The Comandos Toolset for Distributed Systems Management

GERRIT KERBER, HELMUT MEITNER AND FRIEDEMANN REIM

Fraunhofer-Institut für Arbeitswirtschaft und Organisation, Nobelstraße 12, D-70569 Stuttgart, Germany

The management of distributed information systems is a complex and time consuming task. Concepts and tools to support experts with the design and operational control of such systems are summarized. An adaptive approach to system management is described, which is needed to handle changing demands during system evolvement. Relevant management tasks are identified and tools and facilities to support these tasks are presented. This comprises system observation and system control facilities, a system design tool, a risk management tool and a system administration tool.

Received May, 1994

1. INTRODUCTION

The major goal of distributed systems management is to achieve a performance optimal, secure and reliable configuration of the system that fulfils the requirements of the business. Distributed systems management comprises all activities for collecting information from the existing system, modelling system behaviour evaluating configuration alternatives, taking design decisions and modifying the system configuration.

The various aspects of management are dealt with by interrelated tasks including organizational design, application management, configuration management, security management, network management, service management, load balancing, short-term system monitoring, long-term system observation and system control. Attributes of distributed systems, such as separation, autonomy, heterogeneity, flexibility and transparency (Geihs, 1993), have to be considered in each of the management tasks.

Distributed systems management has been addressed by several national and international research projects. In the ISA-DEMON project, the focus is laid on short-term system monitoring. Basic mechanisms for event ordering and a configuration language for graphical visualization are provided (Hoffner, 1992). In the COIN environment, the graphical short-term observation of distributed applications is supported by the visualization of animated interactive event dependency and object/threads graphs (Buhler and Sturm, 1991). In Domains, management is considered as a distributed activity itself. Structuring principles and runtime support for a distributed architecture are given (Fink *et al.*, 1993). In Domino large-scale distributed systems are considered. Therefore the conception of management domains and their structuring are supported. In REX, common aspects of management both of parallel and distributed systems are regarded. A configuration language for structural description and support for dynamic configuration and change management is given (Kramer *et al.*, 1992). The constantly growing demand for management support of distributed systems has lead to the OSF DME release. Here, existing products of multiple suppliers are being put together to support system managers with services for software distribution, printing, license management, etc. No explicit, coherent management model is established.

ESPRIT project 2071 Comandos (Construction and Management of Distributed Open Systems) took an adaptive approach for the management of distributed systems that allows the original design of the infrastructure to be modified as experience is gained and as user requirements towards the operating environment change (Ness, 1990). The approach is based on an object-oriented model of the information system. The basic management tasks, three distinct tools and their integration into the running system are described in this paper: Disdes (a tool for organizational design), RiskMa (a security management tool) and UsrAdm (a tool for system administration). To interact with the running system each of the tools is able to use supportive facilities for system observation and system control.

2. ADAPTIVE MANAGEMENT APPROACH

Most approaches to organizational design (Grochla, 1982) and to information systems design (Lockemann and Mayr, 1986) have been oriented towards a life-cycle model and a project organization with phases for action. This puts most emphasis onto the early stages. However, these approaches tend to neglect the use and operation of a distributed open system after implementation. For distributed systems with continuing performance over a long period of operation and a dynamic environment a new line of thought should be followed. This calls for design and management approaches that explicitly surrender the assumptions underlying a phase oriented approach (Floyd, 1981). The adaptive system management approach of Comandos (Horn *et al.*, 1988)

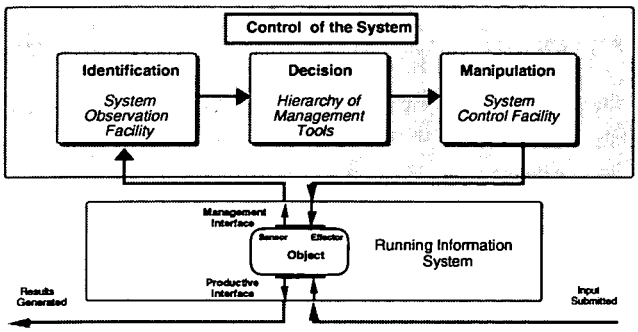

FIGURE 1. Adaptive framework for system design and management.

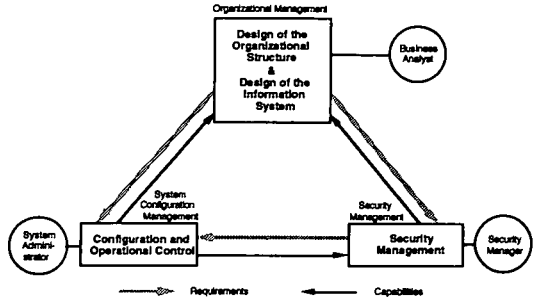

FIGURE 2. Basic management tasks addressed in the Comandos project.

integrates the ability to change into the distributed open system (Figure 1).

The adaptive framework for the management of Comandos systems can be mapped onto three generic types of activities. Observation activities are concerned with collecting information while the system is running. The decision activities support the actual design, the configuration and the security management decisions. They are carried out by a human designer or manager using an interactive management tool, specific to the task. The control activities realise a design or configuration decision, i.e. the implementation of a change. These activities are supported in Comandos by a set of cooperating management tools. Every object in the distributed information system is viewed as having two different kinds of interfaces, productive and management. The productive interface provides the functionality required by the system. The management interface allows observation and modification of the object.

3. BASIC MANAGEMENT TASKS

A distributed information system must be structurally responsive to changes in its environment and allow for a selective degree of distribution transparency.

Information system management is often viewed as a set of individual design processes. These processes differ in focus. They may take an organizational or a technical view. In distributed information systems a specific view on security is also recommended. Within each process, issues of an overall information processing (IP) strategy that, for instance, determines the desired degree of decentralization, have to be addressed. On the basis of this IP strategy, issues of organizational structure, technology selection and its economy, installation and implementation have to be dealt with. The number of design alternatives is reduced when proceeding from IP strategy selection to operation of the system.

Three types of design tasks are distinguished here: organizational management, configuration management and security management.

Organizational management is responsible for the appropriate design of the organizational structure, i.e. its engineering, which in turn determines the required functionality of the distributed information system.

The *configuration management* consists of hardware configuration and system administration that is more oriented to logical concepts than physical components. A system administration model comprises persons, users, groups, accounts, hosts and home directories.

Security management has to achieve and enforce integrity, availability and confidentiality of information. For distributed information systems, availability of information is a crucial point. Risks due to failure of system components have to be analysed and evaluated.

All these design processes are strongly interrelated. A conceptual and technical integration among these processes is required.

Several methods and tools exist that support different design tasks for distributed information systems. However, no tool or set of tools covering the range from organizational design to configuration management exists that is integrated into the distributed information system itself, thus capable of fully exploiting its power. OSI management tasks address management issues at the level of each of the seven layers. They are not designed to integrate the management tasks mentioned above.

Comandos provides an infrastructure and tools to construct and manage distributed information systems. It comprises a set of tools supporting management decisions on the general design as well as administration activities that transform these decisions into an operational distributed system (Balter, 1989). The tools are coupled, thus integrating the various design processes. The basic management tasks supported by computer based tools are depicted in Figure 2.

Two types of relationships exist between the management tasks: requirements and capabilities. The posing of the requirements sets the goals to be achieved by the other basic task and determines the major dependency direction in the hierarchy of the basic tasks. However, dependency is bi-directional because the basic management task posing a requirement must be informed about the capabilities of the other basic tasks in order to pose realistic requirements and to fully use the potential.

A general description of the implementation architecture for the Comandos tools is given in Reim (1991).

4. MANAGEMENT TOOLS AND FACILITIES

An integrated toolset to support the basic management tasks has been developed (Fig. 3). The Disdes tool

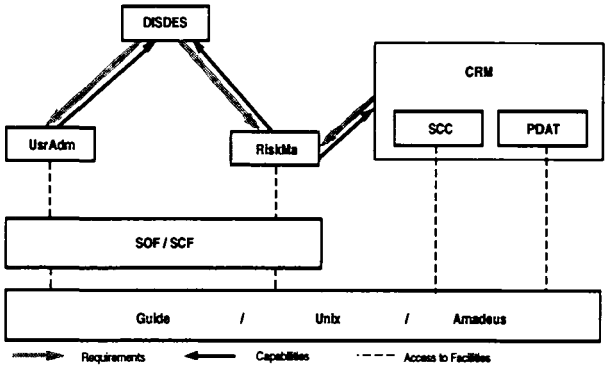

FIGURE 3. Relationships between management tools and facilities.

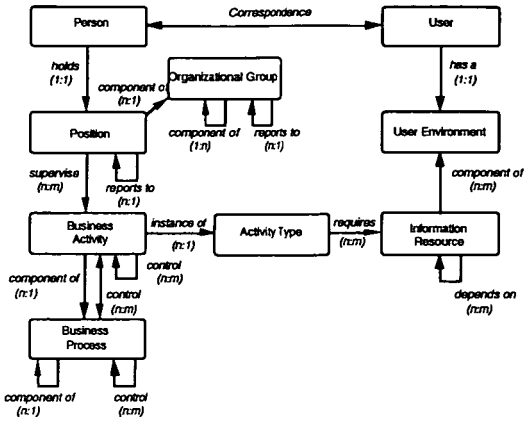

FIGURE 4. Model elements of the organizational and information system model.

supports the design of the organizational structure and the design of the information system. Configuration and operational control are supported by the UsrAdm tool. Support for security management is provided by the RiskMa tool. RiskMa is related to two other tools of a Comprehensive Risk Management (CRM) which are not in the scope of this paper. The Security Control Centre (SCC) allows to build factual models, requirement models, assessment models and decision models to investigate security features of a distributed system (Fichtner and Persy, 1992). The Protocol Data Analysis Tool (PDAT) applies filtering techniques to analyse audit data (Weiss and Baur, 1990).

To interact with the underlying system, supportive facilities are used by the management tools RiskMa and UsrAdm. The underlying system can be based on Unix or on the Comandos kernels Guide or Amadeus (Cahill *et al.*, 1993; Decouchant *et al.*, 1988). The System Observation Facility (SOF) and the System Control Facility (SCF) allow to retrieve information from objects in the underlying system and to control them.

4.1. Management tools

4.1.1. Distributed information system designer tool (DISDES)

The organizational design is based on a process-oriented view on business activities (Bracchi and Pernici, 1984). Activities and processes are the basic elements for modelling the flow of business tasks. Actor-oriented features also are represented, thus allowing the modelling of capacity aspects. Figure 4 shows the model entities and their relations.

Business Activities are atomic tasks carried out by a position without interruption. *Business Processes* are sequence of a number of possibly parallel activities or (sub-) processes with a unique start and a unique end activity. *Positions* are the only organizational units capable of carrying out activities; they are held by humans. Relationships may exist between the entities. The *Control* relation represents the control flow between activities. *Supervise* poses responsibilities for carrying out activities upon positions.

The organizational model furthermore comprises *Organizational Groups and Persons*. In order to enable the configuration of a distributed information system, the above entities are conceptually related to the concepts for describing the distributed information system. This is achieved by typing the activities. The *Activity Types*—they also could be called activity classes—represent a classification of business work. Each activity type poses requirements (*Requires* relation) upon an *Information Resource* to be installed. Every *user* known to the information system *Has A User Environment* that consists of a set of information resources.

The model represents the conceptual core of *Disdes*. *Disdes* is a decision support system for the organizational designer and business analyst. His responsibilities include the selection of the information system components appropriate to support the business tasks.

The decision support tool Disdes supports the following tasks:

- Representation and generation of a model of a company and its distributed information system: The user of the tool benefits from the graphical, interactive user interface of Disdes. Cut and paste features ease the generation of alternative solutions.
- Assessment of alternative design solutions. The assessment components of Disdes allow the evaluation of the performance of proposed alternative solutions to a distributed information system design problem. The dynamic analysis is based upon a discrete simulation on the office model.

The design of the organizational structure and of the distributed information system has no direct coupling to the operational system. Coupling to the operational information system is achieved through a *Requirement Generator*. This component generates a specification of the required functionalities and gives hints on how to configure the distributed information system, e.g. by identifying the location of primary usage of software and data. Thus knowledge available during the organizational design process is made available directly to

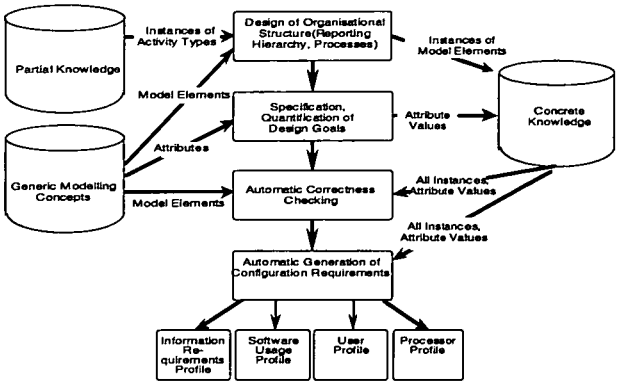

FIGURE 5. Necessary steps to generate requirements by using Disdes.

configuration management. The underlying model (Figure 4) allows for a selective level of detail—with respect to organizational as well as information system aspects.

Disdes supports the derivation of the required information system functionality including hints about expected usage patterns. To obtain requirements, Disdes needs tailoring to the specific enterprise. The information resource components available within the enterprise need to be loaded into the model and graphical view management systems of Disdes. This is mainly an initial task. Subsequent modifications can be done graphically interactive as they are needed.

In a similar way, a set of activity types has to be loaded into the model and graphical view management system. The activity types represent a classification that is used to handle the huge number of individual activities in an enterprise. It eases the assignment of information resources to activities. Without classification, assigning information resources to activities would be a prohibitively costly task. In accordance with the CIM-OSA reference model this kind of information is called partial knowledge (Stotko, 1989). Figure 5 depicts the steps necessary to obtain requirements derived from the organization model (Reim, 1992).

Table 1 summarizes the content of the various profiles generated by the requirement generator. Requirements for security management are passed in a similar way.

TABLE 1. Requirements generated by Disdes

Information requirements profile	How frequently and at which location are the data accessed/updated?
Software usage profile	What are the intensity and the location of software usage?
User profile	What users have to be installed? What access rights do they need?
Processor profile	What processing power is needed in order to guarantee response times?

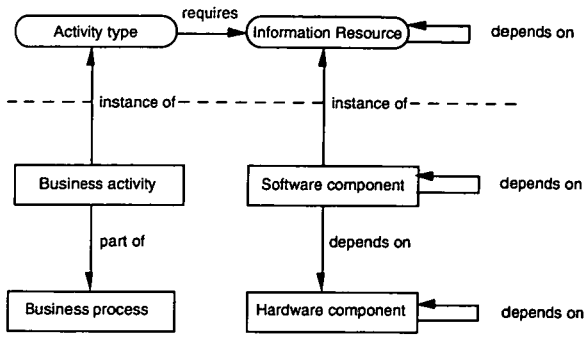

FIGURE 6. Relationships between the elements of the RiskMa model.

4.1.2. Risk management tool (RiskMa)

Although the general risk management approach is broader (Meitner, 1990), risk management in Comandos concentrates on determining the impact of system component failure on the continuation of business activities and business processes designed with Disdes. Instances of the leafs of the information resource type hierarchy are software components (Figure 6) like application programs or data. Software components depend on hardware components in order to operate. Hardware components are physical entities like processors, disk partitions and transceivers. They can be defined at different levels of abstraction. For example, a host can be considered as a hardware component consisting of processor, main memory and disk partitions.

Business processes and business activities are considered as assets. An asset is a valuable object and has an associated profit loss value indicating the loss caused by unavailability or destruction of the asset. Assets can be endangered by threats. Failue of software components and failure of hardware components are considered as threats because they may lead to failure of a business activity. The relationship between assets and threats is many to many. One threat can endanger several assets as well as an asset can be endangered by several threats. A threat causes a damage with the probability of a damage event. Because a threat can endanger several assets the damage of a threat is calculated at each asset. The severity of the damage gives the degree to which an asset is endangered by a threat. In the case of failure the failure duration can be regarded as the severity. A risk exposure due to a threat is defined by the product of the damage caused by the threat, the probability of the damage event and the severity.

To diminish a risk, countermeasures have to be taken. A risk can be diminished by reducing the damage, the probability of the risk event and/or the severity. Reduction of the probability for software component failures, for example, can be achieved by increasing the degree of redundancy.

RiskMa provides support for model generation, analysis of threats, evaluation of risks for the continuation of business processes and system modification (Table 2). Dependability measures for the running

TABLE 2. Results generated by RiskMa

Tool support	*Results*
Model generation	Imported business processes and activities from Disdes Consistent model of hardware and software components Dependencies between business activities and software components
Threat analysis	Dependability statistics on the running system Impact of failures for business activities
Risk evaluation	Quantitative dependability measures for business activities Qualitative estimates for business activity assets
System modification	Modified configuration

system are provided by a specific SOC generating statistics based on failure events. The statistics for hardware and software components include number of failures, mean failure duration, mean time between failures and availability.

The system model can be interactively built using different views. The hardware component view shows the relations between the hardware components. The software component view shows the relations between the software components and the hardware components, and the relations between the software components themselves. The process flow view shows the relations between the business activities and the software components.

Risk evaluation for specific assets is supported by static and dynamic analysis. In static analysis the state of the entire system is evaluated according the component states defined during model generation. As a result the risk exposure for a process can be estimated. Dynamic analysis allows to simulate the state transitions of the hardware and software components using a dependability graph. Failure rates given for each component can be real rates observed in the running system or rates given by the security manager.

If the risk involved in a system configuration is unacceptably high the configuration has to be changed. Changes of the system configuration can be replication of an application, moving of application software, or making hardware resources redundant. The change order is given to the SCF which realizes the change by triggering effectors.

4.1.3. User and host administration tool (UsrAdm)

Administration of users, groups and hosts is a costly task due to number of items involved and the lack of knowledge about the user preferences. UsrAdm supports the system administrator in understanding the problem, modelling, inventing and developing a better configuration. Representation and manipulation of the system administration model are supported.

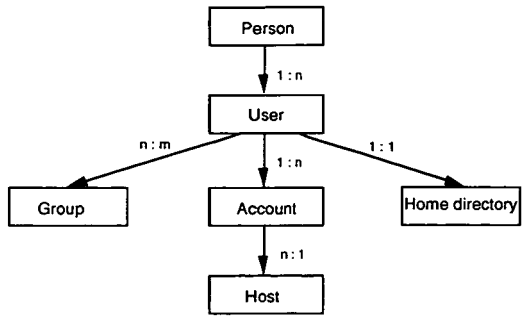

FIGURE 7. The system administration model.

The system administration model comprises persons, users, groups, accounts, hosts and home directories (Figure 7). A host represents a physical computer like a mainframe, workstation or personal computer. A user represents a natural person. Each person allowed to work with the system has at least one associated user. Users get accounts on several hosts. A user is only allowed to work on a host if he possesses an account on that host. Each user can have one or more accounts. The home directory of a user is a directory in which all the objects for that user are available. A user has exactly one home directory, although he has several accounts. Some users together can be addressed as a group. Users being in the same group have all the access rights of that group.

On the basis of the model, statistics on users and hosts can be produced. Information on users and hosts is generated and stored by the SOF. The query interface of the SOF is used by UsrAdm to retrieve host- and user-related information. Host-related information includes statistics on processor load, usage of disk partitions as well as server and client calls within the network file system. User-related information includes statistics on user activities on hosts and size of the user home directories (Table 3).

On the basis of the model and the retrieved statistics, modifications within the architecture of the distributed system can be decided to optimise the configuration. SCF is used to implement the changes into the running system.

4.1.4. Integration among the tools

The relationship between the tools is defined with the formulation of requirements and capabilities. With the analysis and evaluation of model data in Disdes, requirements for administration of users and hosts and

TABLE 3. Results generated by UsrAdm

Tool support	*Results*
Model generation	Model of network segments Model of users and groups
System observation	Host-related statistics User-related statistics
System modification	Modification of host configuration Relocation of user home directories

for risk management can be expressed.These requirements are interpreted by the UsrAdm and RiskMa tool and are the basis for further operations like implementation of changes with the usage of the SCF facility or the definition of dependability graphs.

Information about the model elements consists of an aggregation and interpretation of attribute values of elements of the generated models. For the formulation of *requirements* to the UsrAdm tool, this is for instance a list of names of persons with certain attributes generated from the Disdes model. This information can be used for the modelling of accounts, users and home directories for administration purposes. Together with information about required information resources and locations this is used for management decisions like the placement policy of user home directories. Requirements for the building of user groups in a distributed information system can be derived from the grouping according to the organizational criteria. For example, the building of a task force with the definition of specific tasks to be performed can be used to specify a user group within an information system with necessary access right on information system resources supporting these tasks. Positions in the Disdes model are used to represent hosts of the UsrAdm model. Therefore, information about potential bottlenecks that are retrieved from static and dynamic assessments of the Disdes model can be the basis of decisions how to balance system resources, e.g. where to place user home directories. Analogously, information on business, processes, business activities, activity types and information resources is interpreted by the RiskMa tool to model and assess dependabilities of the information system.

Information about model data of UsrAdm and RiskMa can be the basis for the formulation of *capabilities* that are interpreted by Disdes and can be used to restrict the modelling of the distributed information system and the assessment of this model with respect to these capabilities. Information can for instance be obtained with the connection of the UsrAdm tool to the SOF facility. With that, data about usage and balancing of system resources like computing capacity and storage capacity can be collected. This information can be the basis for specification of Disdes model element attributes like the capacity of positions. Information about groups and persons can be also basis for the formulation of capabilities. Data about user imposed load on system resources obtained with the SOF facility is the basis of the assignment of users to groups and therefore for organizational regrouping. Positions can be used to represent the capabilities of hosts, like a certain capacity. Therefore information about hosts gained from UsrAdm can be used for the formulation of constraints on the modelling of positions in Disdes. Capabilities expressed from the RiskMa tool are for instance dependability measures for business activities that can be used as parameters during the dynamic simulation with Disdes.

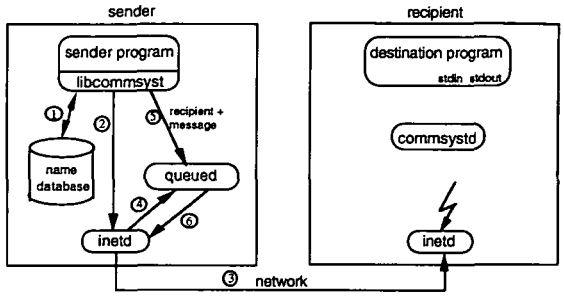

FIGURE 8. Working method of BCS with reliable data transmission.

4.2. The facilities for system observation and control

4.2.1. The bi-directional communication service (BCS)

A distributed communication service is required to collect data from various hosts in the distributed system and to communicate statistics. The communication service must guarantee that no data is lost and that ordering is maintained during transmission. To meet this requirement the BCS has been specified and developed. It consists of a library to be linked to the sender program that comprises the functions for establishing a connection, sending data and closing a connection. A demon at the receiver side is started upon request and invokes the destination program.

In case of a failure during transmission an additional demon program is started on the sender host and administers the message in a queue (Figure 8). Periodically, it tries to transmit the messages again. As long as the queuing demon exists all messages to the same destination program are forwarded to the queuing demon to ensure the correct sequence of the messages. If the connection to the recipient host is up again, the messages in the queue are sent in the correct sequence.

The name database hides the distribution of the destination programs. Destination programs are identified by logical names resolved by the name database. A logical name is mapped to a triple consisting of host name, name of destination program and arguments for the destination program.

4.2.2. The system observation facility (SOF)

The SOF carries out the identification function of the adaptive approach for systems design and management. It is a service in a distributed information system to collect, to aggregate and to make available information for the management of the distributed system. SOF itself is distributed in the running system and uses the facilities provided by the execution environment.

The SOF consists of a set of SOCs and a set of sensors. Sensors are installed on every observed host to generate specific information on objects to be managed. They are specified by the host to be observed and the type of information to be generated. Every sensor in the distributed information system is identified uniquely and reports to exactly one SOC. When installing the sensor, the name of the corresponding SOC must be known. This

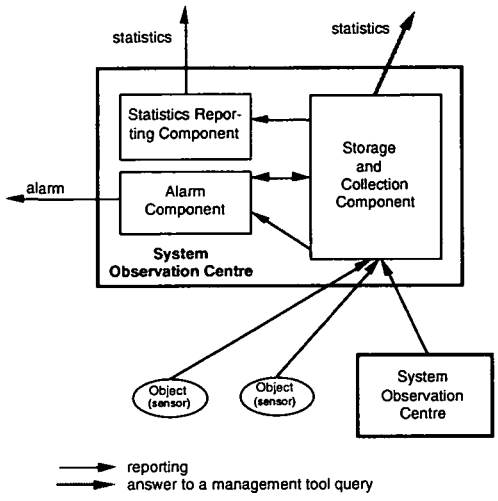

FIGURE 9. Structure of a SOC.

name is distinct in the management domain and is maintained in the name database of BCS. SOCs collect, aggregate and store information received from sensors (Figure 9). Information can be retrieved from the sensors and from other SOCs. Aggregated information is provided for higher-level SOCs, for management tools and for system operators.

The types of information are statistic and alarm. Statistic is information that is produced through data aggregation to update the behaviour model of the distributed system. Statistics stored in the SOC is provided to higher-level SOCs periodically and can be queried on request by management tools. Alarm is information that describes a condition of the information system which requires immediate action from an operator. It is event driven and initialized if specific data is generated by the sensors and sent to the SOC. Predefined alarm conditions are checked by the Alarm Component. An alarm is produced and reported to the operator through an alarm mechanism. Furthermore, statistics are produced on the occurrence of the alarms.

At least one SOC is required in each SOF configuration. A hierarchy of SOCs can be built if more than one SOC is used. The hierarchy reflects the level of aggregation of information. The low level SOC receives information by the sensors and reports it, possibly

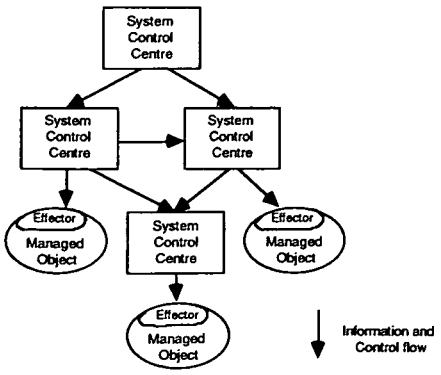

FIGURE 10. Hierarchical structure of the SCF.

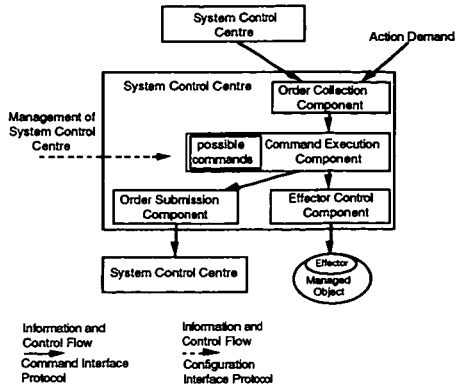

FIGURE 11. Structure of the SCC.

aggregated, to a higher SOC. Additionally, every SOC within the hierarchy is able to receive information from sensors and provide information for management tools. The hierarchy is built up as directed graph. Closed loops are not allowed.

4.2.3. The system control facility (SCF)

The SCF is a service in a distributed information system. It allows to carry out structural changes in the information system. The changes can be composed of several steps and affect various locally distributed objects.

The SCF is composed of three kinds of components:

- Effectors are part of the managed objects within the information system
- System Control Centres (SCC) for execution of commands and control of effectors
- A distributed service for command transfer between SCCs and effectors.

The SCF is structured as a directed, acyclic graph. Administration and management tools issue modification action demands to a SCC, which either directly accesses an effector in a managed object or issues an action demand to a lower-level SCC. The hierarchical structure of the SCF is shown in Figure 10.

The SCCs within the SCF maintain the distribution of management orders throughout the information system. Orders can be collected and executed within an SCC. Sequences of orders need to be transaction oriented. Different SCCs can be of different degree of specialization. The resulting hierarchy corresponds to the levels of abstractions of orders in the distributed system. Figure 11 shows the Structure of an SCC.

Incoming orders are stored temporarily and serialized in a FIFO queue. Sequences of orders are executed as transactions. The transaction is reduced to orders within a SCC. In multistage SCC hierarchies transactions can contain intermediate commits.

The instances that place execution orders have to wait for the termination of a transaction. In Figure 12, the waiting condition is shown as a petrinet (Clauss, 1991).

An order that cannot be executed immediately (i.e. there is no additional token available at transition T_{i1},

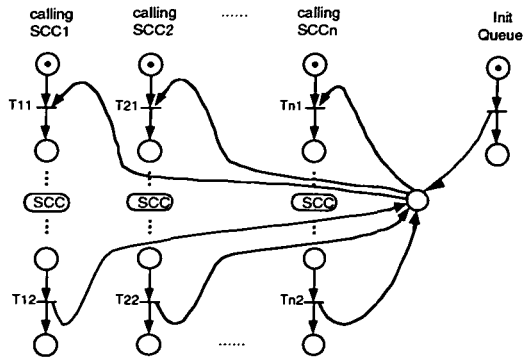

FIGURE 12. Waiting condition for calling SCCs.

$i = 1 \ldots n$) is added to the waiting queue. If the execution of an order terminates in the SCC which is needed for continuation (i.e. the token at transition $T_{j2}, j = 1 \ldots n$, $j \neq i$, moves to the right), the first order in the waiting queue can be executed.

Orders that are passed to an SCC for execution vary in complexity. In the SCC they are converted to a sequence of orders of reduced complexity, which are then passed directly to effectors or delegated to other SCCs. Depending on the degree of complexity, multi-host commands (MHC) and single-host commands (SHC) can be distinguished. For execution of MHCs, access to several hosts is needed. For the transmission of orders the communication service *rexe* is used. This is a service for the remote execution of commands based on the remote procedure call (RPC) mechanism. It consists of a client and a server component. The exit status of an executed command is returned to the calling instance. Additionally, information about the availability of the remote system is provided.

4.2.4. Integration with Comandos kernel

SOF and SCF can be used to manage the Comandos kernel implementation 'Guide' (Decouchant *et al.*, 1988). Managed objects are Guide 'persistent objects', Guide 'containers' and Guide 'sites'. Sensors have been implemented to get information about these objects. Possible control activities to be executed by SCF are given in Table 4.

The integration of the tools and concepts showed that in principle there is no obstacle to have a full integration of the management tools and the managed Guide system. The sensors have been implemented in Guide using the Guide language, which serves as an appropriate platform for applications in system management.

5. EXPERIENCES

An adaptive approach to distributed system management has been presented, which allows the observation and modification of a distributed information system according to changing requirements. The integration of functions for the identification of system states, decision support and manipulation of system resources allows to apply a consistent management model.

Several projects have been started using the findings of the Comandos system management approach. The concept of on-line risk management has been further developed in cooperation with a manufacturer in information technology. The work has been carried out in the framework of a security control centre development. Specific parts of the conceptual work and of the RiskMa implementation proved to be useful for the security control centre. The security control centre is dedicated to continuous security control of distributed systems, detection of critical situations and analysis of audit data. The risk management model used by RiskMa has been used for the development of dependability models used by the security control centre. Model parts dealing with dependability measures have been provided. Software sensors and aggregation algorithms used by

TABLE 4. Sensors and control activities for Guide managed objects

Managed object	*Information*	*Sensor*	*Control activity*
Persistent object	number of migrations, source and destination of migration	migration-sensor	fixing of object
Persistent object	average remain duration within one node	remain-sensor	fixing of object
Persistent object	local or remote invocations (net-traffic)	net-traffic-sensor	migration of objects
Persistent object	age and last access	ageing-sensor	removing of old objects or moving them to a special container
Persistent object	which user invokes object	Access-sensor	actions ensuring data security and data privacy
Persistent object	number of invocations	invoc-sensor	relocation of objects to other sites
Container	entire and free capacity	container-sensor	balancing of capacity of all containers and identifying requirements for new ones
Site	average processor-load	load-sensor	migration of jobs and activities and location of home-directories
Site	availability of site	avail-sensor	relocation of sensible objects to sites with high availability

RiskMa for system observation have been integrated in the security control centre.

The decision support of RiskMa has been used in a case study to examine the dependability of business processes in a manufacturing company. The computer-based store keeping of the manufacturer along with various configuration alternatives have been modelled using RiskMa. The simulation of the failure behaviour of business processes for store keeping and the underlying distributed system provided useful hints for the improvement of the system configuration.

In a project with a telecommunication company, the modelling and simulation capabilities of Disdes have been used for the determination of organizational and configurational requirements. The underlying model has been useful for the determination of required restructuring tasks and to validate different configuration alternatives.

The Comandos management approach has also been applied in a project with a machine manufacturing company. The architecture for system observation, system control and administration support was applied to specify a distributed system, able to incorporate existing infrastructure. For this purpose, new sensors and effectors to access MS-DOS PCs in a distributed Unix system have been integrated into the architecture.

6. FUTURE WORK

Each of the management tools and the supportive facilities have been implemented as prototypes. It is for further work to extend the functionality of the prototypes and to develop products based on these prototypes.

Using the model generated by Disdes two directions for further investigations are of interest. One is the link between the organizational model and application programming. The organizational model comprises information on the information resource required by the activities of a business process. This information can be useful in the application design. A feasibility study has been conducted with the DOCASE environment of Digital Equipment (Mühlhauser *et al.*, 1988). The second direction is towards integration of work flow support systems. Systems supporting business process work flow are available on the market (Rathgeb, 1993). A mapping of Disdes models onto work flow system configuration is currently being investigated.

Management tasks have been considered for distributed systems consisting of one-processor machines. Future distributed systems will also include parallel machines. New management tasks with respect to the computing elements of parallel machines emerge. The Comandos management model has to be extended and additional management tools are required. Aspects to be covered by management of parallel machines include load balancing between processing elements, optimization of input/output synchronization between processing elements, fault detection for processing elements and error handling.

ACKNOWLEDGEMENTS

The authors would like to thank Andreas J. Ness, Alexander Roos, Isabella Hofstetter, Wolfgang Clauss and Michael Rathgeb for their contributions to the project. This work was partly funded by the Commission of the European Communities through ESPRIT projects 834 and 2071 (Comandos).

REFERENCES

Balter, R. (1989) Construction and management of distributed office systems—achievements and future trends. In *ESPRIT '89, Proc. 6th Ann. ESPRIT Conf.*, Brussels.

Bracchi, G. and Pernici, B. (1985) The design requirements of office systems. *ACM Trans. Office Inform. Syst.*, **2**, 151–170.

Buhler, P. and Sturm, P. (1991) *COIN—An Object Model and Environment for Distributed Programming*. Report SFB124-26/91, University of Kaiserslautern.

Cahill, V., Balter, R., Harris, N. and Rousset de Pina, X. (eds) (1993) *The Comandos Distributed Application Platform*. Springer-Verlag, Berlin.

Clauss, W. (1991) *Konzeption, Spezifikation und Implementierung einer Systemmodifikationseinrichtung für vernetzte UNIX-Systeme*. Diplomarbeit 100/1493, University of Stuttgart.

Decouchant, D., Duda, A., Freyssinet, A., Paire, E., Riveill, M., Rousset de Pina, X. and Vandôme, G. (1988) Guide: an implementation of the COMANDOS object-oriented distribution system architecture on UNIX. In *EUUG Autumn Conf.*, pp. 181–193, Lisbon, Portugal.

Fichtner, J. and Persy, C. (1992) Concept of a security control center. In *SAFECOMP '92: Safety of Computer Control Systems*, Frey, H. H. (ed.), Pergamon Press, Oxford.

Fink, B., Baldus, H., Möller, M. and Kraemer, R. (1993) An integrated architecture for networked systems management. In *Proc. IEEE Int. Conf. on Communications*. Geneva.

Floyd, C. (1981) A process-oriented approach to software development. In *ICS '81, Systems Architecture, 6th ACM European Regional Conf.*, pp. 285–294, Westbury House.

Geihs, K. (1993) Infrastrukturen für heterogene verteilte Systeme. *Informatik-Spektrum* **16**, 11–23.

Grochla, E. (1982) *Grundlagen der organisatorischen Gestaltung*. Poeschel Verlag, Stuttgart.

Hoffner, Y. (1992) *Management in Object-Based Federated Distributed Systems*. ANSA Report APM/RC.389.01, Cambridge.

Horn, C. J., Ness, A. J., and Reim, F. (1988) Construction and management of distributed office systems. In *Information Technology for Organisational Systems*, Bullinger, H.-J., Protonotarios, E. N., Bouwhuis, D. and Reim, F. (eds), pp. 378–385, North-Holland, Amsterdam.

Kramer, J., Magree, J., Sloman, M. and Dulay, N. (1992) Configuring object-based distributed programs in REX. *IEEE Software Eng. J.*, Special Issue on Object-Oriented Systems, March 1992.

Lockemann, P. C. and Mayr, H. C. (1986) Information system design: techniques and software support. In *Information Processing '86*, Kugler, H.-J. (ed.), North-Holland, Amsterdam.

Meitner, H. (1990) Phasenkonzepte für das Risikomanagement. In *Zeitschrift für Kommunikations und EDV-Sicherheit KES6 6*, **6**, 394–399.

Mühlhauser, M., Schill, A. and Heuser, L. (1988) Software engineering for distributed applications: an object-oriented

approach. In *Proc. Int. Workshop Software Engineering and its Applications*, pp. 264–284, Toulouse.

Ness, A. J. (1990) *Eine Systemarchitektur für die Gestaltung und das Management verteilter Informationssysteme*. Springer-Verlag, Berlin.

Rathgeb, M. (1993) Work flow management auf der Basis verteilter Informationssysteme. In *Dokumenten-Management —Workflow Automation und Information Retrieval*, Bullinger, H.-J. (ed.), Springer-Verlag, Berlin.

Reim, F. and Meitner, H. (1991) A toolset for administration and management of distributed information systems. In *Human Aspects in Computing: Design and Use of Interactive Systems and Work with Terminals*, Bullinger, H.-J. (ed.), pp. 374–378, Elsevier Science Publishers, Amsterdam.

Reim, F. (1992) *Entwicklung eines Verfahrens zur rechnergestützten Gestaltung verteilter Informationssysteme*. Springer Verlag, Berlin.

Stotko, E. C. (1989) CIM-OSA. *Cim Management*, **5**, 9–15.

Weiss, W. R. and Baur A. (1990) Analysis of audit and protocol data using methods from artificial intelligence. In *13th National Computer Security Conf.*, pp. 109–114, Washington, DC.