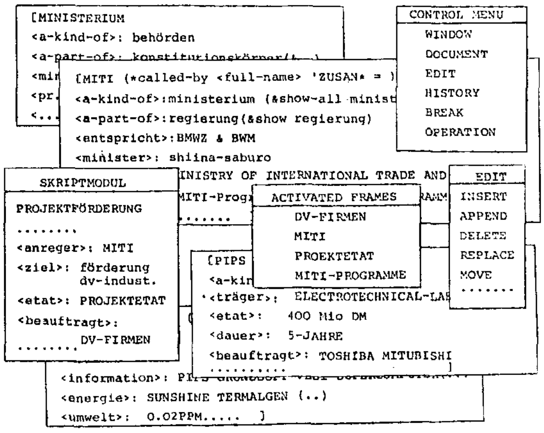


(Fig.3) Hierarchchalscript frames

5 User interface by a window system

A few number of AI research groups are using multiple window systems on a high resolution bit-map display such as ALTO (XEROX),the CADRs of LISP-machine(MIT), PARQ (Three River), etc. There are some cognitive-perceptive reasons for using a multiple window user interface. One of the most important reasons is that the user has a multiple (or parallel) contact with the machine, which offers him parallel decision alternatives as well as supports for short term memory during the course of interactive processes. In contrast with the act of scanning a newspaper through a pinhole the user can avoid back-trackings of dialog listings while keeping simultaneously various kinds of information each displayed in a window. This makes the user feel free from the labyrinth effects, being always aware of what the machine is doing and in which state of interaction he is. By reducing the labyrinth effects on the user, the machine offers him a better opportunity to plan his imput behaviour. The application of a window system to our interactive text processings adds more consistency of representations to the system.

A window may display a whole frame for the user to fill in some slots while one other window shows the other frame, whose name is just mentioned by the first one. Another one contains only names of those frames which are directly or indirectly mentioned in other frames. There are control windows which offer the user a set of control commandos and software tools such as editors (LISP editor, frame editor, kowledge editor), by which he controls the interactive process for embedding new information, revising the old one or creating new frame scripts, etc. (Fig.4)



```
[MINISTERIUM
<a-kind-of>: behörden
<a-part-of>: konstitutionskörper(...)
<mi
<pr
<...

[MITI (*called-by <full-name> 'ZUSAN* = )
<a-kind-of>:ministerium (&show-all minist
<a-part-of>:regierung(&show regierung)
<entspricht>:BMWZ & BWM
<minister>: shiina-saburo
INISTRY OF INTERNATIONAL TRADE AND
MITI-Prog                          RAMM
.......

PROJEKTFÖRDERUNG
........
<anreger>: MITI
<ziel>: förderung
        dv-indust.
<etat>: PROJEKTETAT
<beauftragt>:
        DV-FIRMEN
........

[PIPS
<a-ki
<träger>:   ELECTROTECHNICAL-LAE
<etat>:   400 Mio DM
<dauer>:  5-JAHRE
<beauftragt>: TOSHIBA MITUBISHI
..........                        ]
<information>: PI
<energie>: SUNSHINE TERMALGEN (..)
<umwelt>:  0.02PPM.....  ]
```

These itelligent machine responses to the user on the window levels are also controlled by the distributed control information embedded in frames. Some special frames such as **window-agent** and **frame-agent** take care of mapping between windows and internal representation of frames. We could view these agents as a two dimensional interactive syntax, which rules what kind of information should be given in which window, depending on its semantics. In this sense the interactive representation of meanings through a multiple window system can be viewed as more natural (ie. cognitively efficient) way to transfer the expert's knowledge to the machine. A user interface which disregards this cognitive aspects of user's man-machine communication behaviour leads to a bad interactive system which doesn't serve as an efficient expert's assistant. It is one of our goals to investigate the rules which underlies the expert's behaviour of representing his text understanding. Embedding these rules into a system, we could make the system intelligently reactive to the expert's interactive behaviour.

6 The State of the Implementation

In order to make the present task procedures of the human experts consistent to the machine support, we are following the expert's task process by using the same data and becoming ourselves a bit experts of our world. For standardization of terminologies to identify the corresponding concepts we bind together those terminologies in different reports which have the same meaning. This is the first step to understand the world in view of a machine but we find out this is not an easy task. Beside this, there are many contradictions and mistakes as it is well known to our experts.

Parallel to this sort of the world analysis, we now examine our frame-like language OBJTALK which is available in our computer at the moment beside FRL, to decide whether it gives us a sound base for our approach and if not exactly, what should be done to tailor the language for it. From our test example we experience what additional features we need (see section 3). Without having a high resolutation bit-map-display our implementation of the multiple window system is restricted to the display terminals available to our computer. Hoping to get such a terminal in the near future a multiple window system has been simulated on the HDS-terminals in LOGO (14).

7 Reference

(1)Brown,J.S. and Burton,R.R.,Bell,A.G. ; SOPHIE: A step towards a reactive learning environment, International journal of man machine studies, Vol.7,1975, pp 675-696

(2)Schank R.C. Abelson, P.R.;Scripts, Plans, Goals and Understanding, Lawrence Erlbaum Press, 1977

(3)DeJon,G.; Prediction and Substantiation: Process that comprise understanding, IJCAI,1979,Tokyo, pp217-222

(4)Bobrow,D.G.,Kaplan,R.M., Kay,M.Norman, D.A.Thompson,H.,Winograd,T.; GUS, A fram driven dialog system, Artificial Intelligence Vol.8,No1,1977

(5)Tanaka, H.; EXPLUS- A semantic processing system for japanese sentenses, Trans. IECE, Japan, '78/8 Vol.61-D No.8

(6)Bobrow,D.G.,Winograd,T.; An overview of KRL, a knowledge representation language, XEROX, Palo Alto Research Center,CSL-76-4,JUlY 4, 1976

(7)Minsky,M.,; A framework for representing knowledge, in The psychology of computer vision,by Winston P. (Ed),McGraw-Hill, New York, 1975

(8)Laubsch,J. ; OBJTALK, mmk-memo 12, Universitaet Stuttgart, Institut fuer Informatik, 1979.

(9)Ingalls,D.; The SMALLTALK-76 programming system: design and implementation, ACM SIGPLAN,Tucson, Arizona 1978.

(10)Winston,P.; Learning structural descriptions by exsamples,in The psychology of computer vision, by Winston,P.(Ed),McGraw-Hill, New York.1975.

(11)Feldman,J.; The first thought on grammaticak inference, Stanford AI memoNo.55, 1967

(12)Watanabe,M.S.; Knowing and Guessing, John Wiley & sons,Inc., New York,1969

(13)Cullingford,R.; Script application: computer understanding of news paper stories, Ph.D. Thesis, Yale University, New Haven, CT,1978

(14)Boecker,D.; Implementation of multiple window system on the HDS-terminals in LOGO, IfI-report, Inst.f. Informatik, Universaitaet Stuttgart, 1980

(15)Lorenz,K.; Die Rueckseite des Spiegels, Piper Verlag, Muenchen,1973

(16)Campbell,D.T.; Evolutionary epistemology, in P.A.Schilp (Ed), The philosophy of Karl Popper, Open court, Lasalle, 1974