

CUBUS—An Assistant for Fundamental Corporate Analysis

Michael F. Wolf, Dieter Wenger, and Klaus Kirchmayr

Granting credits to commercial customers is one of the central activities of a bank. Avoiding incorrect credit decisions and minimizing the required effort are crucial factors for the economic success of a bank. The basic task of credit granting is a *correct valuation* of the customer, which includes the analysis of his(her) annual account and the estimations of the customer's solvency. The bank is increasingly forced to deal with nonfirst-class debtors to keep and, perhaps, improve its market share. This situation leads to two main problems: (1) the increasing risk of loss because of the activities itself and (2) the increasing costs connected with the supervision of solvency to control the risks. With knowledge of these facts, it becomes evident that a bank is required to possess innovative means to check and supervise the solvency that allow an immediate and efficient realization of risks, that is, reduction in costs, in terms of solvency as well as the rational use of personnel.

Only a minor part of the knowledge that is necessary for the analysis and valuation of commercial customers is available in a well-documented form. Individual experience is of great importance considering the fact that there are only a few standard procedures and few generally ac-

cepted valuation guidelines. Therefore, the quality of the valuation depends on the abilities of the credit specialist.

The interpretation of an annual account requires analyzing the relations between many different items. A final judgment has to be based on the integrated consideration of all these relations, which often contain conflicting information. For these reasons, it takes a considerable amount of effort to even analyze the necessary information to get a first impression of a company. Too much time is wasted on the valuation of evident cases, and often, the complicated cases receive incomplete analysis. In addition, a request often has to pass more than one department in the bank; that is, the information available has to be analyzed and judged by several specialists. On the way from one department to the final department (general management), each credit specialist analyzes the balance sheet, gives his(her) opinion, and refines and adds the information that s/he believes is missing. For example, the specialist who has direct contact with the company contributes information on the quality of the management, and a specialist in the back office might add his(her) knowledge about the regional characteristics of the economic branches. A lot of time is wasted with these repeated analyses.

CUBUS Overview

The program CUBUS (computerunterstütztes Bonitötsuntersuchungssystem = computer-supported system for analyzing financial solvency) was designed to solve some of these difficulties and support the decision procedure in terms of decision quality and reduced cost. The system gives a general valuation and a first global impression of a company. It also draws the user's attention to the most critical items. In addition, a detailed analysis and valuations of each annual account item (that is, the position of the balance sheet and the profit and loss account) are provided.

The existing solvency analysis systems often provide (sometimes after elaborate calculations) general valuations of important abstract concepts such as solvency or capital structure, but the detailed valuations at a concrete level are missing. It is assumed that such systems will never be perfect, and the users always have to check the results provided by the system. Thus, this approach does not fulfill the requirements of practical decisions: A credit specialist usually does not think in terms of highly aggregated, mathematically calculated concepts, meaning that s/he cannot evaluate the results of such a system or say why his(her) own results might differ from those produced by the system. The user, however, knows a great deal about the quality and adequacy of concrete items in the annual account; for example, s/he knows

whether a company has too much stock or whether the proprietary capital is high enough in a specific case. If a system is to be transparent and meaningful for the user, results and valuations should be provided at this concrete level. Essential for the practical usability of the system is that it allows the user to overwrite specific single valuations and include this input in further automatic processes. In this way, the capabilities of the system and the user's additional knowledge and information can efficiently be combined.

Information about the specific economic branch is insufficient for the construction of norms that constitute the base of all valuation processes because it is the aspects of the company's individual annual account that must be studied. For example, norms for the liability side cannot be defined without knowing what the asset side looks like. All norms that only refer to the customer's branch are doomed to fail, which means that an individual norm (dynamic pattern) has to be defined for each company. This self-calibrating process is a critical success factor for the usability of the system across a variety of branches, regions, and countries. The dynamic pattern approach was a theoretical breakthrough that is now studied at universities. It contributes to the avoidance of branch-specific norms. The construction of such an individual norm (see Domain-Specific Task: Technology and Basic Algorithm for Solvency Analysis) requires so much knowledge of business economy and so much computation that it cannot be done manually in daily work. Even the theoretical model for the dynamic pattern generation could not be transferred from existing economic theories but could only be built iteratively with system prototypes. Before CUBUS was available, the norms based on branch data were intuitively modified by the credit specialists.

To be useful not only for an isolated one-shot analysis but for the whole institutional decision process, which concerns several organizational instances (departments), all the contributions of all the specialists involved have to be represented by the system and included in the automatic analysis. This approach means that a multiperson decision process is to be supported. The system is useful to experienced experts as well as less skilled staff. It is up to the user to decide whether s/he merely wants to obtain a rough overview of the most important elements or whether s/he seeks details that are of special interest to him(her). The handling of the system is easy, so that even persons who are unfamiliar with computers have no problems in using the program. The system provides the following functions: autonomous valuations, reactive assistance, active consulting, and interactive cooperation (see Human-Computer Cooperation). The system must apply to about 300 local Swiss Bank Corporation (SBC) branch offices and run in a com-

mercial computer environment. The facility to communicate with a host computer is important, especially the downloading of the customer's annual account data, which are already available on a central database.

CUBUS Architecture

In this section, we discuss the CUBUS architecture, analyzing the elements we used and the steps we followed in developing and deploying the system: knowledge representation, process control, the task concept, truth maintenance, fuzzy logic, the user interface, maintenance and modification.

Knowledge Representation

The technology applied can be described as a combination of a data-driven production rule and a frame approach. To avoid an accumulation of dissipated superficial heuristics with all the problems of extension and maintenance, as much knowledge as possible is represented as data: In a deep modeling approach, frames and other hierarchical relations are used to represent relations and dependencies between objects and attributes. Even the regulations for numeric operations are represented as attributes of the elements. This approach allows formulating general rules and applying object-oriented programming in an efficient way. New objects can easily be included. Even if they affect many procedures, changes are only required on the data level, which is how a high-flexibility system is reached, and changes can easily be made without having to rewrite the processing rules. This approach allows delegating modifications of the application to a system administrator.

Data-Driven Process Control as a Prerequisite for Flexibility and Actual Interactivity

Modifying existing and generating new information is done by agents (Spirgi, Probst, and Wenger 1990). Each agent knows when to be active because the conditions in the knowledge base that trigger it are part of its definition. Thus, the processes are controlled by the situation in the knowledge base. This approach allows designing applications in which different data constellations induce different processes. Even if new information is added by the user after the system has finished its valuation processes, these data can cause the activation of all the agents that are related to this kind of information. Therefore, new information is treated with the existing processes and can fully be integrated, with the results generated autonomously. This approach leads to actual cooper-

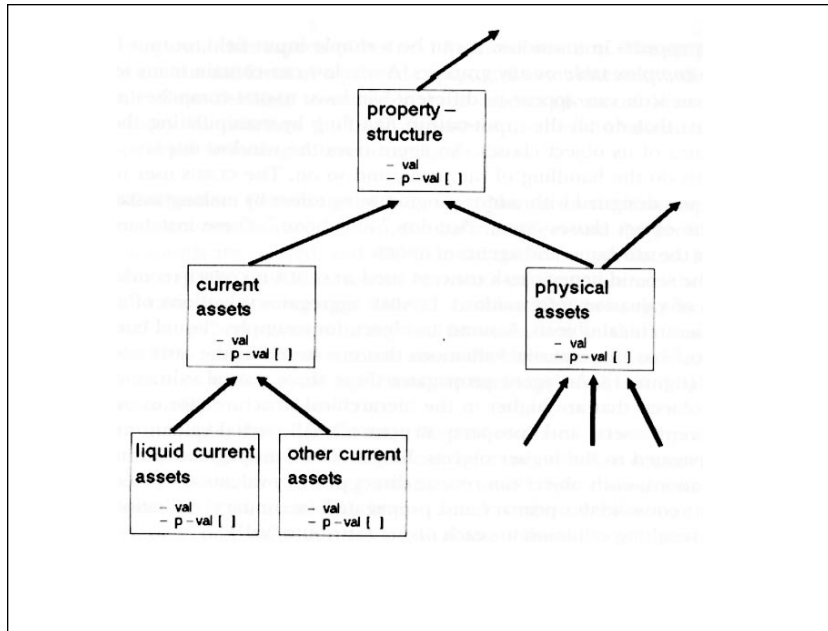


Figure 1. The Generic Task Concept CONDEV.

ation and interaction between user and computer: When the user gives his(her) input to the system, the system's results are affected, and the user's cognitions can be affected by the system results, which might lead to a new user input, and so on. In this sense, a real discussion with mutual influence takes place.

Generic Task Concept

The approach to formulate general agents that are appropriate for a variety of different situations consequently leads to the idea to identify tasks that have to be accomplished in almost all applications (Bylander and Chandrasekaran 1987) and to provide generic techniques (Spirgi, Probst, and Wenger 1991) that can be used when a new application has to be developed. Two of these generic task concepts, which were developed in a separate methodology project, are used in CUBUS: IO-GEN and CONDEV.

IO-GEN is a generic task concept for handling screen input and output information. It is a generic user interface. The most general object class is "screen unit." "Window" and "icon" are object classes that are specializations of "screen unit." The object class "view" consists of one or more windows. A window can appear in more than one view. An

icon appears in a window. It can be a simple input field, output field, or a complex table or any graphics. A window can contain many icons, and an icon can appear in different windows. IO-GEN comprises many agents that do all the input-output handling by manipulating the attributes of its object classes. An agent does the window displays, and agents do the handling of the icons, and so on. The CUBUS user interface was designed with minor programming effort by making instances of the object classes “view,” “window,” and “icon.” These instances inherit the attributes and agents of IO-GEN.

The second generic task concept used in CUBUS is *CONDEV* (condensation of valuation information). *CONDEV* aggregates valuations of a net of hierarchical objects. Assume an object, for example, “liquid current assets,” has three partial valuations that are stored in the attribute “p-val” (figure 1). An agent propagates these three partial valuations to the objects that are higher in the hierarchical structure (for example, “current assets” and “property structure”). All partial valuations are propagated to the higher objects. Besides these propagated (indirect) valuations, each object can receive direct primary valuations. A second agent consolidates primary and propagated (secondary) valuations to one resulting valuation for each object (attribute “val”).

Truth Maintenance

To guarantee consistent valuations after adding, modifying, or deleting information, the implementation of a truth maintenance mechanism was necessary. The valuations that are no longer valid are retracted, and new valuations are asserted. *CONDEV* is used to consolidate partial valuations and propagate them to more abstract concepts wherever information changes.

Fuzzy Logic

With each additional partial valuation, the resulting valuation might be modified, and its certainty will increase or decrease according to the level of conformity. A simplified fuzzy logic approach was developed for managing the certainty of the system valuations. This approach allows working with missing and contradictory information (see the Aggregation of Valuations subsection that follows).

Domain-Specific Task: Technology and Basic Algorithms for Solvency Analysis

The application was built by applying *CONDEV* and IO-GEN and several domain-specific technologies. Input for the CUBUS analyses is the customer’s annual account data. Output is the valuation of the customer’s

solvency and other aspects of his(her) financial situation. The output is presented in the form of a printed report and as an interactive display, which allows the user to consider those aspects of special interest to him(her). We overview the basic algorithms used in the valuation processes in the following subsections (figure 2).

Self-Calibration: Generation of Individual Norms. In the first step, the annual account data are read in, the balance sheet and the profit and loss accounts are created, and the financial ratios are calculated. In the second step, the individual norms (dynamic pattern) that are the base for all valuation processes are defined. In this self-calibrating process, a *normal*, or expected, value is defined for each item in the annual account. Here, implicit assumptions about the branch are taken into consideration as well as some aspects of the company's individual balance sheet as it is.

For example, the financial ratio "capital rotation" is one input that calculates what cash-flow margin can be expected: A company with high capital rotation (trading company) usually has a lower cash-flow margin than a company with lower capital rotation (production company), or a company with many physical assets should have high depreciations. The concrete procedures required for the calculation of each norm are complex because a great number of general parameters have to be linked with many case-specific facts (too numerous to describe here). In this way, a second balance sheet and profit and loss account are defined as an individual norm. In further steps of the analysis procedure, the actual annual account is contrasted with this norm. The knowledge of how to define norms for the single items is not available in published scientific papers and had to be developed by specialists at our bank. The existence of norms for all annual account items allows the definition of valuation scales for ratios that consist of such items. When ratio z ($r-z$) is defined as $r-z = \text{item-a}/\text{item-b}$, then the norm of $r-z$ is defined as $\text{norm-}r-z = \text{norm-item-a}/\text{norm-item-b}$. In this way, for each financial ratio, i , a value, $\text{norm-}r-i$, is calculated that is considered normal (not good, not bad), and a valuation scale is defined with $\text{norm-}r-i$ as middle or normal, $\text{norm-}r-i + x$ as good, $\text{norm-}r-i - y$ as bad, and so on.

Credit and Blame Assignment: Passing on the Valuations. On this basis, the company's actual financial ratios can be valued on a scale of "very bad" to "very good." The next part of this step (figure 3) is the identification of those components of each ratio that make the ratio either too high or too low (for example, when the ratio liquidity I is too low, either the item "extended liquid assets" is too low, or the "current liabilities" are too high). The ratio's valuations are passed on to those

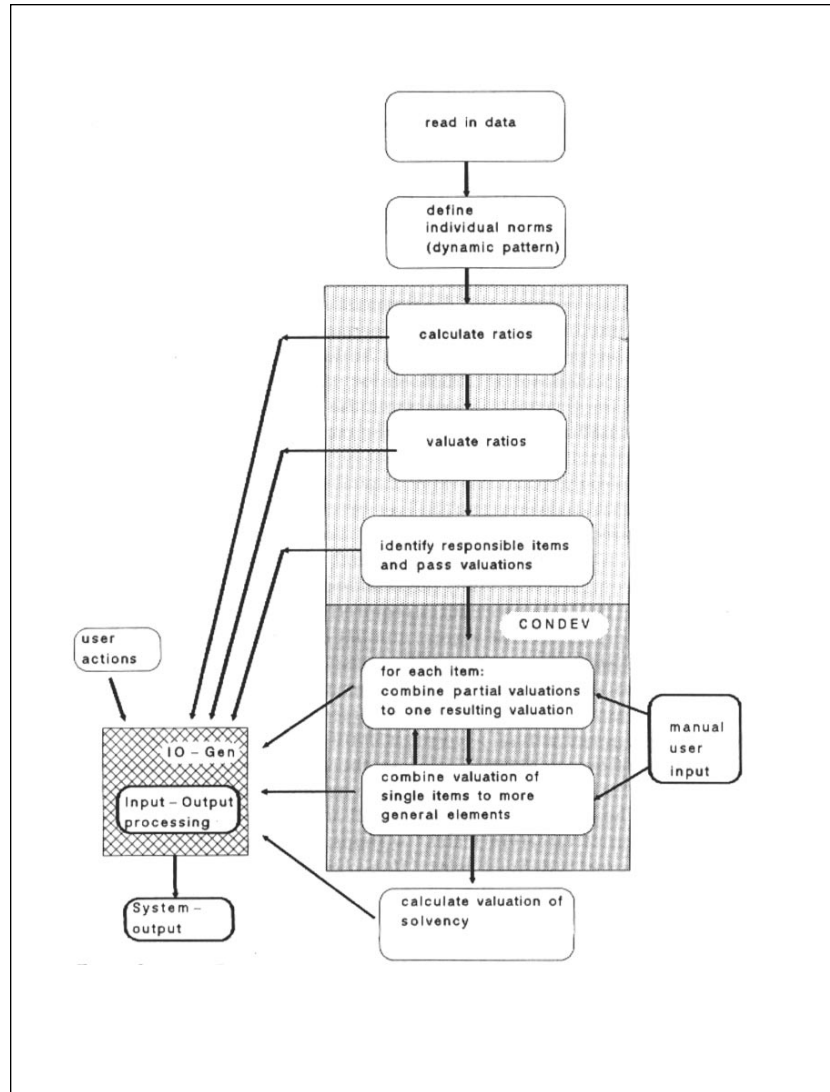


Figure 2. CUBUS Functions.

elements with the most significant deviation from the anticipated values, which are represented in the dynamic pattern. If such a component (for example, “extended liquid assets”) itself is an aggregation of further subcomponents (“liquid assets” plus “security holdings”), the valuation is passed on until an atomic target element is found.

Aggregation of Valuations (CONDEV). Each element in the annual ac-

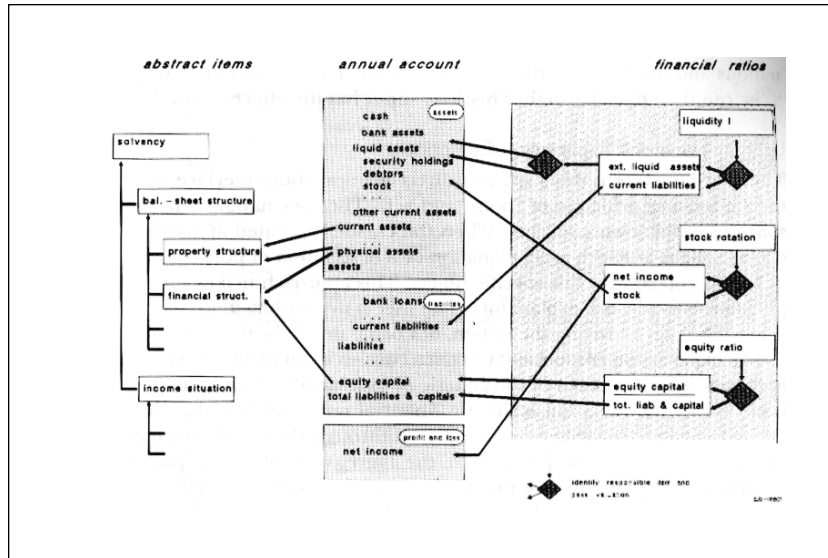


Figure 3. Valuation Model of CUBUS.

count can receive values directly from one or more ratios and indirectly from other elements. Therefore, the final valuation of a component usually has to be combined by several conflicting partial valuations. More than one conforming partial valuation increases the certainty of the resulting valuation, whereas conflicting valuations result in lower certainties. The same principles are applied when the valuations of the basic elements are aggregated to the valuation of more abstract concepts (for example, the valuations of current assets and physical assets are aggregated to the abstract concept "property structure"). The applied algorithm is CONDEV.

Different Views and Graphically Oriented Simulation

In addition to the valuation of the company's current situation, an expectation for the following year is estimated. For each item, the pure system valuations and the combination of manual and system valuations can simultaneously be displayed. Furthermore, the user can switch between two viewpoints at any time: with or without consideration of hidden reserves. All system results are separately calculated by taking these two possibilities into account. Graphically oriented simulation is also provided. The user can simulate changes in the interest rate, the net income, or other factors and obtain a graphic display of the effects on the income situation of the company (what-if analysis). Additionally, s/he can simulate the manual modification of system val-

uations and analyze how the valuation of other concepts will change. S/he can then decide whether his(her) input has the effects desired.

User Interface and Explanation Component

The user interface is designed as a direct manipulation interface, with the underlying principle of "point and act." The user navigates in the space of well-known concepts. When the cursor is pointed at a screen object, a help line with an explanation or a description of possible ways of action appears for this specific object. This approach makes the application basically self-explanatory. The user is never obliged to do anything. The user is driving the system, not being driven by the system.

The explanation component of CUBUS consists of an elaborate graphic representation of each of the generic or task-specific techniques. Figure 4 shows the valuation scale of a financial ratio and how the static (s) and dynamic (d) valuations are passed through the net of concepts to the items responsible. In figure 5, the aggregation of several partial valuations into one resulting valuation of the item's current liabilities is shown. The aggregation of valuations in the concept net is presented in figure 6 (figures 5 and 6 refer to CONDEV). A comparison between an expected balance sheet (narrow columns) and the actual situation (wide columns) is displayed in figure 7. The user can call these explanations for each item any time s/he wants, which is how an interactive, user-driven explanation component is realized.

Maintenance and Modification

The principle of representing as much information as possible in the data and formulating general rules allows making even fundamental changes (for example, including a new financial ratio, changing the net of concepts, including new concepts, modifying central parameters) without having to modify the rules. The variables that are subject to change are stored in a single file that can easily be edited and modified by a system administrator. Therefore, maintenance and modification can easily be done. The existence of a self-adapting dynamic pattern reduces the necessity of such modifications to a minimum.

Innovations

This section analyzes the innovative elements of the CUBUS system. These innovations include self-calibrating dynamic norms, support of multiperson decision processes, human-computer cooperation, a system development methodology, structured analysis and valuation procedures, and new knowledge management dimensions.

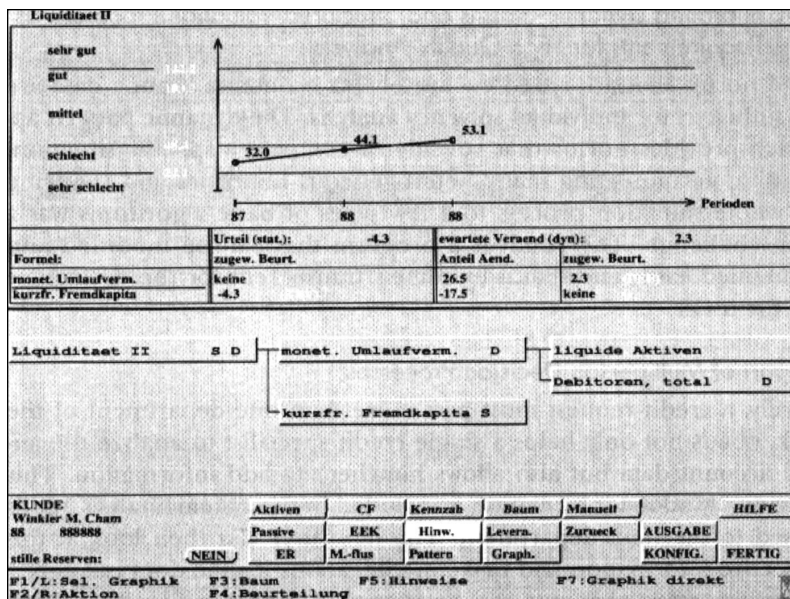


Figure 4. Valuation Scale of a Financial Ratio.

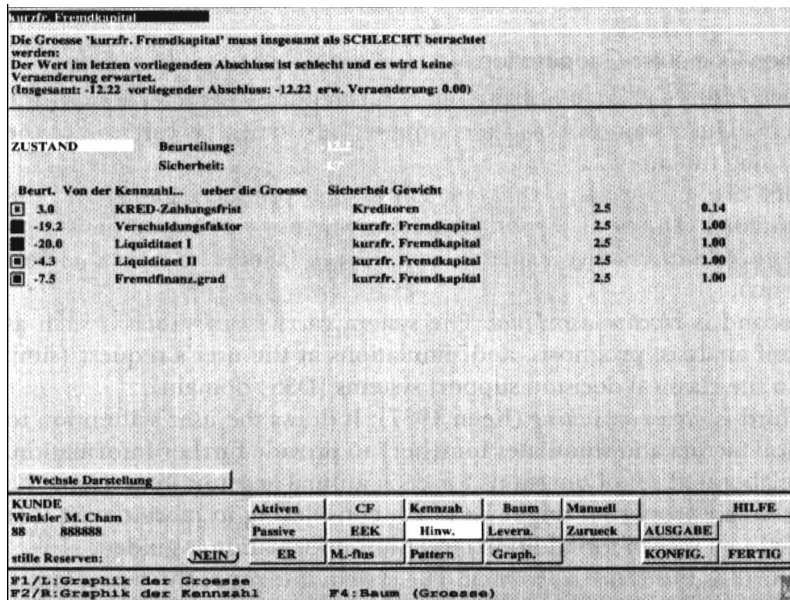


Figure 5. Aggregation of Several Partial Valuations to One Resulting Valuation of the Item Current Liabilities.

Self-Calibrating Dynamic Norms and Structured Valuation Processes as Basic Requirements for High-Quality Analysis

Fixed norms that are based exclusively on economic branch data are not sufficient for individual solvency analysis. The dynamic pattern approach provides norms that consider a company's specific situation. Avoiding accumulating many heterogeneous heuristics and reducing the whole valuation process to a few types of basic algorithms was a great challenge. The result is a program that can be modified and maintained easily and that is extremely transparent for the developers and the users.

Support of Multiperson Decision Processes

Usually, a credit request must pass more than one department of the bank. CUBUS not only helps a single credit specialist to analyze the annual account data but also allows him(her) to add information. This individually added information is included in the system analysis and is passed to the next department. The next user can then look at this subjective information and make modifications if s/he does not agree. The overwritten elements can be viewed in a history file. This file is how the ultimate user possesses the system valuation as well as the information provided by all previous users.

Human-Computer Cooperation

Flexible interaction is one of the most outstanding characteristics of the system. The system works as a competent assistant that carries out the following functions:

First are *autonomous valuations*: The system produces global and differentiated valuations without requiring any user intervention and offers flexible interactive explanatory facilities (similar to typical expert systems).

Second is *reactive assistance*: The system carries out subtasks such as what-if analysis, prognosis, and simulations at the user's request (similar to the classical decision support systems [DSS] domain).

Third is *active consulting* (Keen 1987): It draws the user's attention to critical factors and stimulates him(her) to provide further information. An elaborated set of questions for each annual account item shows the user which properties might be worth considering in more detail; that is, his(her) search for additional relevant information is guided.

Fourth is *interactive cooperation*: The system integrates the user's input, modifies its own results accordingly, and gives feedback about the implication of this input for the system's valuations. This new constellation might provide new insights that can result in new user activities, for ex-

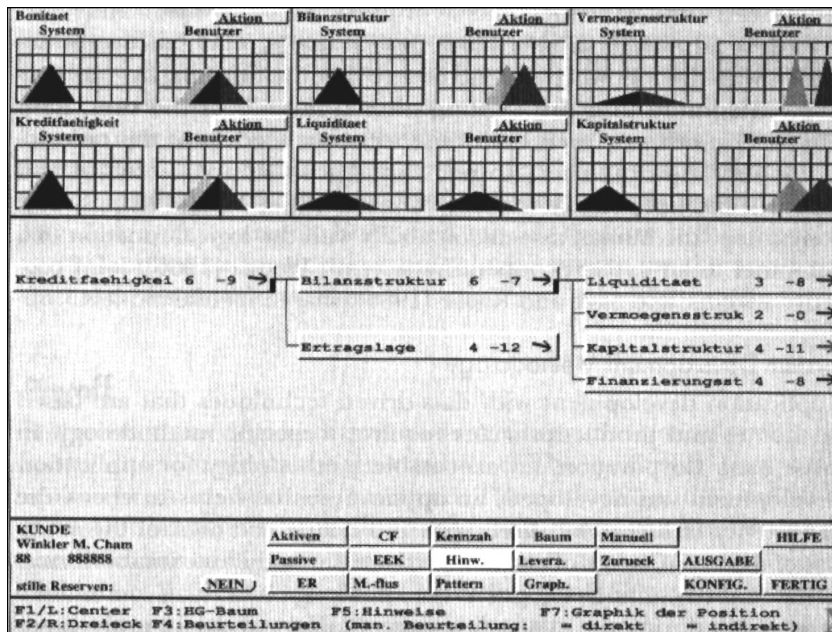


Figure 6. The Aggregation of Valuations in the Concept-Net.

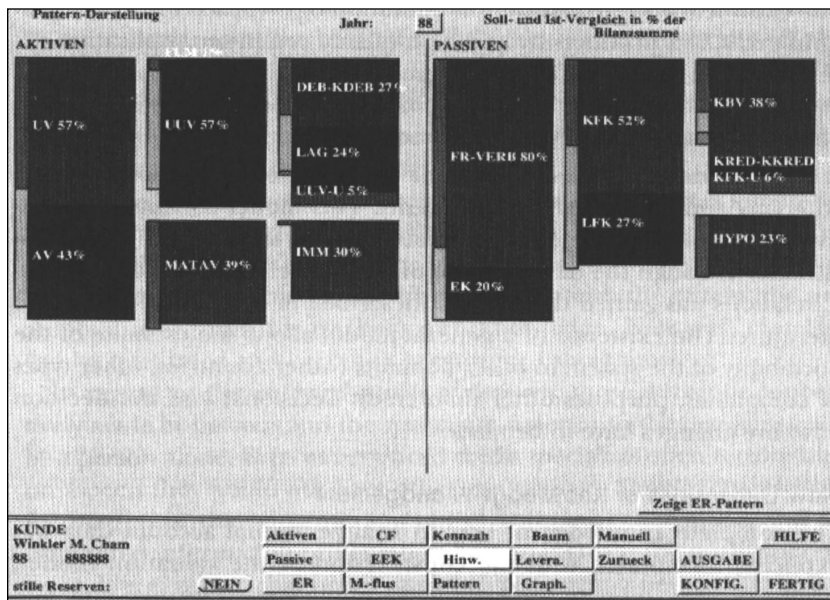


Figure 7. A Comparison Between a Norm Balance-Sheet and the Actual Situation.

ample, carrying out further analysis or providing more information.

Fifth is *presentation of information and knowledge*. This function is designed with the following aims: to ensure that the information can easily be integrated into the existing mental models of the user, help him(her) modify or even build his(her) own models of the task domain, and allow him(her) to reach analyses that are profound and complex and not possible without the system. (See Niva [1986, p. 336]: combining "the human association ability with the logical function of a computer as an explicit cooperative system"; Woods [1986]: joint cognitive systems; and Sage and Rouse [1986]: shared mental models.)

System Development Methodology

Application development with data-driven techniques that are based on frames and production rules requires a specific methodology. In Swiss Bank Corporation, an executable methodology for application development was developed, an application that helps to check the consistency of hierarchical relations and define and control the activities of agents in terms of activating constraint and contextual information and the resulting changes in the knowledge base. This methodology was built during the CUBUS development. Now, about 10 projects are being developed with the help of this methodology. The integration of our experience in an executable methodology will lead to a tremendous growth in productivity and high-quality computer application development at Swiss Bank Corporation.

Structured Analysis and Valuation Procedures

CUBUS provides a coherent model for the valuation of the solvency of a company and all annual account items. This model consists of a relatively small number of generic and task-specific techniques (basic algorithms). Through the development of this model, insight into economic relations was gained that was not described in such detail in specific literature. The existence of a general model allows the estimate of the portability of the system to other domains (other countries, other types of companies, purposes other than credit decisions) and the decision of where changes have to be made.

New Dimensions of Knowledge Management

The complete knowledge of how to analyze annual account data was explicitly formulated with the construction of the valuation model. Until this time, most of the knowledge applied was personal and subjective. Because the knowledge is now transparent, it can be controlled and criticized by expert committees and can be developed further.

Significant improvements have been achieved this way. New knowledge can rapidly be delivered with new versions of the system, and it is automatically applied. There is no need for the credit specialists to read a lot of papers and change their daily working procedures. Instead, they use the new CUBUS version, and the new norms or valuation procedures are applied. By controlling, improving, deploying, and implementing new knowledge, new dimensions are brought to an organization's knowledge management. Now, even models that are highly complex can be applied because the computer knows how to use them and produces the result within seconds.

Payoff

Since CUBUS was deployed (and as done during the testing periods), a systematic evaluation has been conducted: The quality and acceptance of the system was intensely examined. The profit will also be estimated in collaboration with our controlling division. The numeric quantification of the payoff is difficult because a lot of aspects are qualitative, and many of the financial aspects can only be assessed in several years when the credits have to be paid back.

In regard to *efficiency*, tests showed that the time necessary for analyzing the annual account of a company can significantly be reduced. Today, CUBUS is used by the analysts of 8 local branches. When all SBC credit specialists at all 20 branches can profit from CUBUS (which will be required in 1992), personnel costs will be reduced by approximately US\$500,000. The capacity gains will be used for improved customer consultancy and thorough analyses of difficult cases.

In regard to the *increase in the know-how level*, the system's knowledge and analytic capabilities are equivalent to the level of experienced experts. These facilities are now included in the work of less skilled persons when they use the system. When working with the system, the user automatically improves his(her) own knowledge. New staff members can be integrated and become operative in a short time.

In regard to the *improved quality of decisions*, those decisions cooperatively made by persons and the system are superior to the decisions made by a person alone. Even experienced credit specialists often receive hints on aspects they would not have otherwise noticed. Simulations, facilities for prognosis, the immense amount of complex calculations, and the elaborated information presentation help to increase the quality of decisions. It is expected that better decisions concerning new credits and an automatic check of the financial situation of all customers with running credits will help to reduce bank loss by US\$1 million each year.

In regard to the *improved understanding of the problem field*, the experts' knowledge of the problem domain has largely been improved by the development of the application. Because this knowledge was structured and made operative in the system, the generation of new approaches and theories (see dynamic pattern) has been possible.

Finally, CUBUS is a management instrument that helps to establish management decisions and guidelines. The system provides standards for fundamental corporate analysis that are automatically applied when the system is used. In contrast to written instructions in manuals, these standards cannot be misinterpreted or ignored. New guidelines and techniques can effectively be introduced through its tutoring facilities.

At least US\$1.5 million are expected to be saved each year. In this way, the development costs (less than US\$2 million) will be paid back within 1½ years. This calculation does not consider that along with the system development, by-products such as IO-GEN and CONDEV were created and can be used in many other projects. Furthermore, CUBUS versions will be created for special purposes. CUBUS offers completely new possibilities for estimating the solvency structure of the bank debtors that are of growing interest in the asset and liability management of SBC.

Development and Deployment Timeline

CUBUS is now implemented on IBM PS2/70 workstations. The development and deployment time line for CUBUS is as follows:

1987

March: Joint project of German Dresdner Bank, Spanish Banco de Bilbao, and Swiss Bank Corporation started to develop a system for fundamental corporate analysis.

May: Intensive interviews conducted with credit experts. First prototype realized with Prolog by an external firm.

June: Swiss Bank Corporation buys TI Explorer II with Inference Corporation's ART and starts its own development.

October: First ART prototype based on common rules of thumb (for example, IF long-term assets $\geq 1,2 * (\text{net worth} + \text{long-term external fund})$ AND hidden reserves in fixed assets > 0 THEN the financial structure is ok).

1988

January: Comparative evaluation conducted of the Prolog and ART prototypes. Decision made to further develop the ART prototype as a single Swiss Bank Corporation project. First ideas developed for a deep modeling approach instead of rules of thumb.

July: First deep model prototype finished. Extensive testing started.

September: Different deployment platforms evaluated. Tests on MicroExplorer conducted, and porting to the host under IBM Knowledge Tool started.

1989

March: Development of the prototype on the Explorer continued. Bad experience in search for a convenient deployment platform leads to the evaluation of the personal computer-based ART-IM.

May: ART-IM infrastructure built. Porting the ART-Lisp prototype to ART-IM with integrated further development started.

July: Two systems compared. Decision made to stop development on the TI Explorers.

September: Alpha version operational and installed at a Swiss Bank Corporation branch. Improvements made at the user interface and data transfer from host to personal computer based on the feedback of about 10 users. Further modifications initiated.

1990

January: Beta version operational and released at 3 Swiss Bank Corporation branches. Regular user training started.

May: The dynamic pattern approach makes CUBUS usable for all branches and many countries.

September: Initial production version 3.7 operational. System installation process at the 18 regional Swiss Bank Corporation branches started.

November: Production version 3.9 deployed.

References

- Bylander, T., and Chandrasekaran, B. 1987. Generic Tasks for Knowledge-Based Reasoning: The "Right" Level of Abstraction for Knowledge Acquisition. *International Journal of Man-Machine Studies* 26:231–243.
- Keen, P. G. W. 1987. Decision Support Systems: The Next Decade. *Decision Support Systems* 3: 253–265.
- Niva, K. 1986. A Knowledge-Based Human-Computer Cooperative System for Ill-Structured Management Domains. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-16(3): 335–342.
- Sage, A. P., and Rouse, W. B. 1986. Aiding the Human Decision Maker through the Knowledge-Based Sciences. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-16(4): 511–521.
- Spirgi, S.; Probst, A. R.; and Wenger, D. 1991. *Generic Techniques in EMA: A Model-Based Approach*. Berlin: Springer-Verlag.
- Spirgi, S.; Probst, A. R.; and Wenger, D. 1990. Knowledge Acquisition in a Development Methodology for Knowledge-Based Applications. In

Proceedings of the First Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop, eds. H. Motoda, R. Mizoguchi, J. Boose, and B. Gaines, 382–397. Tokyo: Ohmsha.

Woods, D. D. 1986. Paradigms for Intelligent Decision Support. In *Intelligent Decision Support in Process Environments*, eds. E. Hollnagel, G. Mancini, and D. D. Woods, 153–174. Berlin: Springer.