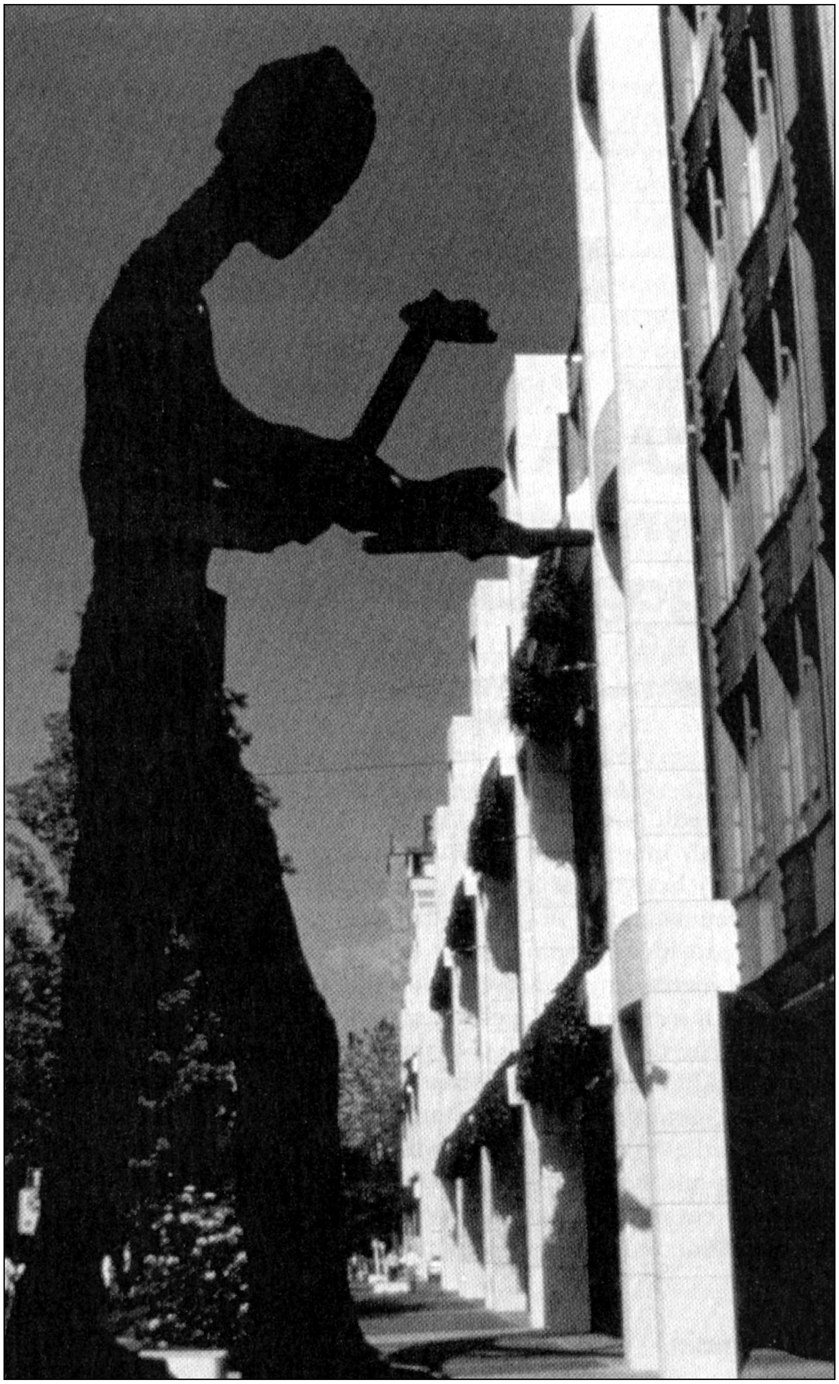
MOCCA: A Set of Instruments to Support Mortgage Credit Granting

Steve Hottiger and Dieter Wenger, Swiss Bank Corporation

Mortgage credit granting has to be supported by task-oriented instruments (highly interactive support tools that enable cooperative problem solving) because of increasing competition and unfriendly economic circumstances. MOCCA (mortgage controlling and consulting assistant) provides instruments for decision support, customer consultancy, and management. Most of the underlying models in MOCCA are innovative in terms of being either completely new or operative for the first time. The implementation of the application is based on a combination of different knowledge-based techniques, such as agent-based processing, generic application building blocks, fuzzy measure, data-driven paradigm, and a two-layer windowing system. MOCCA (mortgage controlling and consulting assistant) is now being used in the major branch offices of Swiss Bank Corporation (SBC). The development of the application required two person-years.

Task Domain

Mortgage credit granting is a central activity in the Swiss banking business. More than $35 billion or nearly one-third of all SBC assets are



covered directly or indirectly by real estate. Based on decades of high real gross national product and a restricted amount of real estate set against growing demand, investments in property were considered nearly risk free for a long time. In the last couple of years, however, the economic stagnation has affected the real estate market. Traditional models and techniques are no longer able to provide support in defending against growing risks, such as the interest rate risk and business and customer risks, or competing better given the unfriendly economic circumstances.

The interest rate risk became important when mortgages could no longer be refinanced by favorable long-term savings. This situation led to more and more money being absorbed on the interest volatile and expensive markets. As a consequence, mortgage rates had to be adjusted several times, which caused solvency difficulties for more customers. Because no operative models were available for the prediction of credit worthiness, no workable management policy could be developed. There was also no cost-calculation instrument to separate the profitable deals from the others. Recently, the decreasing demand resulted in a remarkable reduction in prices, causing anxiety among some local financial institutes engaged mainly in mortgage lending. The value of the real estate had been estimated too high; therefore, the invested money was no longer fully covered. To complicate matters, these valuations were often done by external architects whose judgments couldn't be checked sufficiently.

Global banks such as SBC have to compete on the mortgage market with a range of specialized financial institutions that are able to provide more attractive mortgage rates and services. Moreover, today's customers are used to having connections to more than one financial firm, depending on the service and the security provided. Therefore, like any other company, a bank is forced to offer attractive and competent consulting services, including areas that are not traditional for a bank, such as tax and budget consultancy. A global bank has the advantage of a large variety of products compared with more specialized competitors. By promoting cross-selling at each point of sale, this advantage can be transformed into higher profits. Acquiring knowledge of the variety and complexity of the offered products is a great challenge to a customer consultant. With the complexity of the products growing, it is nearly impossible for him/her to offer all the products without the support of a computer, especially if he/she cannot invest more time in the consultation. The supporting tools must be graphically attractive, self-explanatory, readily understandable, and flexible in any situation. They have to motivate the user to explore the underlying models. Furthermore, they have to be efficiently adaptable to new banking products.

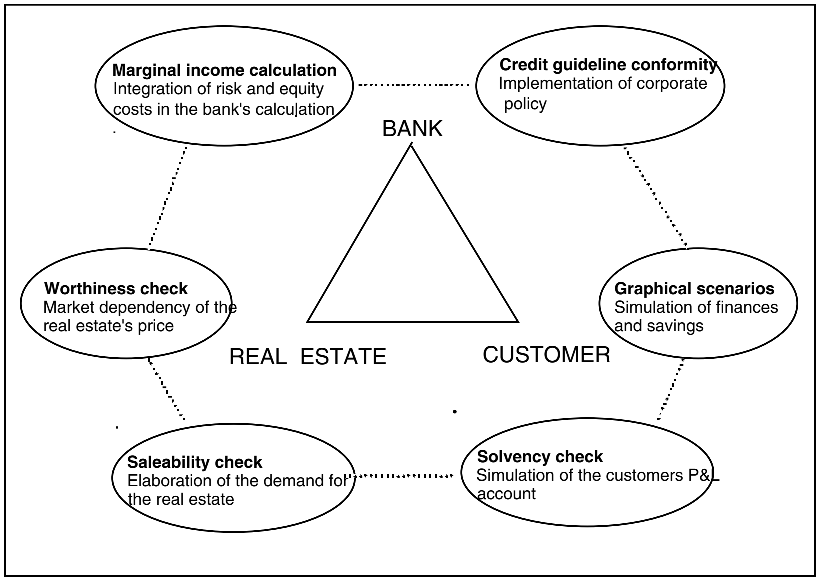


Figure 1. Overview of MOCCA.

The use of knowledge-based techniques promised several advantages compared with more traditional approaches in the modeling of the complex, incomplete, and heterogeneous task domain. Today, this assumption has been confirmed completely. In the following sections, we introduce MOCCA and elaborate on the combinations of knowledge-based techniques on which the application is built.

Overview of MOCCA

MOCCA broadly covers the areas of decision support, risk control, customer consultancy, and management support. It appears as a homogeneous application based on a set of heterogeneous instruments.

MOCCA is decision oriented. It is based on a model that collects, relates, and evaluates all components of the decision process based on their importance. MOCCA guarantees that depending on the complexity of the case, the most critical and economically important elements are always examined, elements such as a positive marginal income for the bank, degree of conformity with internal credit guidelines, customer solvency, worthiness of the real estate, and salability (figure 1).

Besides the decision process, MOCCA also models the mortgage business in various dimensions (figure 2). Each dimension has different levels of abstraction that require different methods and information.

They are represented by an object hierarchy in which the different processes can be attached according to the level of abstraction.

MOCCA is not an expert system in the sense of imitating the expert's problem-solving behavior. It is a system for cooperative problem solving. The occasion to develop a new system has allowed SBC to implement models based on theory as well as knowledge from other banking sectors, particularly if they provide better information and are acceptable to the expert (see the following section). The intention behind MOCCA is to shape the future, not rebuild the past.

The implementation of the selected models was driven by the requirements of flexibility, self-explanation, and easy handling. Thus, the expert was motivated to explore the models, test their usefulness, and develop decision-oriented measures for the evaluation. If no existing models were available, they had to be developed in an iterative prototyping process in close cooperation with the expert. Examples of these models include solvency check and salability investigation.

The set of customer consulting instruments is closely attached to the decision models. In this way, MOCCA encourages the user to work with the customer and analyze his/her financial background to get better decision quality. In the following section, we present some of the economic innovations of MOCCA.

Economic Innovations in MOCCA

With MOCCA has come some economic innovations, including task-oriented application, the ability to make theory operational, and the use of interactive simulation instruments. These innovations are discussed in the following subsections.

Task-Oriented Application

MOCCA is completely task-oriented. It is used as an intelligent assistant at different levels of the mortgage credit-granting process (figure 3).

MOCCA allows the different levels (management and front) to work closely together, primarily because it facilitates communication.

The architecture of MOCCA is multidimensional, based on a system-oriented approach where several subsystems are integrated into one system. The user is driven by the critical points of the current case and not by the structure of the user interface. To carry through each manipulation automatically, MOCCA makes use of the data-driven paradigm.

MOCCA is strongly decision oriented. The evaluations of all aspects are related to a scale. In this way, deals become comparable, and the

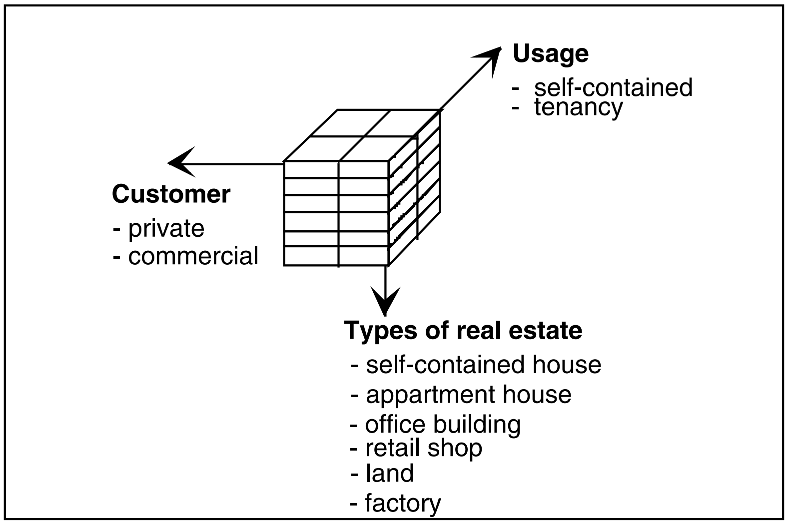


Figure 2. Dimensions of the Mortgage Business.

results are prepared for classification. In addition, MOCCA supports multiperson decisions. It enables the user to add information in an informal way when he/she wants to change an evaluation of the system.

Theory Becomes Operative

There are several ways in which MOCCA has turned theory into operation. The property-worthiness analysis is based on two competitive models: the Lageklassenmodell of Nägeli (1980) and the Strukturwertmodell of Fierz (1987). Although both were regarded as valuable, they were never used because of the large effort required to perform one single evaluation. In MOCCA, the models are implemented efficiently and in a flexible way.

Now, the strengths and weaknesses of these models have been explored widely.

The model for the marginal income calculation was developed in another banking department. With the implementation in MOCCA, it was possible to develop decision-oriented standards for the evaluation of the results. Interactive simulation instruments proved to be an excellent medium to bring different experts together for the elaboration of such standards. Now that the model is decision oriented, it is really operative.

Interactive Simulation Instruments Replace Rules of Thumb

The solvency check for private customers was based on a simple rule of

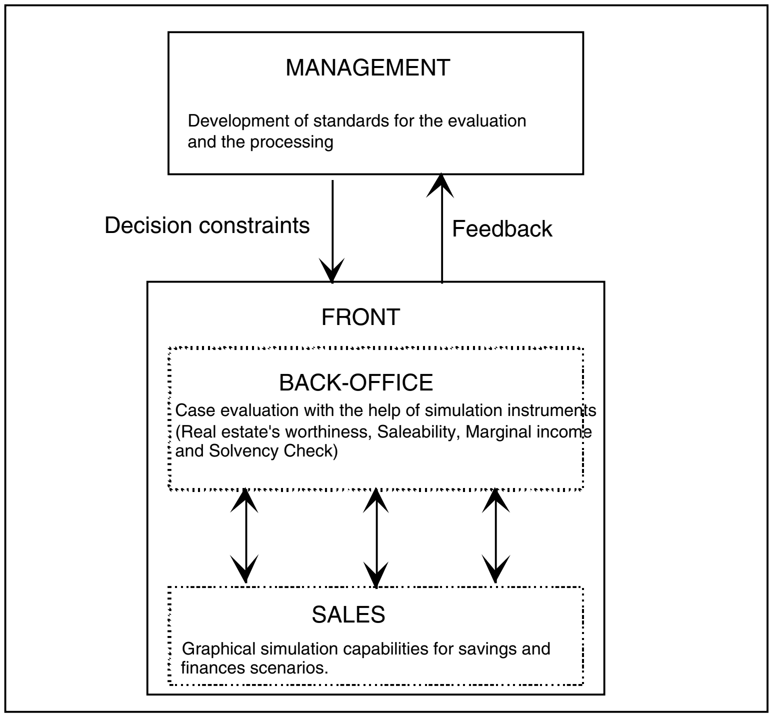


Figure 3. Levels of Mortgage Credit Granting.

thumb: If the costs for mortgage interest, repayment, and maintenance exceed one-third of the income, then solvency is considered critical. It is obvious that for higher incomes, this rule is too restrictive, whereas for lower incomes, it is not restrictive enough. The new solvency check in MOCCA is the result of an intensive prototyping process conducted during the first few months of the project. Expert and developer met once every week for at least two months until the credit section was satisfied with the new model. The evaluation is now based on the balance between income after mortgage interest, repayment of principal, maintenance and utilities, other interest and taxes, and a family budget that guarantees a reasonable standard of living (figure 4). The budget can be customized separately, as is the tax calculation. Although a larger base of information must be examined, the model is easy to understand and, therefore, is much better accepted than the initial rule of thumb.

New Decision Parameters Reduce Risk

The salability investigation is crucial for the credit decision because it is

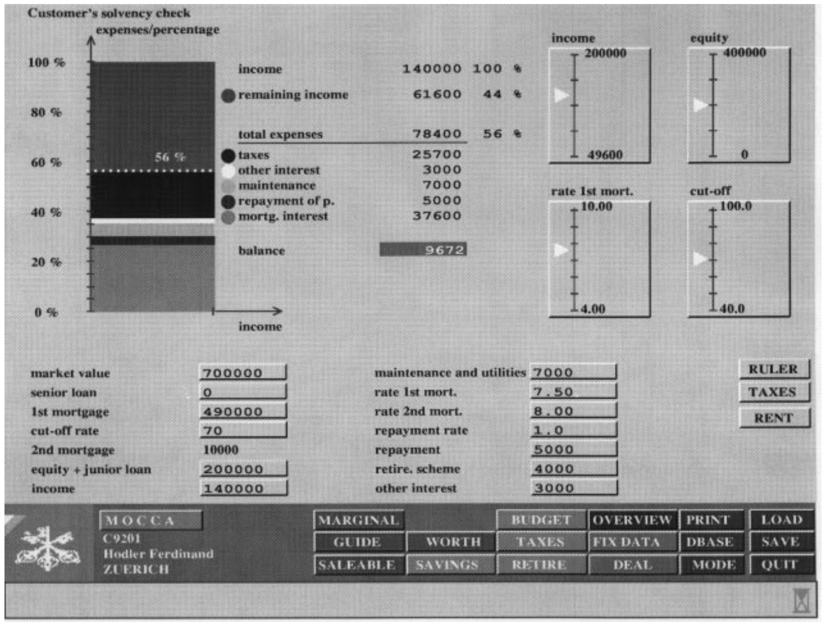


Figure 4. Solvency Check.

a leading indicator of property risk (along with the worthiness investigation). The bank has to avoid the risk of becoming proprietor of the real estate when there is no longer any acceptable demand for the property. The importance of this risk has recently become apparent in different parts of the business. The model implemented in MOCCA is completely new—not even a rule of thumb exists, in contrast to the solvency check. It was elaborated by the expert who was inspired by the solvency check solution and the worthiness investigation. He combined them into a new model that calculates the minimum required income at different financial constellations and the quality of the property according to the worthiness model and compares them with the possible demand. In this way, any object can be classified by its market attractiveness.

With the elaboration of taxes and personal budget, important competitive advantages are achieved, requiring little effort by the staff. In addition, these elements enlarge the information base for a more competent evaluation. Although they are crucial for any solvency investigation, they had never been implemented before.

The development of all economic innovations is closely tied to the underlying technological environment, which consists of a combination and an extension of different knowledge-based techniques. The following section discusses the technological innovations that allow real prototyping with full reusability, the composition of heterogeneous instruments, and the overall automatic truth maintenance mechanism.

Technological Innovations in MOCCA

The technology described in this section enabled iterative model-oriented development (model-oriented prototyping), which was an effective and, we believe, general methodology for developing knowledge-intensive, cooperative applications (instruments). *Agent-based processing* is a concept for modeling and mastering knowledge-driven processing that claims a powerful knowledge representation by combining objects, lists, and tables. The highly iterative development of operative prototypes demands fast technical realization that is achieved by generic building blocks for the problem-solving components and the user interface. In addition, the outlined two-layer windowing system guarantees platform independence.

Agent-Based Processing

Agent-based processing is part of the executable methodology for knowledge-based application development (EMA) (Spirgi, Probst, and Wenger 1990, 1991). It combines the object-oriented, data-driven, and effect-oriented paradigms. The global processing built into MOCCA is based on this approach. Processing is executed by agents that can be seen as a cluster of conceptual rules.

An *agent* consists of an activation, a retrieval, and a processing part (figure 5). The *processing part* produces effects that change the state of the application. An *effect* is a manipulation of information in an area, for example, the internal global data area (mainly built of schemata), the database area, or the screen area. The *retrieval part* accesses all the information that is processed by methods to set up the necessary effects. All areas can be accessed. The *activation part* is responsible for the activation of the agent. A matching is done that compares internal global data with the pattern. The activation pattern, the retrievals, and the effects declare the agent exhaustively as well as the processing on a conceptual level.

In figure 5, the pattern of the activation part matches the information pieces i1, i2, and i3. The agent accesses internal global data (i4, i5, and i6) and the database area (i7). The effects, produced by the processing part, manipulate the internal global data (e1, e2, and e3), the database area (e4), and the screen area (e5).

The impact of the agent-based approach was tremendous for the prototyping, the cooperative work of the different subsystems, and the

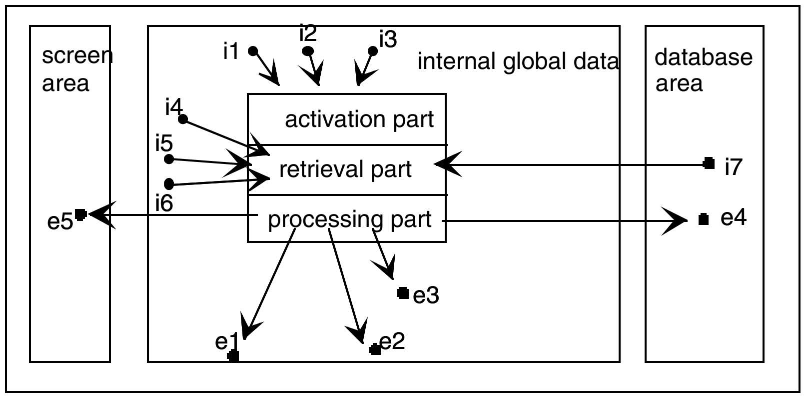


Figure 5. Structure of an Agent.

flexibility and evolution of the application.

Objects, Lists, and Tables: One Conceptual Unity

All these representations have their advantages and weaknesses. In MOCCA, they are combined in a complementary fashion to benefit from the advantages of each.

Object-oriented representation is powerful for deep object structures (many abstraction levels and concepts), but with respect to mass data (thousands of objects), it is inefficient and consumes much storage space. The strengths and weaknesses of a table are just the opposite. Therefore, both representations are used and combined. Figure 6 points out the principle. Each instance of the object DEAL (figure 6a) equals one row of the table (figure 6b).

Extending the object-oriented representation by lists improves the capability for building associations. In MOCCA, in addition to a deep structure, a shallow relational structure exists (associations 1:n, n:1, and n:m). A simple example is shown in figure 7. In the worthiness investigation, different descriptors have an impact on different aspects of worthiness. There is an n:m relation between descriptor and aspect. The relation object is called p-eval and has the attributes value and weight. Figure 7a shows the conceptual representation, and figure 7b shows the technical representation. The object p-eval has disappeared, and an attribute p-eval has been installed that relates aspect and descriptor. In EMA, this technical representation can be built automatically from the conceptual representation with the help of some guidelines. At the end, we get a list of value, weight, and descriptor. This example is only a simple one. More complex shallow structures lead to

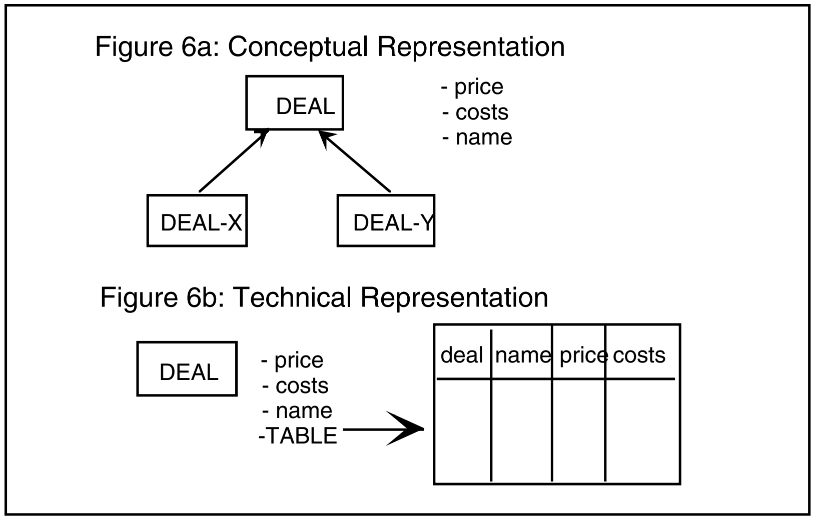


Figure 6. Combination of Objects and Tables.

more complex lists (list of lists). The conceptual unity of different kinds of data representation lets us model on a conceptual level and lets us achieve high efficiency.

Generic Application Building Blocks

The generic systems IO-GEN, IO-PRINT, and G-TECS provide the generic application building blocks for MOCCA. *Generic* means that all blocks can be instantiated and combined easily without a large effort.

IO-GEN provides the user interface building blocks. The main principle is, Point, look, and act. The user can explore the application by moving the mouse to a screen object and looking at the help line, where an explanation of possible specific actions appears. Each screen becomes self-explanatory without any help function. The user can drive the application by clicking a mouse button or giving input through the keyboard. The characteristics of every screen object can be specified by defining methods for the displaying of the object, the changing of the mouse position, and the reaction toward input given through the keyboard or mouse. With the combination of this principle with the data-driven paradigm and the IO-GEN predefined objects, it was possible to create powerful simulations in a short amount of time. With a set of graphic functions, the developer can extend the standard set of screen objects if a new element is needed. If such a new item is useful for similar topics in other applications, it is generalized and

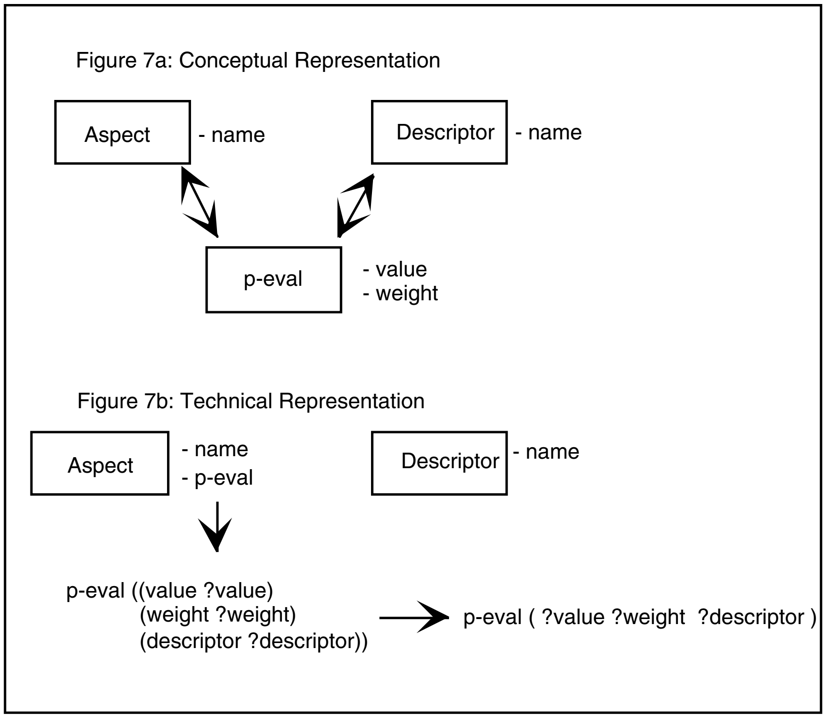


Figure 7. Relations and Lists.

added to the standard set. This procedure ensures that the user interface becomes more powerful with every application.

IO-PRINT uses IO-GEN structures for making printouts. For the developer, there is no difference between designing screens and printouts.

Information-Based Problem-Solving Techniques (G-TECs)

A problem-solving technique should be adaptable, transparent, and efficient. It should be transparent to the developer and the user. Within our applications, G-TEC (generic technique) (Spirgi, Probst, and Wenger 1990) consists of a data structure, which includes an object (name of G-TEC), the problem-solving algorithm (method); and the user interface building blocks that provide the graphics for representing information, results, and intermediate results of the problem-solving technique. The developer instantiates a certain G-TEC and gets an object that is integrated in his/her application.

CONDEV stands for condensation of evaluations. It is G-TEC that is integrated in MOCCA. It operates on a net, an example of which is shown in figure 8.

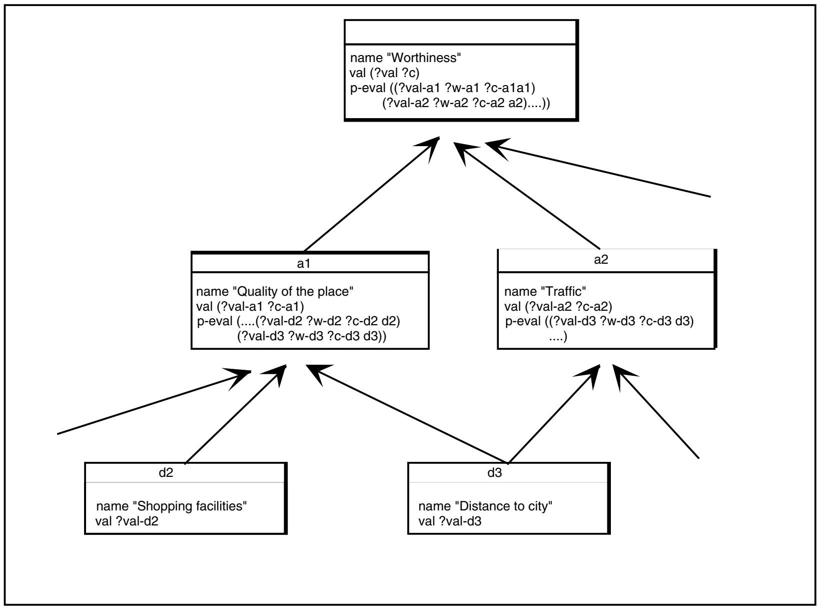


Figure 8. G-TEC CONDEV.

The input information is primarily partial evaluations, which are propagated to the nodes of the higher levels. They consist of a value and a certainty (for example, ?val-a1 and ?c-a1 in figure 8). For every node, CONDEV builds the evaluations (for example, a1@val) based on all propagated partial evaluations. It is robust against incompleteness of information and contradictory information. The worthiness check (a part of it is shown in figure 8) is a problem in MOCCA that was solved with CONDEV. The values and the weights of all descriptors (the nodes d2 and d3) are propagated to all aspects (the nodes a1 and a2); these aspects are finally propagated to the worthiness value, which produces an ideal ratio between worth of the entire property and worth of the land. To propagate the certainty in CONDEV, a simplified fuzzy measure approach was implemented. The partial evaluations of each aspect are combined to form one resulting valuation. Each additional partial evaluation modifies the resulting certainty according to its significance for the resulting evaluation.

Free Composition of Subsystems to Enable Real Prototyping

In the data-driven development environment, an application is driven by the agents that react to internal global data manipulations (blackboard

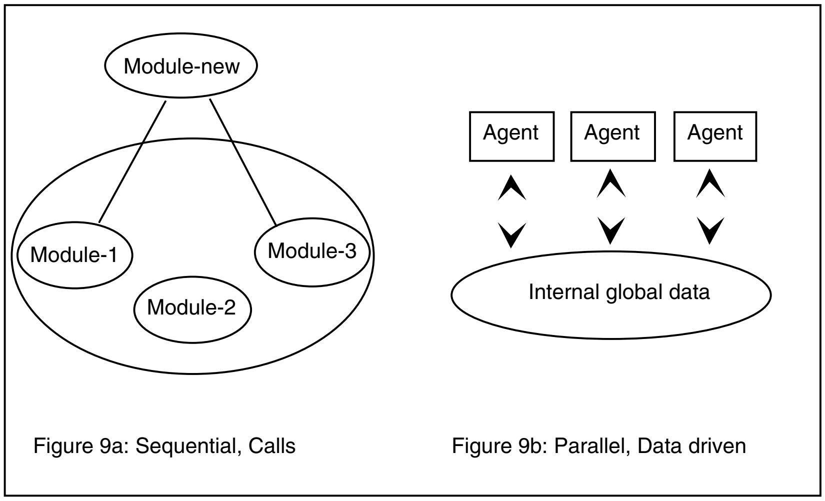


Figure 9. Data-Driven Paradigm.

principle, figure 9b). There is no need for interface programming between the subsystems (figure 9a). With this open-interface architecture, it is easy to build an application by combining independent subsystems. In MOCCA, we started to build a set of productive applications for mortgages of private properties only. In this way, we could explore the task domain in one area and see what features are important. When we started to implement the rest of the application, we could reuse these components entirely.

The basic principle is the development cycle (figure 10), which can be described as model-oriented prototyping based on a conceptual model. In each step, the missing, contradictory, and incomplete knowledge is rebuilt by verifying and validating the prototype to obtain a model that contains all the information that is important for the application. These activities are supported by EMA, which helps to formalize unstructured knowledge and support the developer in bringing the acquired knowledge into a prototype, verifying and correcting it. Building up knowledge is the main task in developing knowledge-based systems. The valuation of the customer's solvency in MOCCA is a typical result of prototyping. For many weeks, the current decision process was investigated, supported by EMA, and then replaced by a new set of decision variables. Within the prototyping process, a lot of new information, which had previously been neglected, became important (taxes, private budgets). The new solution can easily be transferred to similar problems, such as the salability investigation, that occurred during the prototyping process.

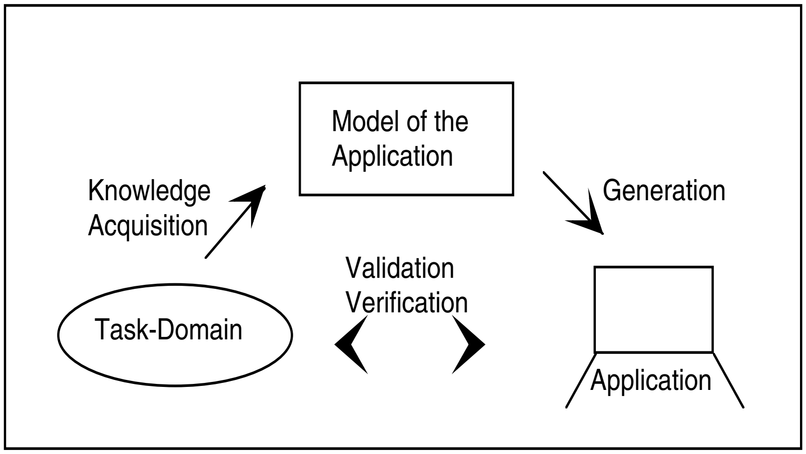


Figure 10. Application Development Cycle.

Another area of prototyping is the promoting of a new idea. For a new mortgage savings model, it was possible to develop a simulation prototype within two days that helped management decide on what terms and conditions the new product was usable and profitable.

Two-Layer Windowing System

In our windowing system, there is a distinction between operating system windowing (for example, Microsoft WINDOWS, OSF/MOTIF, OPEN LOOK, PRESENTATION MANAGER) and the application's object-oriented windowing in IO-GEN. With IO-GEN, the applications can be handled in the same way regardless of the underlying platform. Only a minimal set of graphic C routines have to be adapted to carry IO-GEN into any operating system environment. Any further development progress is instantly usable for each product on any platform because of full portability. Therefore, our products are independent of any new windowing standards that will come on the market. It allows us to concentrate entirely on the development of the applications themselves.

With the embedding of MOCCA in an operating system windowing environment such as WINDOWS 3.1, we could reuse other commercial knowledge-based systems for certain features; for example, the solvency check for commercial clients can be done by our credit-analysis system CUBUS (Wolf, Wenger, and Kirchmayr 1991).

Application Development and Deployment

MOCCA was developed in a short time by a small interdisciplinary group.

The local manager of the mortgage credit section acts as the expert responsible for all commercial aspects of the application and invests a maximum of 10 percent of his time in the development, promotion, and verification of MOCCA. Steve Hottiger acts as full-time technological project manager. For some problems, other specialists in the bank are called on for support. Important for the project's success was the technical support from other knowledge-based project managers in the group. The development environment is based on ART-IM 2.1 for DOS on an IBM PS/2.

The Project's Timeline

1990

April/May: Initial workshop conducted, where the expert and the developer explore the task domain. They develop a network of all relevant components used in the decision process. They decide to concentrate on the most important components (figure 1).

June: We start implementing the first prototypes (solvency control and worthiness investigation). The expert uses the system right from the beginning in his daily work.

October: First demonstration is given of the prototypes. A similar project in the retail banking section starts up where certain instruments will be reused.

1991

January: A small group of motivated managers from other mortgage credit sections is formed to support and introduce the new product. They are invited to contribute any input for further development.

July: The application fully supports the business section for private customers and self-contained buildings. The implementation for the other business sections (figure 2) starts.

September: The two main SBC branches introduce MOCCA.

The total personnel investment in the application is two person-years. More than 60 percent of the time was spent prototyping and developing the models. The remainder was used for customizing, demonstrating, and introducing the system.

Use and Payoff

The quantification of the payoff has to respect the different uses of the application, both monetary and nonmonetary. Qualitative advantages are mentioned to give a comprehensive idea of the use of MOCCA.

Cost savings by reducing external consultancy: MOCCA helps to separate the easy deals from the difficult ones. In this way, internal experts can be used better for the examination of crucial deals, and less external support is required.

Increasing decentralization: The comprehensive objective analysis of all critical items causes a shift of responsibility to the front staff and, thus, minimizes the effort spent on control.

Improved sales assistance: Self-explanatory and attractive instruments for all relevant bank products are basic for promoting certain products. They also contribute to a better bank image that demonstrates competence and fairness.

Reduction in risk: MOCCA provides a set of instruments to counter the increasing risks of loss. Quality is preferred to turnover.

Increased management control: MOCCA is a medium to quickly carry through any guidelines and standards in the mortgage business. Multidecision possibilities enable management to control the granting of mortgage credit.

Improved staff know-how: The easily understood model accelerates the understanding of the staff members of the crucial points of their business.

With all these factors, we estimate the profit to SBC from using MOCCA to be $2 million each year.

MOCCA is currently used by more than 20 people working in different mortgage credit sections of the bank. It is fully accepted and used daily. Supporting the front office staff, MOCCA is extremely market driven. Therefore, the maintenance of the system has to be done by the developer to ensure that the application is adapted to any important new requirements. With the open design based on the data-driven development environment and the embedding of the application in EMA, any changes can easily be performed. Furthermore, the use of EMA ensures that any developer familiar with the environment can continue the work.

Outlook

MOCCA is the mission-critical system for mortgage lending at SBC. Other applications based on the same development environment are successful in the finance and the commerce departments (for example, CUBUS). The development of highly interactive cooperative tools based on new decision patterns has a tremendous impact on the work process. Now, a number of banking specialists are motivated to work on the development of knowledge-based systems.

References

Fierz, K. 1987. *Wert und Zins bei Immobilien.* Schriftenreihe der Schweizerischen Treuhand- und Revisionskammer Band 56, 2. Auflage.

Nägeli, W. 1980. *Handbuch des Liegenschaftenschätzers*, 2. Auflage.

Spirgi, S.; Probst, A. R.; and Wenger, D. 1991. Generic Techniques in EMA: A Model-Based Approach for Knowledge Acquisition. In Proceedings of the Sixth Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW91).

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(JKAW '90), eds. H. Motoda, R. Mizoguchi, J. Boose, and B. Gaines, 382–397. Tokyo: Ohmsha, Ltd.

Wolf, M. F.; Wenger, D. ; and Kirchmayr, K. 1991. CUBUS—An Assistant for Fundamental Corporate Analysis. In *Innovative Applications of Artificial Intelligence* 3, eds. R. Smith and C. Scott, 271—291. Menlo Park, Calif.: AAAI Press.