ECONSTOR
Make Your Publications Visible.

A Service of
ZBW
Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Unland, Rainer; Kirn, S.; Wanka, U.; O'Hare, G. M. P.; Abbas, S.

Working Paper

AEGIS: Agent oriented organisations

Arbeitsberichte des Instituts für Wirtschaftsinformatik, No. 36

Provided in Cooperation with:
University of Münster, Department of Information Systems

Suggested Citation: Unland, Rainer; Kirn, S.; Wanka, U.; O'Hare, G. M. P.; Abbas, S. (1995) : AEGIS: Agent oriented organisations, Arbeitsberichte des Instituts für Wirtschaftsinformatik, No. 36, Westfälische Wilhelms-Universität Münster, Institut für Wirtschaftsinformatik, Münster

This Version is available at:
https://hdl.handle.net/10419/59342

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

WWW.ECONSTOR.EU

Mitglied der Leibniz-Gemeinschaft

Working Papers of the Institute of Business Informatics
Editors: Prof. Dr. J. Becker, Prof. Dr. H. L. Grob, Prof. Dr. K. Kurbel,
Prof. Dr. U. Müller-Funk, Prof. Dr. R. Unland, Prof. Dr. G. Vossen

Working Paper No. 36

AEGIS: AGENT ORIENTED ORGANISATIONS

R. Unland, S. Kirn, U. Wanka,
G.M.P. O'Hare* and S. Abbas*

* University of Manchester
Institute of Science and Technology (UMIST)

University of Münster, Institute of Business Informatics,
Grevener Str. 91, D-48159 Münster, Germany, Tel. (0251) 83-9750, Fax (0251) 83-9754
Februar 1995

1 Introduction 4

2 Problem Description: Credit Advice 6
2.1 The Banking Environment 6
2.2 Credit Advice Consultation Process 7

3 Emergent Organizational Strategies: A Brief Review 8
3.1 Business Process Orientation 8
3.2 Fractalization 9
3.3 Fractalization versus Business Process Orientation: An Emergent Conflict 9

4 Distributed Artificial Intelligence 11
4.1 Agency within Distributed Artificial Intelligence 11
4.2 Multi-Agent Systems 12
4.3 A Distributed Artificial Intelligence Perspective on Planning 13
4.4 Agent Modeling 14

5 Functional Design of AEGIS 16
5.1 Organizational Model 17
5.1.1 Agent Model 19
5.1.2 Initial Design 20
5.2 Architecture of the AEGIS Multi-Agent Organization 21

6 Engineering and Customization of Computerized Business Processes 25
6.1 Process Modeling Categories for Business Process Engineering 25
6.1.1 Construction and Modification of Processes 26
6.1.2 Abstractions on Processes 26
6.1.3 Modeling of Interactions among Processes 27
6.1.4 Modeling of Interactions of Processes with their Environment 28

7 Realization of the AEGIS System 28
7.1 Plan Model for AEGIS/CAS 28
7.1.1 Modification of Processes: The Process Dimension 29
7.1.2 Abstractions on Processes and Actions: The Granularity Dimension 29
7.1.3 Modeling of Interactions among Processes 30
7.1.4 Modeling of Interactions of Processes with their Environment 30
7.2 Example: Plan Refinement in AEGIS/CAS 31
7.3 Implementation Environment 33

7.4 Implementing Agent Interfaces with CORBA 33

8 Conclusion 34
Literatur 36

Abstract

Today's enterprises are faced with highly dynamic, sometimes even hostile environments. This paper dicovers that the two most prominent organizational strategies addressing these challenges, namely that of widespread decentralization and of business process orientation, are inherently conflicting.

We argue that cooperative knowledge processing technology can contribute to dissolve this conflict. For this purpose, we introduce the reader into the field of Distributed Artificial Intelligence. On that basis we then develop the concept of Agent oriEnted orGanISations (AEGIS), in order to support the flexible modeling and management of business processes in decentralized organizational settings.

Applying to methods of Distributed Planning a set of process modeling and process interaction operators is defined. These operators also permit to automatically create and customize computerized configurations of business processes. The concepts are presented in the context of an application in private banking, namely that of a Credit Advisory System.

1 Introduction

Today, organizations are faced with permanently changing markets, global competition, rapidly decreasing cycles of technological innovations, worldwide (and just in time) availability of information, and considerable changes in their cultural, social, and political environments. The ability of enterprises to achieve competitive advantages and to continuously survive in dynamic, sometimes even hostile environments, largely depends on their organizational flexibility. This can be achieved through Business Process Orientation which is facilitated by harnessing appropriate enabling technology.

From research on organizational networks and virtual organizations it has been revealed that one of the most important problems in such settings is how to efficiently coordinate sequences of activities which are under decentralized local control. Moreover, business process (re)engineering not only (re)designs organizational processes but also involves a significant shift of tasks from human workers to so-called *computational agents*. As a consequence, whenever business process (re)engineering is effected, we are increasingly concerned with adapting and combining software-based sequences of organizational procedures instead of modeling these procedures from scratch.

This evolution has been accompanied by some promising advances in information technology (IT). Client Server Computing has proven to be very successful in breaking down well-established centralized information systems architectures. Another breakthrough has been the definition of the CORBA standard (Common Object Request Broker Architecture) by the Object Management Group in 1991 (Object Management Group, 1992). The recently emerging discipline of Cooperative Information Systems advocates a cooperative approach to coordinate the activities of decentralized, autonomous and heterogeneous information systems. These advances, together with the availability of high-speed communication networks (like FDDI), information super highways and associated information browsers like Mosaic and World Wide Web (WWW), facilitate research efforts into how organizational networking and decentralized (computerized) organizational procedures can effectively be supported by innovative software architectures.

It is our strong belief that, within the context described above, the discipline of Distributed Artificial Intelligence (DAI) can contribute significantly to this challenge. This paper adopts an agent oriented approach to organization modeling namely that of **Agent oriEnted orGanISations (AEGIS)**. Agent oriented organizations offer the ability to:

- ☞ dynamically adapt themselves to organizational aims and objectives, structures, processes, and constraints,

- ☞ provide efficient support for the cognitive skills of an organization, such as organizational perception, learning, problem solving, communication, and

- ☞ actively contribute to business process orientation and the processing of computer-based organizational procedures, through a toolbox of decentralized coordination mechanisms through which a broad variety of organizational structures and coordination principles can be supported.

Within such a framework computational agents take charge of explicit organizational roles. Thus, evolving into self-contained organizational agents which no longer suffer from the inertia of today's business information systems. It has been argued also, that one of the most important advantages of agent oriented organisations is their capability to support customer orientation through flexible process management. Within this paper we concentrate on process orientation, namely that of automated configuration and customization of computerized workflows and business processes.

Due to the conceptual equivalence of multi-agent plans and (formally represented) organizational processes our technical approach draws heavily upon multi-agent planning. We suggest a plan coordination approach that provides for customizing intra- and inter-organizational configurations of software-based, adaptable organizational procedures. This can be achieved by providing a diversity of mechanisms to plan and control the activities of decentralized, autonomous organizational units. For this purpose we introduce a set of generic operations by which multi-agent plan creation via plan modeling can be supported. The power of the approach is demonstrated through a plan refinement example taken from a credit application in private customer-banking.

The remainder of this paper is organized as follows: The next section introduces the problem using a credit advice process in a banking environment as an illustrating example. The subsequent section briefly reviews emergent organizational strategies and is followed by a concise review of the concepts coming along with Distributed Artificial Intelligence. The next section is going to show how this will combine within the functional design of AEGIS. The subsequent section on engineering and customization of computerized business processes is introducing four categories of process modeling operations necessary for a holistic approach to

business process orientation. How this is realized within AEGIS is described by the following section and the concluding section is outlining the work that still remains to be done.

2 Problem Description: Credit Advice

Within the context of this paper we will focus on a banking application as this is typical of many modern complex organizations. Specifically we will focus our attention upon a particular banking function the provision of credit advice and the identification of appropriate credit products.

2.1 The Banking Environment

Within a banking environment, products are services and as such can be represented by the (standardized) processes necessary to produce and sell them. Product bundles refer to general financial services offered by a given bank or indeed a financial consortium of which the bank is part. Examples may include *inter alia* combinations of life insurance, equity management, stocks and shares, trusts and so forth. Customer orientation, within the context of a banking application domain, involves the opportunistic assembly of ‘product bundles’ that fulfill the needs of the customer (Buhl et. al., 1993).

A customer oriented approach must be holistic in nature, taking due cognizance of the multi-faceted nature of the modern bank. An advisory system which utilizes Information Technology can significantly assist bank consultants in identifying appropriate bundles. However, true quality is ensured only if the aims, objectives and expertise of all the constituent departments are fully integrated into such an advisory system. Through this quality the bank can maintain their competitive advantage. The effective support of customer consultants of a bank can be achieved by providing assistance in the construction of such bundles using bank-wide searches for, and assessment of, the appropriate processes and a set of operations to combine and adapt the most promising ones.

The quality of the services provided by a given bank is intimately dependent upon the quality of the customer consultation process. This process is important in two respects. Firstly, the accuracy and completeness of the information accrued will directly influence the detailed analysis that will recommend particular product bundles. Secondly, financial services need to be thoroughly explained to the customer in order that they appreciate the relative advantages of competitive products and associated contractual obligations.

Current systems within the financial services sector are primarily aimed at the creation of models which simulate money flow and financial models. Little support exists as yet, for the consultation process and what support does exist does not yield true intelligent decision support but merely supports passive retrieval of data that will be used by the consultant in such decision making.

2.2 Credit Advice Consultation Process

A typical problem scenario would involve three steps:

Step 1: The customer makes contact with the bank:

Two concurrent processes evolve. Firstly, the bank tries to learn about the customers needs, preferences, and requirements. Then, the bank translates this information into an assessment of the expected "value" of that customer (that is, the potential income the bank can generate with such a partnership, will the customer be an important contact or potential business opportunity and so forth.). Secondly, the customer tries to ascertain whether or not the bank could provide him with the appropriate solutions for his requirements to its internal utility ranking.

Step 2: Definition of "Contract Parameters"

The bank then fixes an initial set of credit contract rules. Such rules may look like:

```
IF  risk < x% AND profit > y% of total_amount_borrowed
    THEN credit_contract
```

These rules depend largely upon the internal cost structures of the bank, the bank’s aims and objectives, the aims and objectives of the experts (agents), and the "value" of the customer for the bank. For example: A credit agent may have to contract 1.000.000 DM per month. After four weeks of month 1 he has contracted only a sum of 500.000 DM. Thus, she would be very interested to attract the customer and, thus, may even accept to share a portion of his personal income with the customer. In month 2, the agent may have already contracted 1.500.00 DM at week 4. Now, the agent will not be interested in having additional credits in month two. Thus, she may chose either the strategy to reject the application, or to defer the decision to the subsequent month, or to try to earn as much money as possible from that customer.

Step 3: Bank Team Involvement

In general, private banking involves (much) more than merely one bank agent. Thus, all agents involved must negotiate in the agreement of a new contract. Each agent has local aims, which are couched in terms of its local earning system. The aims of these agents may be inconsistent, or even conflicting. For instance, some agents may be very interested in conducting business (depending on their local situation), others may prefer to reject the application. Thus, the agents have to negotiate what their collective aims ought to be. This may involve inter-agent money transfers in payment for their services. Several such negotiation strategies need to be developed and evaluated here.

This step supposes the availability of (maybe virtual) global data structures through which the mental state (including the product development process) of the team can be represented.

3 Emergent Organizational Strategies: A Brief Review

In the past, a great deal of work has been devoted to develop new strategies by which organizations can meet the challenges of dynamic, and sometimes even unpredictable environments, rapidly decreasing cycles of innovation, and worldwide competition. Two different strategies can be identified: re-engineering the *processes* of an organization, and radical decentralization, or fractalization, of the *structure* of organizations. Finally, we discover that current approaches to business process orientation are, at least partially, incompatible with the strategy of radical decentralization.

3.1 Business Process Orientation

According to Davenport (1993) a process may be defined as a structured, measured set of activities designed to produce a specified output for a particular customer or market. Process orientation, thus, implies a strong emphasis on *how* work is accomplished, in contrast to a more product-oriented focus on *what* is achieved. Thus, business process orientation represents a revolutionary change in perspective: it turns the organization on its head, or at least on its side (Davenport, 1993, p. 5).

The structure of business processes can clearly be distinguished from the more hierarchical forms of structure within an organization. While the latter is typically a snapshot which shows how responsibilities, resources, communication channels, and information flow (that is, repor-

ting, instructions) are distributed across an organization, the former provides a dynamic view of how the organization delivers value. Furthermore, while structures cannot be assessed or improved directly, processes involve cost, time, output quality and customer satisfaction, they relate to well-known triggering events and they result in well-defined final states. Thus, whenever one reduces cost or increases customer satisfaction processes are improved, and not the organizational hierarchy.

Since the pioneering work of Porter (1985), business process orientation has received an overwhelming attention by both academia and practioners (Scheer, 1994). Its most important contribution is that it provides for a systematic, intelligible approach to the modeling and (re-) engineering of organizations. Thus, for the rest of the paper we assume that the design of future organizations will focus on processes, that is, that it will primarily be concerned with the modeling, management, and control of processes rather than applying to task decomposition, and to the modeling of static organizational structures.

3.2 Fractalization

It has been argued that the enterprise of the future will be radically decentralized, in order to meet the challenges of the increasing complexity of their environment, and the dynamics of world-wide competition. Decentralization involves the allocation of autonomy, resources, and responsibilities to deeper levels of the organizational hierarchy (for instance, see work of Tapscott & Caston (1992) or Warnecke (1991)). This requires enterprises to replace the traditional approach of hierarchical planning by more decentralized concepts of coordination. In turn, autonomous organizational subunits need to exhibit a much greater degree of intelligence and self-referencing skills than they do today. This has given rise to the notion of organizational fractals (Warnecke, 1991). These are equipped with self-organization skills thus enabling them not only to recursively form complex, highly-organized entities but also to modify these entities, for instance with respect to dynamic environments or changing customer demands. Organizational fractals are provided with operational definitions of their local aims, and are capable of cooperatively creating global hierarchies of aims and objectives. They exhibit intelligent local and global coordination skills, and a benevolent style of cooperation. Accordingly, fractalization permits large organizations to exhibit greater flexibility and adaptivity, and it also provides a medium through which they can effectively refresh their learning capabilities (Warnecke, 1991).

3.3 Fractalization versus Business Process Orientation: An Emergent Conflict

Organizational fractals involve a maximum degree of local autonomy, self-control, and self-organization skills. Aiming to maximize their local utility (for instance, in terms of profit), organizational fractals decide on their own whether they are willing to cooperate, or to collaborate with other organizational units. There is no direct means by which fractals can be compelled to behave in a certain manner. The single acceptable way to control the behavior of an organizational fractal, or of a group of cooperating fractals, is through designing a globally consistent system of aims and objectives (Warnecke, 1991).

However, due to bounded rationality, organizations are, in most cases, not able to establish consistent goal hierarchies. Instead, the different goals that exist within an organization are more or less inconsistent, the knowledge about goals and relationships between them remains necessarily incomplete, uncertain, fuzzy, and sometimes even false. Additional goal conflicts may arise between the goals of an organization and the preferences of its customers, between different organizations that wish to cooperate, and between the customers of distinct organizations that wish to pursue their aims in close cooperation.

In contrast, current approaches to business process orientation presuppose that organizations have the time, knowledge and skills to precisely describe, analyze, and design enterprise-wide business processes. They further suppose that, through an iterative (not necessarily algorithmic) procedure of refinement, these descriptions can be augmented, or instantiated to more detailed descriptions of (partial) processes and fully expanded specifications of the respective workflows. This requires a centralized approach to business process engineering, or, at least, a global perspective (Hammer & Champy, 1991) which, by definition, cannot exist within fractalized enterprises. Instead, organizational fractals must cooperate whenever they aim to create an enterprise-wide business process, akinned to bottom-up approach. A similar conflict exists between the local autonomy of organizational fractals, and the need to tie them to existing business processes. It shall be noted that these conflicts are directly related to the degree of autonomy that is exhibited by the fractals involved.

As a consequence, it is very difficult, or even impossible, in general to resolve these conflicts by standardized rules and decision criteria. Instead, whenever one aims to introduce business process orientation into a system of organizational fractals one needs to apply to decentralized, cooperation-based concepts of process modeling and control. Moreover, any coordination concept must sustain the individual autonomy of each single fractal, and it must provide appropriate knowledge processing techniques in order to cope with the epistemological issues of incompleteness and uncertainty.

These challenges can be addressed by recent advances in organizational computing, namely with respect to the technology of cooperative knowledge processing (Warnecke, 1991), (Kirn, 1995), (Kirn & O´Hare, 1995). This work aims to integrate decentralized but autonomously operating data and knowledge sources, and it involves standard techniques from Artificial Intelligence. It has been demonstrated that these techniques are suitable to bridging the boundaries within and across organizations, and to flexibly exchange tasks, know how, and geographically distributed resources between distinct organizational units.

4 Distributed Artificial Intelligence

Distributed Artificial Intelligence represents a departure from the centralized approach advocated by mainstream Artificial Intelligence. It seeks to develop intelligent systems by distributing intelligence (or expertise) across a community of semi-autonomous automated intelligent agents (O'Hare & Jennings, 1995), (Chaib-draa et.al., 1992).

Various motivating factors exist for developing Distributed Artificial Intelligence systems. Firstly, the growth in inexpensive and flexible telecommunications networks and multiprocessor hardware and software has provided the means to develop distributed systems (Gasser, 1992b). Secondly, the expertise necessary to achieve adequate solutions to complex problems demand the provision of multiple cooperating systems. Furthermore, Distributed Artificial Intelligence can potentially solve problems that are too complex for one centralized system, which may be due to resource limitations of a given level of technology. Distributed Artificial Intelligence can also provide a means of connecting multiple knowledge bases with differing, but possibly overlapping expertise, thereby enabling the solution of problems whose domains are outside that of one knowledge base (Hern, 1988), (Huhns, 1987). Further advantages of the Distributed Artificial Intelligence approach to system development is that it facilitates resource sharing, flexibility, efficiency, reliability, and speed.

The following sections expand upon the notion of 'agent' and 'Multi-Agent Systems', their relevance to human organizations, emerging programming paradigms and the importance of planning in developing Multi-Agent Systems.

4.1 Agency within Distributed Artificial Intelligence

Agents are often physically and logically distinct and are typically capable of reasoning, planning, communicating and cooperating (Hern, 1988). Agents may be robotic and be defined

in terms of sensory input, motor control and time pressures, they may perform cognitive functions, react to stimuli, contain symbolic plans, or possess natural language capabilities. Shoham (1993) points out that:

> *"An agent is an entity whose state is viewed as consisting of mental components such as beliefs, capabilities, choices, and commitments."*

An agent is thus regarded as an entity, that functions continuously and largely autonomously, with little guidance or intervention. An agent functions in an environment which is typically highly dynamic and unpredictable, within which other agents coexist and perform.

4.2 Multi-Agent Systems

According to Werner (1992), a multi-agent system may either be a software system, a number of robotic agents, it may be a group, organization, or society of humans. In essence, multi-agent systems may be described as reactive systems that interact with their environment to perform a particular task. Such systems are concerned with coordinating behavior among a collection of possibly preexisting intelligent agents. A multi-agent system attempts to coordinate the agents' expertise, knowledge and plans in order to achieve a common purpose. Agents within a multi-agent system interact and their mode of interaction may be guided by cooperation strategies to improve their common performance.

Research in multi-agent systems endeavors to deal with the problem of integrating and coordinating the activities and expertise of agent communities, in an opportunistic, mutually beneficial manner, in the pursuit of cooperative solutions, to shared problems. These groups and the individuals within them evolve, as do the relationships between their team members. No one agent may have sufficient information to solve the entire problem, be the most appropriate to perform a particular task, or be able to supply relevant information. Agents must necessarily collaborate, demanding mutual assistance referred to as 'collective problem solving'. Such collective behavior evolves and generally reflects the development of inter-agent relationships.

Attempts to understand the fundamental social phenomena that underpin agent behavior have lead to the development of two crude classes of multi-agent system, those of deliberative or reflective systems and those regarded as 'reactive' systems (Wooldrige & O'Hare, 1991). Deliberative multi-agent systems ascribe roles to their agents, with the appropriate expertise facilitating cooperation between agents. The concept of 'role' is cited by Werner (1992), as an important mechanism for building complex social structures, these structures he terms

‘organizations’. According to Werner (1992), in order for agents to be cooperative they must have information about their environment together with some planning capability, or at the lowest level, procedures or programs. Agents are able to make inferences in choosing the most appropriate action, (this can be viewed as ‘rational’ behavior) and are able to predict the behavior of other agents. These agents make ‘evaluations’ or ‘deliberations’, over a choice of action or reaction. When faced with a problem, agents take the appropriate action to arrive at a solution. This manifests itself as emergent social behavior that is normally associated with cooperating agents in organizations, such as ant colonies and human communities such as human organizations (Lesser & Corkhill, 1981), (Fox, 1979), (Fox, 1988).

4.3 A Distributed Artificial Intelligence Perspective on Planning

Within multi-agent systems, coordination of decentralized activities is required. According to the equivalence of plan and process we apply here to a planning approach to coordination. Within Artificial Intelligence planning there is usually a set of goals given together with an initial state and a set of allowable actions in a planning environment. The planning task is to find a sequence of actions that fulfills the respective constraints and which allows the system to achieve all of the desired goals. Exemplary Artificial Intelligence planning systems are STRIPS (Fikes & Nilsson, 1971, p. 189-208) and NOAH (Sacerdoti, 1977). However, Artificial Intelligence planning is not applicable to typical multi-agent settings because of its restrictive assumptions (v. Martial, 1992):

- ☞ the world is assumed to be static and only affected by the actions of one single agent (the planner),
- ☞ there is only one agent who designs the sequence of actions (the plan) necessary to solve that task, and who itself is responsible for carrying out the actions represented in that plan,
- ☞ plans are concerned with prevention of local conflicts, not with coordination of cooperative or competitive local behavior of multiple agents,
- ☞ single agent planners are not capable of solving complex tasks collaboratively,
- ☞ single agents are not capable of reasoning about actions that are not under their local control, and

- ☞ there is no concept of concurrent activities.

These limitations have stimulated research on multi-agent planning, i.e. on developing towards planning approaches where one or multiple agents cooperatively design multi-agent plans to coordinate the activities of a set of autonomous computational entities (Bond & Gasser, 1988). Within the field of Distributed Artificial Intelligence, planning involves the recognition of any potential interference between the activities of the agents and as such the planning activity attempts to align their activities so as to avoid destructive goal conflicts. Interference could involve not merely incompatible ordering of actions but also incompatible states and conflicting demands on resources.

v. Martial presents a taxonomy of planning systems. Two fundamental approaches exist to multi-agent planning: goal-driven planning (tasks [goals] are decomposed into subtasks [sub-goals], plans are constructed from scratch) and plan coordination, where pre-existing plans are to be coordinated and adjusted through cooperative plan reconciliation before they are executed. The input for a plan coordination problem consists of several (partial) plans. To coordinate individual plans the agents must be able to modify them with respect to a set of (globally agreed) goals (e.g. long-term profit maximization). The outcome of the multi-agent planning task (i.e. the solution to the actual plan coordination problem) consists of a set of coordinated individual plans which, subsequently, are to be executed under decentralized control.

Thus, besides having communication skills, the (group of) coordinating agent(s) must be *capable* and *legitimated* to perform the following principle operations:

- ☞ to identify and reason about the interferences between plans,
- ☞ to develop solutions for the problem of plan coordination and, thus
- ☞ to modify the plans of other computational agents.

As far as organizational structures and dynamic plan interactions are concerned, the approach of AEGIS primarily benefits from the concept of generalized partial global planning (Decker & Lesser, 1992). As far as plan adaptation issues are concerned, the AEGIS approach uses and extends the plan coordination concept suggested by v. Martial (1992).

4.4 Agent Modeling

We introduce here a brief review of agent modeling techniques. Much research has been conducted within this area. Therefore, within this paper it is only possible to convey the spirit of this work. A detailed discussion of the various approaches to representing agent mental states is beyond the scope of this paper.

In attempting to plan the activities of a group of agents it is necessary to have at least implicit knowledge of other agents that exist within the community together with some awareness of the environment. Such knowledge is mandatory if meaningful agent interactions are to be achieved generally. Much work has been directed at the problem of agent models, attempting to provide a formalism which can characterize itself and its' community. These models need to contain knowledge pertaining to agent capabilities, state, responsibilities and resource requirements. In addition to this there is a need for knowledge regarding agent actions, plans, goals and beliefs. Agents need to be aware of other agent actions and the implications of these on the community as a whole. In order to be able to anticipate or extrapolate agent behavior it is necessary to be aware not only of their current actions but the plan to which they are adhering, their goals and particularly the beliefs they have about the world within which they exist.

Two differing approaches exist to the modeling of agent beliefs, modeling based on possible worlds and modeling based upon a syntactic, logic representation. According to the possible worlds approach an agent knows something to be true if it is true in all the worlds that the agent believes exist. This approach is normally implemented using modal propositional logic and Kripke structures and is typified by the research of Appelt (1981). The later approach as exemplified by the work of Konolige (1986) which grounds on first order logic being supplemented with new operators like `'want'` and `'know'`. A meta-level language is also provided which enables beliefs to be established without having to exhaustively enumerate them. Detailed descriptions of these approaches and associated issues like logical omniscience and shared knowledge are beyond the scope of this work. There has been several noble attempts to integrate the work of planning and agent modeling with formal communication protocols. Both Cohen and Perrault (1979) advocate the use of the 'Speech Acts' first proposed by Austin (1962) and developed significantly by Searle (1969), which include operators like `‘inform’` and `‘request’`.

Bond and Gasser (1988) suggest that systems based upon speech act models firmly based upon plans and beliefs will serve to unify more closely theories of coordination and communication. Within AEGIS we believe that in order to design organizations effectively it is necessary to provide the appropriate apparatus, namely, an expressive medium within which one can encode

agent descriptions, clear inter-agent communication protocols with an associated semantics and an ability for agents to plan their activities.

5 Functional Design of AEGIS

AEGIS/CAS provides a customer consultant with advice as to how a particular customer credit ought to be best arranged. The system architecture reflects a real bank organization with its different departments by modeling each individual agent and each organizational unit in terms of computational agents.

Supposing a complex consulting task requires a cooperative solution, the agents need to take into account both the individual goals of the customer as well as those of the departments involved, and the aims, rules and constraints of the corporate bank. The necessary coordination of decentralized activities is ensured by multi-agent planning, that is on the one hand driven by the desire to achieve the appropriate aims and on the other hand controlled by a mutually benevolent attitude.

The agents of AEGIS are intelligent, autonomous problem solvers, which have their own aims, resources, capabilities and beliefs. Resources are inherited from the organization. Some of their goals may be private (informal goals), others have been defined by the organization (formal goals). Formal goals may interfere with each other and with (local) informal. The definition of formal individual goals must guarantee that the agents perform their actions such that global goals are also to be addressed. This requires that agents exhibit, at least some degree of customer-oriented problem solving behavior.

In case of goal conflicts (within one single agent) each agent behaves cooperatively. That is, the agents involved in a goal conflict initiate a planning process in order to search for an appropriate compromise. This process may also involve bargaining about transfer payments from one agent to another or from the enterprise to an organizational unit.

AEGIS agents are benevolent, in that they tell the truth about their plans, and they are willing to cooperate in order to improve the outcome of the whole system or to resolve conflicts within the system. Thus, the agents have a cordial relationship and do not behave in a hostile manner.

Agents may incorporate the actions of other agents as part of their own plans (multi-agent planning). This requires the agents to exchange their plans before execution in order to allow other agents having access to their anticipated behavior.

5.1 Organizational Model

AEGIS/CAS represents a computational bank. Its internal organizational structure relies on four different types of agents: *(single) agents*, *(informal) social groups*, *(formal) organizational units* and *enterprise*. On this basis, (at least) four different organizational layers can be identified (see Figure 1). Within Figure 1 the enterprise can be viewed as an embodiment of a hierarchy and inter-agent activities. Additional layers can easily be introduced, e.g. if organizational units (recursively) include one or more other organizational units.

To integrate business process orientation into this organizational model we need to distinguish three different types of processes which refer to the four different organizational levels mentioned above:

☞ *(Organizational) procedure*

An (organizational) procedure is a sequence of atomic operations performed by one individual agent. From the viewpoint of an individual agent the operations within an organizational procedure are presumed to be atomic. Organizational procedures are part of the *operative layer* of an enterprise.

☞ *Workflow*

A workflow is a directed graph of procedures on the level of groups thus involving the work of one or more individual agents. Groups can be regarded as "workflow providers". Workflows represent the *operative layer* of an enterprise, too.

☞ *Business process*

A business process can be viewed as a network of workflows (thus involving the work of one or more groups) and procedures (thus involving the work of one or more individual agents) on the level of a single organizational unit. Organizational units in a similar vein can be regarded as "process providers" (see figure 2). Business processes within a single organizational unit comprise part of the *tactical layer* of the orga-

nization; they represent exactly one single product (that is, satisfying basic customer preferences).

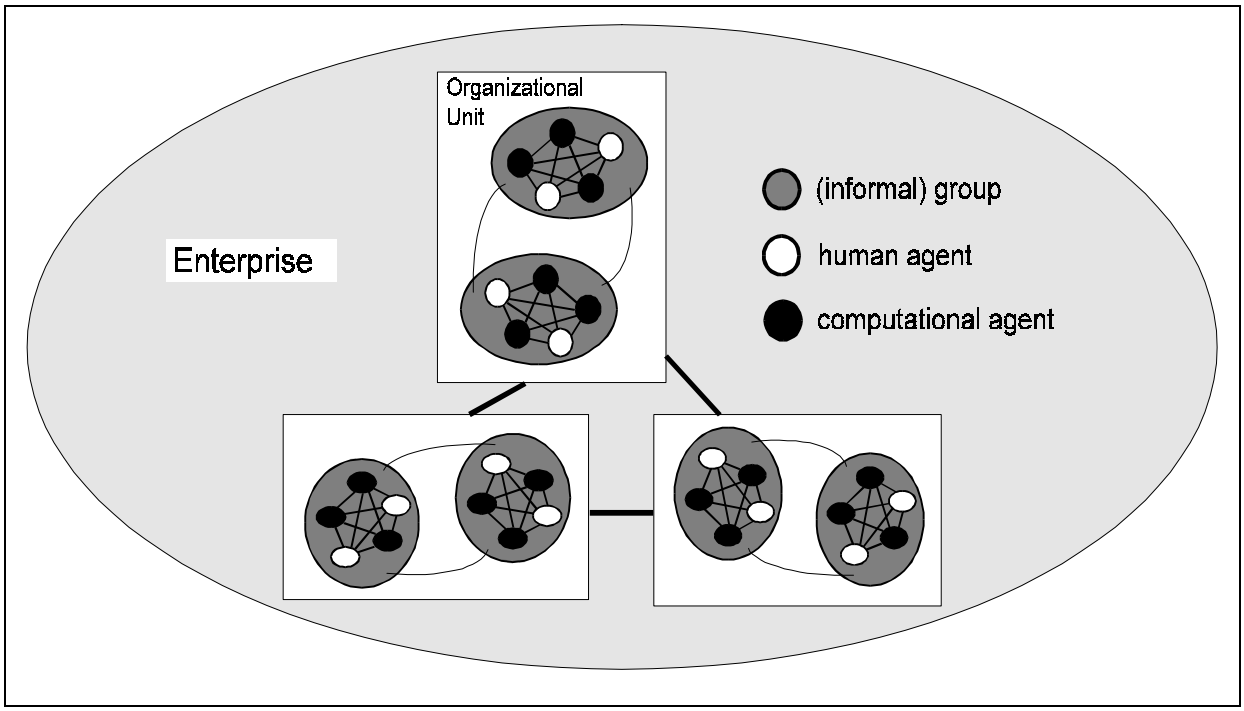

Figure 1: Symbolic representation of a 4-layered organization

Please note, that we use the notion of business processes also to refer to an enterprise-wide network of business processes, which involves one or more organizational units working together on a common problem. In AEGIS an enterprise-wide business process represents a product bundle (that is satisfying complex portfolios of customer preferences). Business processes on the level of the enterprise belong to the *strategic layer* of the enterprise.

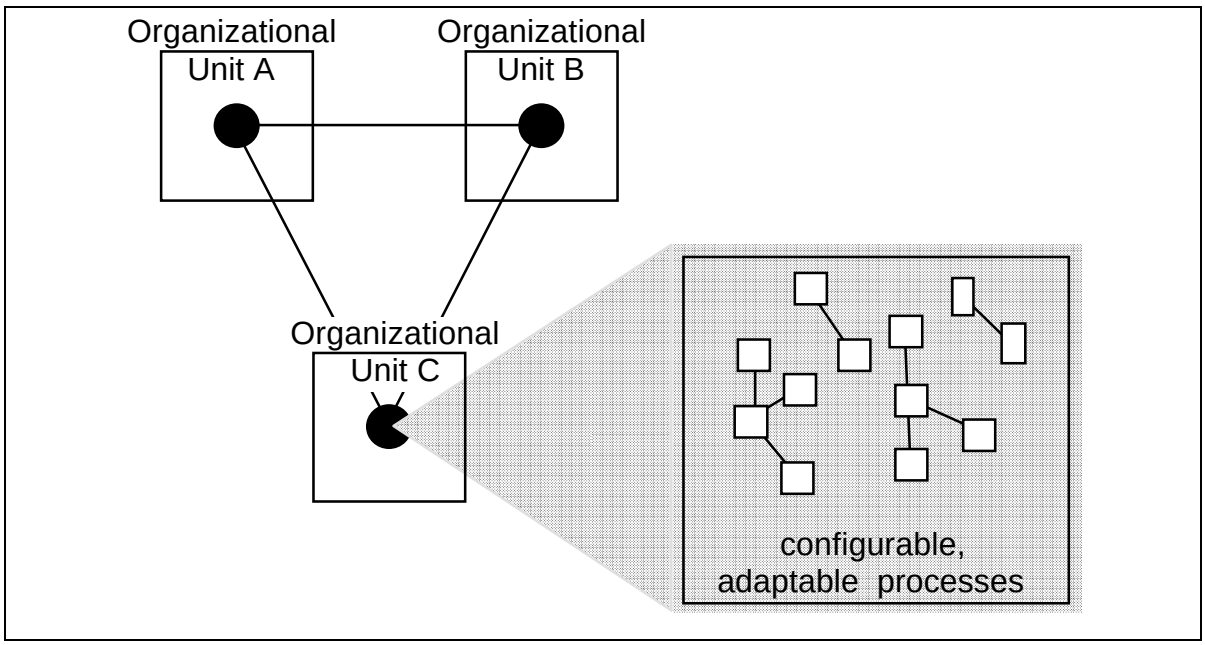

Figure 2: Organizational units representing configurable, adaptable processes

This taxonomy of (organizational) processes necessitates modeling at different levels of abstraction. Organizational procedures typically include a lot of detailed information about particular (atomic) activities, while business processes need abstract away from such operational detail conveying strategic information and thus shielding operational information.

5.1.1 Agent Model

Agents within the AEGIS system can be conceived as an aggregation of several discrete components. Agents consist of a structure such that

```
agent = <agent_name, beliefs, capabilities, acquaintances, commitments,
                                    plans>
```

where

- `agent_name`: represents a unique agent identifier.
- `beliefs`: represents a set of beliefs held by a given agent at a particular instance in time. They encode the model of the environment within which they exist.

- `capabilities`: represents the set of capabilities that the particular agent can provide. Capabilities encode *what* the agent can do not how it achieves it. They are realized through plan execution.

- `acquaintances`: represents the awareness that a given agent holds at any instance in time about colleagues that exist within their environment.

- `commitments`: represents the current set of goals to which the agent is currently committed. The commitments persist until such times as they are no longer relevant to the agent.

- `plans`: represents a set of current plans that are still being utilized by the agent in question. Plans encode *how* agent capabilities are achieved. They may be complete in the sense that the component atomic actions are executed, or they may be partial whereby the plan will be refined further.

5.1.2 Initial Design

Earlier we described the Credit Advice Consultation Process. This process involves several discrete areas of expertise. In our initial version of the credit advisoy system which is a subsystem of AEGIS (AEGIS/CAS), we have identified several agents that contribute to this process. They are: Interface Agent, User Agent, Customer Profile Agent, Corporate Aims Agent, Capital Disposition Agent, Credit Manager Agent, Contract Processing Agent, Credit Design Agent, Credit Rating Agent and Departmental Aims&Objectives Agent. Figure 3 provides a high-level view of the interactions between these agents. Two agents are central within this system, namely the Customer Profile Agent and the Credit Rating Agent.

AEGIS/CAS commences with a stereotypical customer profile and specializes it through consultation to accurately capture the respective customer peculiarities. The customer profile agent maintains the customer profile with respect to the corporate aims and to the information acquired about the customer during the consultation process. This profile is to be exported to other agents that make use of it through instantiating certain variables according to the respective values recorded for the customer in question.

The credit rating agent analyzes whether the customer is credit-worthy and arranges a distribution of the securities for the credit that meet the requirements of the bank. In this context each possible security must be assessed by determining a variety of values (i.e. declaration of value, risk rating, potential win / loss, etc.) in order to come to a profound decision.

Such decisions are based on the customer profile and additional product-specific information provided by the customer, like payback pattern, available securities and so forth. AEGIS/CAS will recommend a particular credit arrangement. This is proffered on the one hand by fulfilling the requirements given by the customer and on the other hand by minimizing the risk of the credit and maximizing the profits of the bank in question.

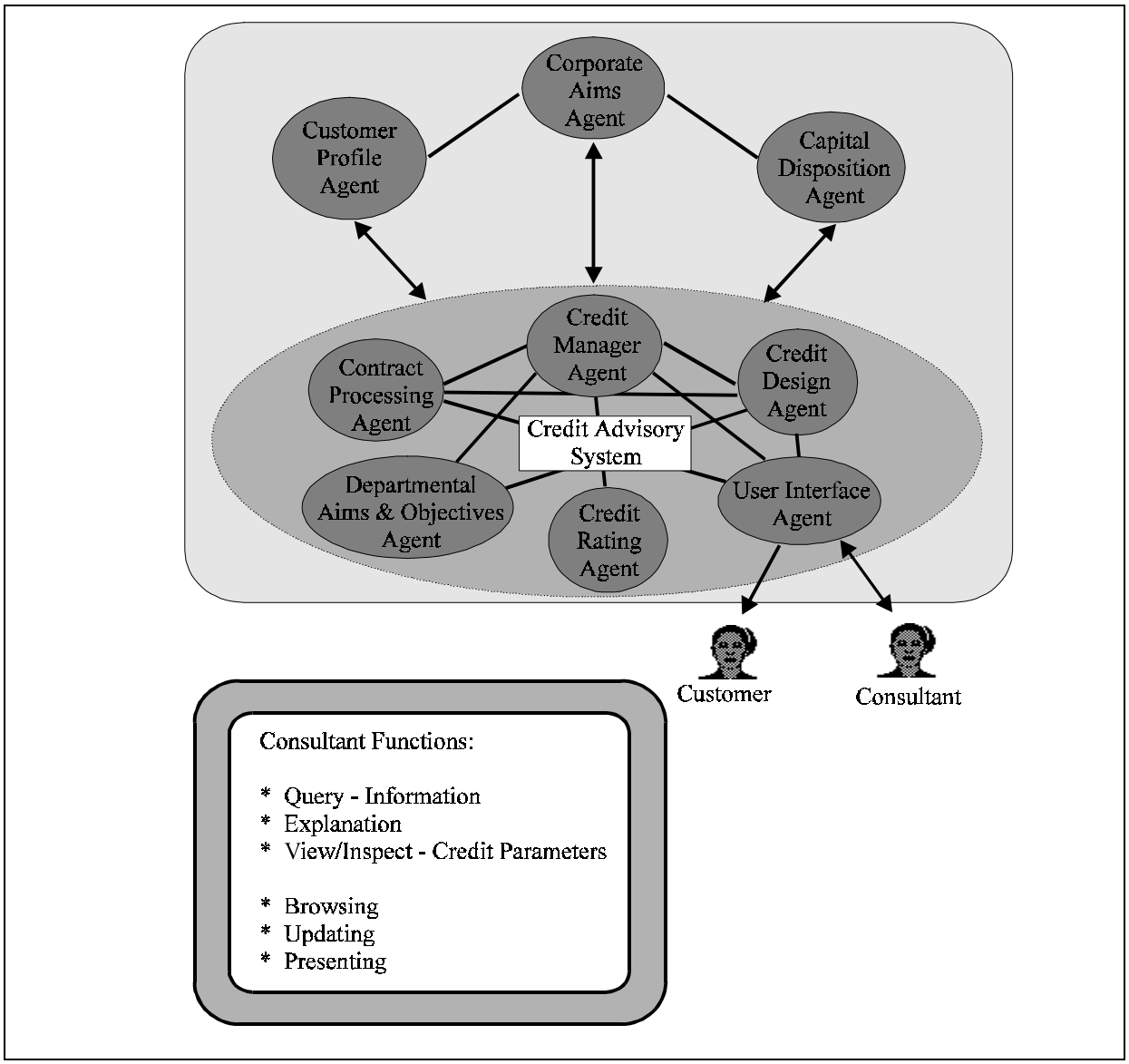

Figure 3: The Credit Advisory System within AEGIS

5.2 Architecture of the AEGIS Multi-Agent Organization

Agent oriented organizations differ from other types of Distributed Artificial Intelligence systems in that they are integrated with human enterprise. Within the context of AEGIS, this enterprise is represented by a bank. Typically, bank organizations involve deep functional (i.e.

product-oriented) hierarchies, in general combined with a second structure, typically that of a geographic hierarchy. The basic structure of a typical bank organization can thus be viewed as a two-dimensional lattice, which exhibits a single central control function namely that of a corporate headquarters.

The design structure of a bank requires organizational designers to develop an appropriate concept of roles (i.e. triples of the form [task, capability, responsibility]) and to assign roles to agents. It also requires designers to establish a network of communication channels, to provide a set of coordination mechanisms, to define control flows within the organization, to implement organizational barriers within the enterprise and between the enterprise and its environment, and to design those constraints that determine to which degree the bank is able to reorganize its internal structure according to the needs imposed by the market and its social and political environment.

The AEGIS multi-agent organization attempts to addresses these with respect to the design of a Distributed Artificial Intelligence system. Thus, the design of AEGIS simulates the functional structure of a bank in that it provides a three-layered, product-oriented architecture: the enterprise layer, the organizational units layer, and the individual agents layer. The architecture of the multi-agent organization represents the decomposition of tasks, the division of labor, and, as such, it serves as a long-term coordination framework (Gasser, 1992a) for the whole system.

The schematic representation of the AEGIS multi-agent organization (see figure 4) can be described as:

- ☞ A global search space, which spans the set of processes provided by the different departments and individual actors together with the set of operators that permit interconnection and modification of these processes.

- ☞ Embedded departments in the AEGIS multi-agent banking organization, which are computational organizational units. These units address to the concept of planning in autonomous systems. They distinguish strategic planning from tactical planning and plan execution (Hertzberger & Groen, 1987). Thus, the *aims and objectives agent* represents the level of strategic planning, the *product design agent* is responsible for tactical planning, and the execution level is represented by the *contract processing agent*.

☞ Each department is able to act as an autonomous, self-contained organizational unit which provides a particular class of products to the market (e.g. different types of credits within AEGIS/CAS).

☞ Each department comprises several agents which exhibit different organizational roles. One of these agents acts as the head of the department, the others provide the necessary (local) functions.

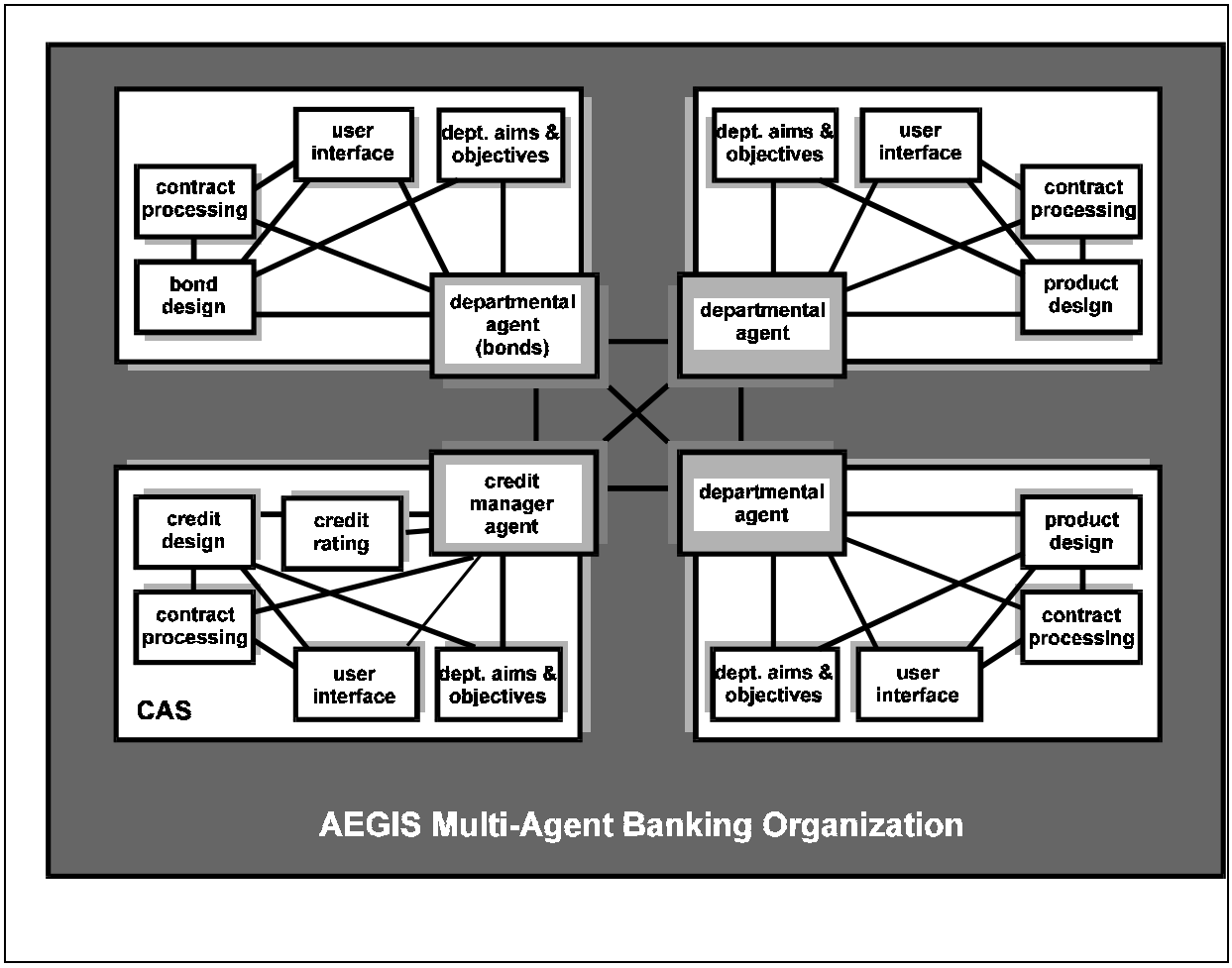

Figure 4: Schematic Representation of the AEGIS Multi-Agent Organizational Structure

☞ *Departmental agents* are allowed to engage the department in a particular task. They represent the departmental interface to its environment, and they act as a coordinating agent (v. Martial, 1992) within their local organizational unit. *Individual agents* plan and act on their own, their plans comprise a set of individual activities (i.e. each individual plan refers to a different organizational procedure).

The problem solving capability of the AEGIS multi-agent organization depends on its ability to select and involve the most appropriate agents for the problem at hand, to instantiate the appropriate organizational structure, and to apply (and to adapt) different types of coordination mechanisms. These criteria define the flexibility of the AEGIS multi-agent organization. Within AEGIS, organizational flexibility is achieved by multi-agent planning, which may be coordinated by the respective departmental agents or which may also evolve as a collaborative activity among single agents being members of different departments. Figure 5 depicts a virtual department that was installed as a result of the creation of a (partial) global plan (i.e. a business process) on the enterprise level. The different agents involved in a virtual department are still connected to their local organizational units; i.e. they are still obliged to approach the respective local aims, to perform the respective local reporting procedures, etc. The reader may note, that this provides for a broad range of virtual organizational structures, depending on the degree to which the different agents (temporarily) leave their standard working environment. In that sense, the representation depicted in figure 5 demonstrates only one type of virtual organizational structures that is possible within AEGIS.

Because conventional organizational theories primarily focus on the structuring of organizational bodies, they prove inappropriate for the organizational model commissioned within AEGIS. AEGIS relies on its ability to dynamically create virtual departments (i.e. task forces) and in turn the creation of these virtual bodies relies on the ability of AEGIS to configure and modify formal representations of business processes which is achieved through multi-agent planning.

Within this section we have concentrated upon the modeling of the more static aspects of our organisations namely that of structure. This structure should merely be viewed as the context within which the activity of the organization is accomplished. We wish, however, also to adequately represent and model this organizational activity. The next section addresses the modeling of processes through plan representation. The intimacy which exists between these two dimensions is of fundamental importance.

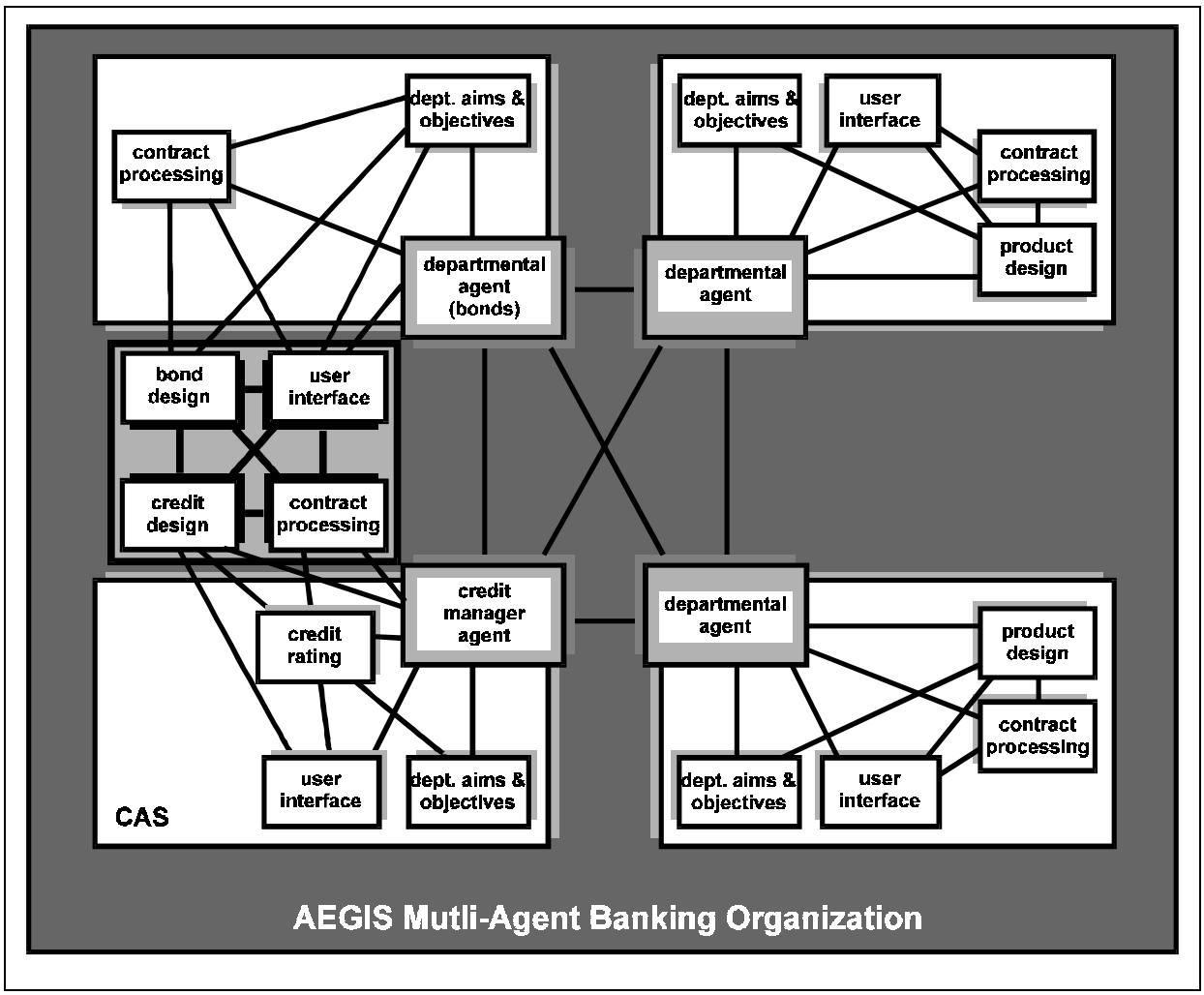

Figure 5: Schematic Representation of a Virtual Department within AEGIS

6 Engineering and Customization of Computerized Business Processes

6.1 Process Modeling Categories for Business Process Engineering

The three different layers within an organization, namely strategic, tactical and operative, may be simulated by a three-layered, vertically integrated, hierarchy of Multi-Agent Plans. Such plans must facilitate *Abstraction on Processes*. In order to support business process re-engineering it is necessary to provide four fundamental process operations. These operations can be categorized as

- modification of processes,
- abstractions on processes,
- interactions among processes, and

- interactions of processes with their environment.

Due to limited space we can merely give an idea of what these operations involve, related work has been presented by Malone et.al. (1993). In the following description plans are represented by upper case letters, while representations of actions (process elements) are represented by lower case letters.

6.1.1 Construction and Modification of Processes

The most fundamental operations demanded is the ability to construct and/or modify a plan. For any given process such operations ought to be able to handle highly complex processes. Such operations must support the construction of new processes from existing ones. Through the judicious assemblage of more primitive processes more complex processes can be constructed utilizing process reuse. AEGIS ensures that pre-existing processes can be modified in order to adapt them to the current coordination situation.

☞ *A* and *B* are composed to make configuration *C*.

A simple example is the sequence.

☞ *A* is removed from *B* resulting in the model *C*.

May be caused, for example, by a reorganization. A, part of B, is to be removed from it.

6.1.2 Abstractions on Processes

In almost every planning domain where complex products have to be constructed anew specification is usually *top down* in nature, i.e. from the more abstract to the more detailed. *Bottom up,* is important for reasons of interactions with other participants. To communicate as efficiently as possible it is necessary to enable the selection of the appropriate level of abstraction so that details which are not relevant to the partner can be hidden. The planning approach developed within AEGIS supports this functionality in such a sense that processes can be abstracted from details or enriched by more specific information, depending on the organizational level from which the process is seen. This means that during planning on a department level a process can be abstracted in a way that only the subtasks that are interfaces to processes of other departments can be considered. This is in contrast to planning on a group

level where it is necessary to show all the subtasks that the individual members are in charge of.

☞ *C* is the refinement of *B*, with $b_2 \in B$ having been refined by *A*.

There is one element which is refined by A. In the next section an example will be given for this operation (refine (a, P´) (P)).

☞ *B* is the generalization of *C*, with *A* having been removed from *C*.

Exactly the inverse operation to "refine".

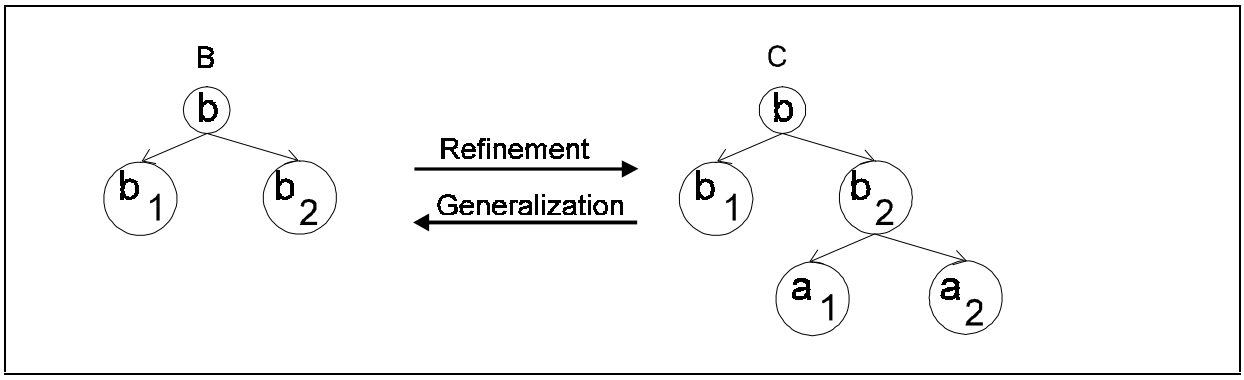

Figure 6: Refinement and generalization of processes

6.1.3 Modeling of Interactions among Processes

When modeling processes it is necessary to adopt a holistic view recognizing that inter process dependencies and interference may occur which must be duly modeled. Thus, process models must enable the explicit modeling of two types of interactions between different processes.

☞ *A* and *B* interact without any coordination.

For instance, this may occur in scenarios like the prisoner´s dilemma, where the actions of the prisoners are interacting without any possibility that the agents coordinate their behavior towards common, or towards their individual goals.

☞ The interactions between *A* and *B* are coordinated.

In other scenarios, the actions A and B of different agents are (implicitly or explicitly) coordinated towards individual or common goals. Well-known examples

refer to concurrent access to a common resource. In such cases, the executions of A and B needs to be coordinated, for instance to be synchronized. In other words: A interferes with the execution of B by means of coordination (et vice versa).

6.1.4 Modeling of Interactions of Processes with their Environment

This category ensures that processes as a whole are explicitly made dependent on certain external conditions (situatedness). In AEGIS this applies for the coordination situation, e.g. the set of currently active problem solving processes.

☞ In a certain situation *s A* must not be executed.

Precondition for A: $\neg s$

☞ In a certain situation *s A* and *B* are exclusive alternatives.

Precondition for A: $\neg s \vee \neg$ *running (B), Precondition for B:* $\neg s \vee \neg$ *running (A)*

In the subsequent section we will illustrate how such processes can be realized in a concrete manner. We advocate the utilization of plans as a formal basis for process modeling and adopt a multi-agent planning approach.

7 Realization of the AEGIS System

7.1 Plan Model for AEGIS/CAS

The operations introduced above are used as a basis for implementing the planning procedure within AEGIS/CAS. In order to have a formal basis for them a plan model is needed that is expressive enough to represent the independent dimensions implied by the first two categories and the requirements implied by the last two.

The basic concepts of our model were derived from a similar formalism first introduced by v. Martial (1992).

7.1.1 Modification of Processes: The Process Dimension

We have to describe processes that are comprised of actions and also knowledge about these actions. Consequently the plans consist of two components. Each component can be viewed as a mathematical set. The first set, `ACT`, contains the actions that the respective process is made of and the membership of the second set, `CON`, is constraints and conditions (e.g. pre-/ postcondition) for the actions together with the relationships (e.g. sequential / concurrent) between them.

```
plan_model´ ::= (ACT, CON)
```

There are two kinds of actions: Actions that are *atomic* with respect to their domain and actions that have to be further refined in order to become operational. The incremental replacement of *non-atomic* elements with atomic actions correspond to process refinement.

7.1.2 Abstractions on Processes and Actions: The Granularity Dimension

The planning procedure mainly consists of repeated refinements of actions that are part of a given process and are as yet non-atomic. This iteration concludes when a process consists exclusively of atomic actions. It may be visualized by a tree, that represents processes at various levels of abstraction. The nodes are actions and the edges show the refinement relationships from more abstract (coarse-grained) to more concrete (fine-grained) actions. So the leaves of the tree can be viewed as the most operational actions reached at this stage of planning. The history of refinement is reflected through the growth of the tree. The operation *refine* corresponds to adding a part of a new leaf level to our tree, whereas the operation *generalize* corresponds to removing a subtree (see figure 6).

To reflect the second granularity dimension as stated above we have to augment the `plan_model´` by the set REF for the refinement relationships:

```
plan_model ::= (ACT, REF, CON) with REF ⊆ ACT x ACT
```

The set REF consists of tuples of actions representing the refinement relation or granularity dimension between those actions (edges in the respective tree).

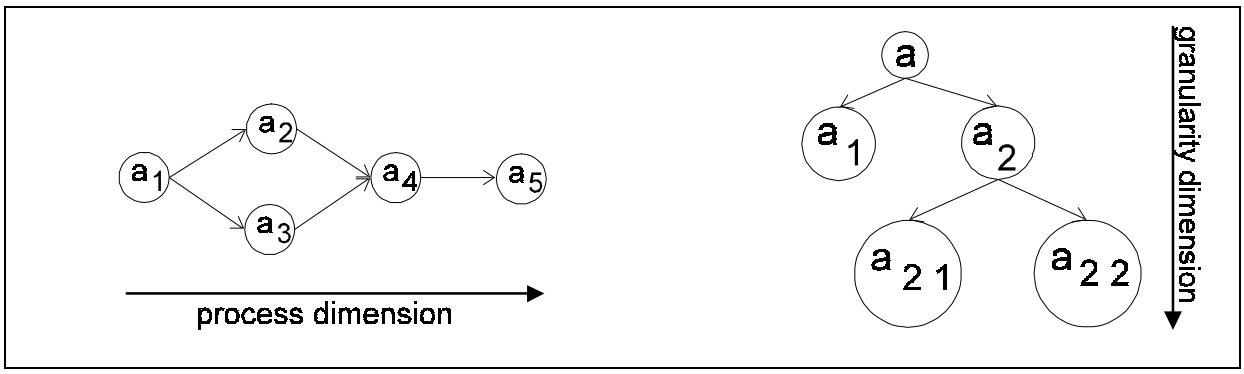

Figure 7: Process model dimensions (v. Martial, 1992)

7.1.3 Modeling of Interactions among Processes

Since interactions among processes necessitate a fine-grained control these interactions have to be modeled on the level of actions (i.e. plan elements). Special atomic actions, for instance `call` and `wait`, can be used in a similar manner to other basic actions of a process.

7.1.4 Modeling of Interactions of Processes with their Environment

The set `CON` contains preconditions for the actions on every level of abstraction of the plan. This is also true for the top level action that represents the entire process. Should no such action exist, i.e. a process is viewed as a sequence of actions at the same level, then a „parent“ action can be added which aggregates this sequence. Thus, a precondition for this action corresponds to a precondition of the process.

The predicate `SEQ(a, b)` describes the sequence of the actions `a` and `b`, the predicate `PRE(a, pc)` stands for a precondition `pc` that must be fulfilled in order to make action `a` executable.

Let `P = (ACT, REF, CON)` be a plan representing process `p` and `pc` be a precondition for `p`. Then, the expression

$$P' = (ACT,\ REF,\ CON \cup \{PRE(a,\ pc)\})$$
$$\text{with root } (a,\ P') \Leftrightarrow \neg\ \exists\, a' \in ACT : (a',\ a) \in REF$$

indicates that the process *p* represented by `P´` can only be executed iff `pc` is true.

7.2 Example: Plan Refinement in AEGIS/CAS

Let `P = (ACT, REF, CON)` and `P´= (ACT´, REF´, CON´)` both be plans. Then the refinement of action *a* by the plan `P´` is defined by

$$\begin{aligned}\texttt{refine (a, P´) (P)} = (&\texttt{ACT} \cup \texttt{ACT´},\\ &\texttt{REF} \cup \texttt{REF´} \cup \{(\texttt{a}, \texttt{a´}) \mid \texttt{a´} \in \texttt{ACT´} \wedge \texttt{root(a´, P´)}\},\\ &\texttt{CON} \cup \texttt{CON´})\end{aligned}$$

The planning procedure will be explained by refining a process model describing the necessary actions to be executed by the Credit Advisory System.

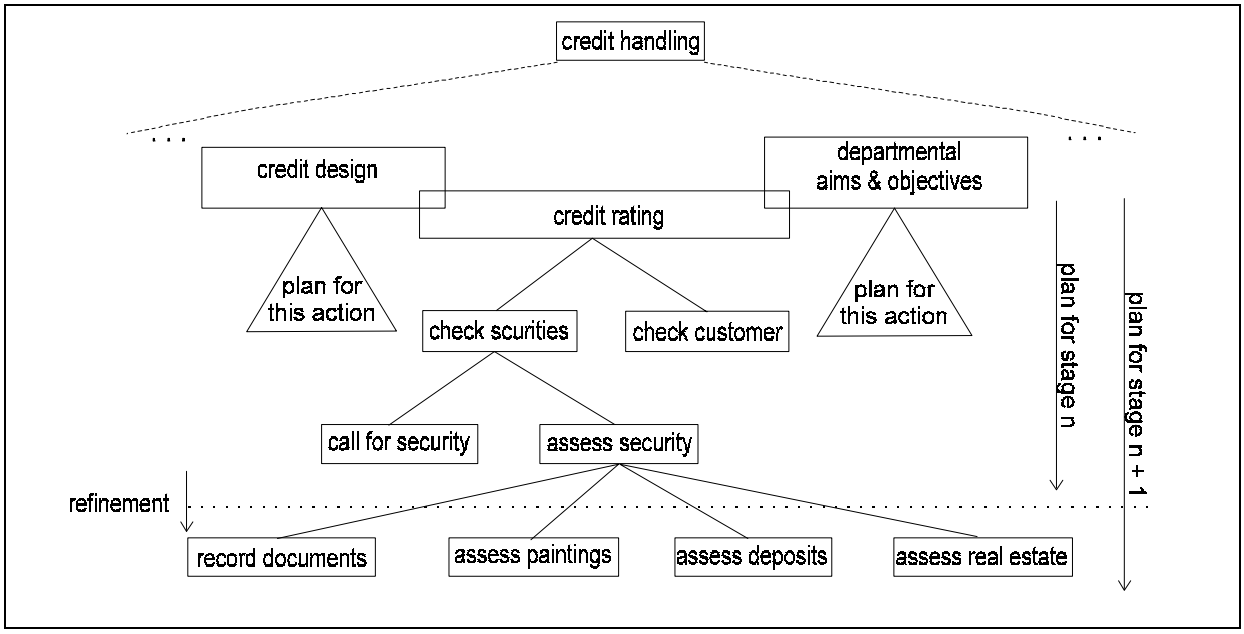

Figure 8: Plan refinement

In order to perform a comprehensive task, e.g. a credit application, that exceeds the capabilities of a single agent a group of agents approach this task cooperatively. This is illustrated in figure 8 by the overlapping rectangles which represent the respective tasks or actions of the credit design agent, the credit rating agent and the departmental aims and objectives agent. The agent we will focus upon is the credit rating agent, a member of the credit department, the head of which is the credit manager agent. We will not elaborate on the necessary consultation process between different agents, but rather on how a single agent produces his local process model for those actions which it wants to perform by itself.

At a given planning stage *n*, the top part of the model presented in figure 8 has been constructed. For our example we will concentrate on the actions `call_for_security` and

assess_security. Call_for_security is atomic because it can be realized by a letter generated automatically. Assess_security may be refined by action record_documents and, depending on the kind of security the customer provided, dynamically by actions assess_paintings, assess_deposits, or assess_real_estate. Because dynamics and context-sensitivity of process model refinement is beyond the scope of this paper, we will not elaborate on this further, instead, within this example, we assume static planning. Therefore, action assess_security can be refined by the action record_documents together with one of the three alternative checking procedures assess_paintings, assess_deposits and assess_real_estate each of them being specialized to the respective kind of security (paintings, real estate or deposits) the customer may provide.

Suppose assess_security ∈ ACT and P = (ACT, REF, CON) is the plan before refinement. The non-atomic action assess_security is to be refined by plan

```
P´ =({ record_documents, assess_paintings, assess_deposits,
       assess_real_estate},
       ∅,
     { SEQ(record_documents, assess_paintings),
       SEQ(record_documents, assess_deposits),
       SEQ(record_documents, assess_real_estate),
       PRE(assess_paintings, painting(security_object)),
       PRE(assess_deposits, deposit(security_object)),
       PRE(assess_real_estate, real_estate(security_object))}).
```

So refine (assess_security, P´) (P) =

```
(ACT ∪ {record_documents, assess_paintings, assess_deposits,
        assess_real_estate},
 REF ∪ {(assess_security, record_documents),
        (assess_security, assess_paintings),
        (assess_security, assess_deposits),
        (assess_security, assess_real_estate)},
 CON ∪ {SEQ(record_documents, assess_paintings),
        SEQ(record_documents, assess_deposits),
        SEQ(record_documents, assess_real_estate),
        PRE(assess_paintings, painting(security_object)),
        PRE(assess_deposits, deposit(security_object)),
        PRE(assess_real_estate, real_estate(security_object))})
```

7.3 Implementation Environment

A first prototype of AEGIS is currently under development which incorporates the plan refinement procedures formulated above. AEGIS is implemented on a network of SUN workstations using HP Distributed Smalltalk which extends ParcPlace VisualWorks. The object oriented philosophy adopted within the project has proven a natural medium for the realization of agent oriented concepts.

7.4 Implementing Agent Interfaces with CORBA

Computational agents in our model are capable of coherent, distributed communication that is independent of the internal structure, resources and capabilities of their underlying information systems. They interact with one another by viewing the network as a single dedicated knowledge object space and ignoring any spatial distribution of objects or agents, respectively (Papazoglou et. al, 1991, pp. 169-202). This implies, first of all, the existence of some standard communication network to interconnect the computational agents and the ability of these agents to communicate by exchanging messages. In our model interconnection is provided on the basis of an implementation of CORBA which can be seen as the communication backbone of the Object Management Architecture (OMA) (Object Management Group, 1992). OMA was defined by the Object Management Group (OMG), which among others wish to standardize (transparent) access to objects within a heterogeneous network. CORBA is the first component of the OMA to be standardized. In the object model of the OMG a request and the appropriate result are the central means of communication between objects (in the terminology of this paper we can equate objects and agents). A request consists of an operation name and the appropriate parameters. Such signatures of operations must be defined with the help of the interface definition language (IDL); i.e. the means of communication are signatures written in IDL.

Objects are identified by a unique object identity. They can react to a number of object-specific messages. After objects have made themselves known to the system, message passing is performed transparently; i.e. the message sending object does not need to know the location of the receiving object nor any implementational issues. If pre-existing information systems are to be included in our network of computational agents, they first have to meet the specification of the CORBA standard.

However, this solves only the problem of pure communication. Agents must not only be able to receive messages; they also must understand the message and react to it. Therefore, we must further attach appropriate front-end extensions to the pre-existing information systems to

enable them to actively participate in a cooperative problem solving process: each pre-existing information system needs to be equipped with some (epistemic) knowledge of the others.

Another issue here concerns the integration of information systems. On the organizational level two distinct concepts of integration can be identified. The first one considers the pre-existing information system as being a passive data storage medium only. That is, the information system is supposed to provide standard data access facilities which can be used by individual agents to improve or broaden their knowledge. The second one refers to the concept of organizational autonomy. It requires the information system to behave as an intelligent computational agent. In that case the integration of the information system requires an agent shell which encapsulates the database (Steiner et al., 1990). As a consequence, each data request needs to be addressed to the respective agent which then decides on its own, whether it is able and willing to participate in a cooperative task and if, how it wants to fulfill the request. Whenever we are concerned with the modeling of business processes and workflows these different concepts of integration play an important role for the task of *organizational* design.

On the technical level the first concept refers to loose coupling of a computational agent and a (set of) data base(s); i.e. loose coupling implies that agent and the databases are autonomous systems which run independently from each other. The second approach is usually bound up with tight coupling between the agent shell and its local data base; i.e. the database and its intelligent shell form an integrated whole.

8 Conclusion

Today's enterprises are challenged by a rapidly increasing demand for flexibility within their internal structures. They need to be able to respond rapidly by customizing their internal processes in order to be able to introduce competitive products into the market place. These requirements have already stimulated huge research efforts in organizational theory, whilst information technology research on how to support customization of computer-based business processes has not yet been adequately investigated.

This paper has addressed the question of how coordination in decentralized, process-oriented organizations can effectively be supported by modern information technology. More specifically, it addresses the issue of cooperative organizational problem solving within such settings in that it suggests to base system design on the concept of multi-agent planning.

The approach presented in this paper allows us to apply a recognized Distributed Artificial Intelligence technique called multi-agent planning as a tool to achieve, at least partially, automated business process (re-) engineering and, via an appropriate design of local and global aims, to proceed towards automatic process customization.

To model and modify multi-agent plans two sets of interaction types (interactions among process models, interactions among a process model with its environment) and two sets of modeling operations (modifications, abstractions), which can be performed on process models, were introduced. The validity of this approach was demonstrated by a credit application example which shows how a multi-agent system can create a multi-agent plan through plan refinement.

The contribution of this paper is threefold:

- ☞ It introduces the concept of *agent oriented organisations* and it illustrates how such an approach could be used within the context of a banking application.

- ☞ It introduces the technical concepts by which this approach can be translated into specifications and implementations of the respective information system architectures.

- ☞ It synthesizes business process orientation and multi-agent planning, and thus demonstrates the applicability of Distributed Artificial Intelligence techniques to business applications.

Future work within the AEGIS project will concentrate on three key issues: (1) further refinement of the agent model, (2) investigation as to how the concept of an organization can be fully realized and in what way is an organization more than merely an aggregation of its agent members, and (3) empirical evaluation of the applicability of the conceptual and technical approach of AEGIS to real-world bank settings.

Acknowledgements

This work has been funded in part through the DAAD Anglo-German Academic Research Grant (ARC, 313-ARC-VII-93/89/scu) entitled Multi-Agent INtentionality (MAIN) and through a Research Grant donated by the German Research Foundation (DFG, Un 94/1-1).

Literature

Appelt, D., E. (1981). Planning Natural Language Utterances to Satisfy Multiple Goals. Phd thesis. Stanford University.

Austin, J., L. (1962). *How to Do Things With Words*, Oxford: University Press.

Bond, A., & Gasser, L. (1988). *Readings in Distributed Artificial Intelligence*, San Mateo, CA.: Morgan Kaufmann Publishers.

Buhl, H.-U., Hasenkamp, U., Müller-Wünsch, M., Roßbach, P. & Sandbiller, K. (1993). Wettbewerbsorientierte IT-Unterstützung in der Finanzberatung. *Wirtschaftsinformatik, 35*, 262-279 (in German).

Chaib-draa, B., Moulin, B., Mandiau, R., & Millot, R.P. (1992). Trends in Distributed Artificial Intelligence. *Artificial Intelligence Review*, *6*, 35-66.

Cohen, P., & Perrault, C., R. (1979). Elements of a Plan Based Theory of Speech Acts. *Cognitive Science, 3*.

Davenport, H. (1993). *Innovating Processes*, Harvard Business Press.

Decker, K., & Lesser, V. (1992). Generalizing the Partial Global Planning Algorithm. *International Journal of Intelligent & Cooperative Information Systems*, Vol. 1, No. 2, June 1992, 319-346.

Fikes, R.E., & Nilsson, N. (1971). STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 3(3-4), 189-208.

Fox, M.S. (1981). An Organizational View of Distributed Systems. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11, pp. 70-80.

Fox, M. S. (1979). *Organisational Structuring: Designing Large Complex Software*. Technical Report CMU-CS-79-155, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Gasser, L. (1992a). DAI Approaches to Coordination. In: N.M. Avouris and L. Gasser (Eds.), *Distributed Artificial Intelligence: Theory and Praxis*, (pp. 31-52), Computer and Information Science Vol 5, Kluwer Academic Publishers.

Gasser, L. (1992b). An Overview of DAI, In N.M. Avouris & L. Gasser (Eds.), *Distributed Artificial Intelligence: Theory and Praxis*, (pp. 9-30), Computer and Information Science Vol 5, Kluwer Academic Publishers.

Hammer, M. (1990). Reengineering Work: Don't Automate, Obliterate, *Havard Business Review*, July-August 1990, 104-112.

Hern, L. E. C. (1988). On Distributed Artificial Intelligence, *The Knowledge Engineering Review*. Cambridge: University Press.

Hertzberger, L.O., & Groen, F.C.A. (Eds.). (1987). *Intelligent Autonomous Systems*. North-Holland.

Huhns, M. (1987). *Distributed Artificial Intelligence*. Los Altos, California: Morgan Kaufmann Publishers.

Kirn, St., & O'Hare, G. M. P. (Eds.). (1995). *Towards The Intelligent Organization: The Coordination Perspective*, London: Springer-Verlag (forthcoming).

Kirn, St. (1995). Competitive Knowledge Processing in Banking: State of the Art and Future Developments. *Cooperative Knowledge Processing: The Competitive Edge in Banking IT. Special Issue of the International Journal of Intelligent Systems in Accounting, Finance and Management.* Wiley & Sons. Issue 2. To appear.

Konolige, K. (1986). *A Deduction Model of Belief* . Pitman/Morgan Kaufmann.

Lesser, V. R., & Corkhill, D. D. (1981). Functionally Accurate, Cooperative Distributed Systems, *IEEE Transactions on Systems, Man and Cybernetics* SMC-11(1), 81-96.

Malone, T. W., Crowston, K., Lee, J., & Pentland, B. (1993). *Tools for inventing organizations: Toward a handbook of organizational processes*, CCS WP #141, Sloan School WP #3562-93. Massachusetts Institute of Technology, Sloan School of Management, Cambridge, Mass.

v. Martial, F. (1992). *Coordinating Plans of Autonomous Agents*. Lecture Notes in Artificial Intelligence, No. 610. Berlin, Heidelberg: Springer-Verlag.

Object Management Group. (1992). *Object Management Architecture Guide.* OMG TC Document 92.11.2, 2nd Edition. Revision 2.0. Framingham.

O'Hare, G.M.P., & Jennings, N.R. (Eds.). (1995). *Foundations of Distributed Artificial Intelligence.* Sixth Generation Computer Technology Series, Wiley Inter-Science, (Forthcoming).

Papazoglou, M. P., Laufmann, S. C., & Sellis, T. K. (1991). An Organizational Framework for Cooperating Intelligent Information Systems, *International Journal of Intelligent & Cooperative Information Systems, Vol. 1*, No. 1, 169-202.

Porter, M. E. (1985). *Competitive Advantage: Creating and Sustaining Superior Performance.* New York: Free Press.

Sacerdoti, E.D. (1977). A *Structure for Plans and Behavior*. New York: Elsevier Publishers, North-Holland.

Scheer, A.-W. (1994). *Business Process Engineering: Reference Models for Industrial Enterprises, 2. edition*, Berlin, et al.: Springer-Verlag.

Searle, J., R. (1969). *Speech Acts: An Essay in the Philosophy of Language.* Cambridge: University Press.

Shoham, Y. (1993). Agent-Oriented Programming, *Artificial Intelligence*, 60, 51-92.

Steiner, D., Mahling, D., & Haugeneder, H. (1990). Human Computer Cooperative Work, *Proceedings of the 10th International Workshop on Distributed Artificial Intelligence.* Austin, Texas: MCC Technical Report ACT-AI-355-90.

Tapscott, D., & Caston, A. (1993). *Paradigm Shift — The New Promise of Information Technology*. New York et.al.: McGraw-Hill Inc.

Warnecke, H.-J. (1991). *Die Fraktale Fabrik — Revolution der Unternehmenskultur*. Berlin et al.: Springer- Verlag (in German).

Werner, E. (1992). The Design of Multi-Agent Systems, In Werner, E., and Demazeau, Y., (Eds.), *Decentralised A.I.3*, Elsevier Science Publishers, 3-28.

Wooldridge, M.J., & O'Hare, G.M.P. (1991). Deliberate Social Agent, *Proc. 10th U.K. Planning Workshop*. University of Cambridge.

Arbeitsberichte des Instituts für Wirtschaftsinformatik

Nr. 1 Bolte, Ch., Kurbel, K., Moazzami, M., Pietsch, W.: Erfahrungen bei der Entwicklung eines Informationssystems auf RDBMS- und 4GL-Basis; Februar 1991.

Nr. 2 Kurbel, K.: Das technologische Umfeld der Informationsverarbeitung - Ein subjektiver 'State of the Art'-Report über Hardware, Software und Paradigmen; März 1991.

Nr. 3 Kurbel, K.: CA-Techniken und CIM; Mai 1991.

Nr. 4 Nietsch, M., Nietsch, T., Rautenstrauch, C., Rinschede, M., Siedentopf, J.: Anforderungen mittelständischer Industriebetriebe an einen elektronischen Leitstand - Ergebnisse einer Untersuchung bei zwölf Unternehmen; Juli 1991.

Nr. 5 Becker, J., Prischmann, M.: Konnektionistische Modelle - Grundlagen und Konzepte; September 1991.

Nr. 6 Grob, H.L.: Ein produktivitätsorientierter Ansatz zur Evaluierung von Beratungserfolgen; September 1991.

Nr. 7 Becker, J.: CIM und Logistik; Oktober 1991.

Nr. 8 Burgholz, M., Kurbel, K., Nietsch, Th., Rautenstrauch, C.: Erfahrungen bei der Entwicklung und Portierung eines elektronischen Leitstands; Januar 1992.

Nr. 9 Becker, J., Prischmann, M.: Anwendung konnektionistischer Systeme; Februar 1992.

Nr. 10 Becker, J.: Computer Integrated Manufacturing aus Sicht der Betriebswirtschaftslehre und der Wirtschaftsinformatik; April 1992.

Nr. 11 Kurbel, K., Dornhoff, P.: A System for Case-Based Effort Estimation for Software-Development Projects; Juli 1992.

Nr. 12 Dornhoff, P.: Aufwandsplanung zur Unterstützung des Managements von Softwareentwicklungsprojekten; August 1992.

Nr. 13 Eicker, S., Schnieder, T.: Reengineering; August 1992.

Nr. 14 Erkelenz, F.: KVD2 - Ein integriertes wissensbasiertes Modul zur Bemessung von Krankenhausverweildauern - Problemstellung, Konzeption und Realisierung; Dezember 1992.

Nr. 15 Horster, B., Schneider, B., Siedentopf, J.: Kriterien zur Auswahl konnektionistischer Verfahren für betriebliche Probleme; März 1993.

Nr. 16 Jung, R.: Wirtschaftlichkeitsfaktoren beim integrationsorientierten Reengineering: Verteilungsarchitektur und Integrationsschritte aus ökonomischer Sicht; Juli 1993.

Nr. 17 Miller, C., Weiland, R.: Der Übergang von proprietären zu offenen Systemen aus Sicht der Transaktionskostentheorie; Juli 1993.

Nr. 18 Becker, J., Rosemann, M.: Design for Logistics - Ein Beispiel für die logistikgerechte Gestaltung des Computer Integrated Manufacturing; Juli 1993.

Nr. 19 Becker, J., Rosemann, M.: Informationswirtschaftliche Integrationsschwerpunkte innerhalb der logistischen Subsysteme - Ein Beitrag zu einem produktionsübergreifenden Verständnis von CIM; Juli 1993.

Nr. 20 Becker, J.: Neue Verfahren der entwurfs- und konstruktionsbegleitenden Kalkulation und ihre Grenzen in der praktischen Anwendung; Juli 1993.

Nr. 21 Becker, K., Prischmann, M.: VESKONN - Prototypische Umsetzung eines modularen Konzepts zur Konstruktionsunterstützung mit konnektionistischen Methoden; November 1993

Nr. 22 Schneider, B.: Neuronale Netze für betriebliche Anwendungen: Anwendungspotentiale und existierende Systeme; November 1993.

Nr. 23 Nietsch, T., Rautenstrauch, C., Rehfeldt, M., Rosemann, M., Turowski, K.: Ansätze für die Verbesserung von PPS-Systemen durch Fuzzy-Logik; Dezember 1993.

Nr. 24 Nietsch, M., Rinschede, M., Rautenstrauch, C.: Werkzeuggestützte Individualisierung des objektorientierten Leitstands ooL, Dezember 1993.

Nr. 25 Meckenstock, A., Unland, R., Zimmer, D.: Flexible Unterstützung kooperativer Entwurfsumgebungen durch einen Transaktions-Baukasten, Dezember 1993.

Nr. 26 Grob, H. L.: Computer Assisted Learning (CAL) durch Berechnungsexperimente, Januar 1994.

Nr. 27 Kirn, St., Unland, R. (Hrsg.): Tagungsband zum Workshop "Unterstützung Organisatorischer Prozesse durch CSCW". In Kooperation mit GI-Fachausschuß 5.5 "Betriebliche Kommunikations- und Informationssysteme" und Arbeitskreis 5.5.1 "Computer Supported Cooperative Work", Westfälische Wilhelms-Universität Münster, 4.-5. November 1993

Nr. 28 Kirn, St., Unland, R.: Zur Verbundintelligenz integrierter Mensch-Computer-Teams: Ein organisationstheoretischer Ansatz, März 1994.

Nr. 29 Kirn, St., Unland, R.: Workflow Management mit kooperativen Softwaresystemen: State of the Art und Problemabriß, März 1994.

Nr. 30 Unland, R.: Optimistic Concurrency Control Revisited, März 1994.

Nr. 31 Unland, R.: Semantics-Based Locking: From Isolation to Cooperation, März 1994.

Nr. 32 Meckenstock, A., Unland, R., Zimmer, D.: Controlling Cooperation and Recovery in Nested Transactions, März 1994.

Nr. 33 Kurbel, K., Schnieder, T.: Integration Issues of Information Engineering Based I-CASE Tools, September 1994.

Nr. 34 Unland, R.: TOPAZ: A Tool Kit for the Construction of Application Specific Transaction, November 1994.

Nr. 35 Unland, R.: Organizational Intelligence and Negotiation Based DAI Systems - Theoretical Foundations and Experimental Results, November 1994.

Nr. 36 Unland, R., Kirn, S., Wanka, U., O'Hare, G.M.P., Abbas, S.: AEGIS: Agent Oriented Organisations, Februar 1995.