Abstract: The office system ProMinanD supports cooperative office work by electronic circulation folders. Based on both, an abstract description of office tasks in terms of work steps, and knowledge of the organizational structure, a folder navigates through an organization from one office worker in charge to the next. Deviations from predefined cooperation procedures are supported by extensive exception handling. The system uses widely information kept in databases and being distributed over the network. Data structures and the mechanisms controlling the migration are discussed. The resolving of addresses and references is supported by an extended slot mechanism. Due to the distributed transaction concept the folder migration is secure and the distributed databases are consistent.

This paper is an outcome of the Project ProMinanD - Extended Office Process Migration with Interactive Panel Displays. ProMinanD is supported in part by ESPRIT - European Strategic Programme for Research and Development in Information Technologies.

1. Introduction

Productivity increase in the office in recent years lags behind when compared to the automatic production of goods in manufacturing industry. One reason is that office work is much more difficult to formalize because of the creative intelligence, experience, and personal judgement of office workers. Another area still resisting any streamlining is the handling of exceptional situations in offices. There are a lot of successful tools for particular types of work in the office such as text systems, form editors, calculation programs, mail, etc. Still missing, however, is a general integrating concept supporting typical cooperative activities in offices in a natural, problem-oriented and user-friendly way. It has been one of ProMinanD’s main goals to support office cooperation with flexible and understandable tools.

From analysis of large office organizations [1] and from other sources [2, 3, 4] the following facts became clear:

- Office tasks consist of steps being performed by several office workers in different office roles (i.e. having different organizational functions in the office).
- The office work is not only routine work. Beside strongly formalized office procedures there are many poorly formalized or non-formalized ones. Exceptions from predefined procedures are inevitable.
- The organizational structures, the office procedures themselves, and the functional contents of office roles are almost always changing.

In conventional paper offices these high expectations have been surprisingly well met by the conventional Circulation Folder (CF), a simple cover containing all documents of an office task; the address of the next office worker in charge is added by his predecessor. Its major advantage is the clear distinction between the carrier, the CF, and the intrinsic work documents contained in the CF. But the CF suffers also from some drawbacks: the way from
one office worker to the next may take a long
time, and it may be difficult to find a particular
CF in an organization. On the other hand, the CF
is an accepted part of the office world. ProMInanD replaces the CF by the Electronic
Circulation Folder (ECF).

2. ProMInanD’s Concepts

ProMInanD provides a user with an electronic
desk top consisting of an in-tray, form-box,
out-tray, etc. The desk top offers a number of
operations which have counterparts in the
conventional environment. The electronic
versions of most of the office tools and
operations, however, are more sophisticated in
order to offer functional improvements. The
main difference may be the addressing of the
office worker of the next step. With CF it is done
manually by the predecessor based on the
knowledge of office procedures and the
assignments of office roles to office workers. In
ProMInanD it is the matter of the system by
interpreting predefined office procedures.

2.1 Electronic Circulation Folder

An electronic circulation folder [5] represents an
office task (e.g. a car accident in an insurance
office) and consists of two main parts: its
description and its contents. The description is
divided into (1) the header consisting of the ECF
type, its system-wide unique identification and
its relationship to the other ECF’s, (2) the status
information reflecting the ECF state of progress,
and the history of all steps performed so far by
the office workers actually involved, and (3) the
migration specification (cf. 2.2). ECF descrip-
tions are complex objects by means of which
migration through the office organization is
specified as well as controlled. ECF descriptions
accompany their ECF’s.

The contents consist of (1) the work part
which keeps the documents necessary to perform
the office task, (2) the optional folder slip
which can be edited by the office worker, and (3)
the optional appendix which serves as a
container for additional documents entered at an
office worker’s discretion. It is the work part of
an ECF on which the intrinsic work on an office
task is carried out. The kind of work is
determined by the migration specification.

2.2 Migration Specification

The migration specification of an ECF is similar
to a flow-chart. It determines possible
migration routes of an ECF as sequences of
steps, decision functions, subtasks and
dependent ECF’s to be carried out. A step defines
which office worker has to perform some portion
of work, which documents are affected, and
which application programs are to be invoked.
Normally office workers are defined by their
office roles. A decision function is used to make
a choice between alternative routes in the mi-
gration specification using some data from the
ECF’s contents. For example, an approved busi-
ness trip is processed in a different way than a
refused one. A subtask is a part of the migration
specification with one way in and one way out.
On a higher level of abstraction it can be
considered as a (composite) step. A dependent
ECF is a normal ECF which has been started from
another active ECF. A dependent ECF runs in
parallel to its father and can join it again or
terminates as a single. For example, to read the
same paper in parallel, each of three referees is
given his own copy as a dependent ECF. The
copies are then joined together for final
judgement.

To contribute his part to the documents of the
ECF content the office worker is supported by
application programs, so called step programs.
Typical step programs are text or form editors,
but any other application program running under
the same operating system can be employed.

2.3 Electronic Organization Handbook

Migration specifications refer to conventional
organizational notions such as “has to report to”.
To that end, basic structural information about
the organization is kept in an electronic organi-
zational handbook, especially about:
- office workers employed,
- roles established in the organization,
- organizational units like posts, tasks and
groups of organizational units,
- relations between organizational units like
<unit1> is superior to <unit2>,
- organizational functions for posts and roles
such as "head of department<dept>" for post, or
"member of project<proj>" for role,
- relations between the organizational functions
  such as "reports to" or more sophisticated "role
 'member of committee<com>' is assigned to post
 'head of department<dept>"",
- actual assignments of office workers to roles
  and posts,
- actual assignments of posts or office workers
  to locations and workstations.

The transport of an ECF from one office
worker to the next one is the task of the
migration system. It interprets the migration
specification of an ECF and finds the office
worker in charge for the next step by looking up
the electronic organizational handbook.

3. Implementation

3.1 Global and Local Migration Servers

The implementation of the migration system
works on two levels. The Local Migration Server
(LMS) represents the desk top level, is present
on every user workstation and builds the user
interface between the office worker and
ProMinanD. The Global Migration Server (GMS)
represents the system level, is running on a
workstation of its own, and supervises the
system wide work flow. Transport of an ECF
from one desk top to the next one is performed
via the GMS under cooperation of both server
types. Both together, GMS and LMS, make out the
migration system.

Functional distribution of the migration
system in GMS and LMS is reflected by the design
of databases for ProMinanD's data: local
databases at each LMS containing data used for
local management, and a global database at the
GMS containing data used on the system level.

Kept in 50 relations and 30 views the GMS
database consists of information about active
workstations, office workers being logged in,
the office roles they are playing, active ECF's
and their locations, as well as the relations
between ECF's.

Kept in eleven relations and three views the
LMS database contains information about:
- the ECF's present in the workstation and their
  state,
- preferences of offices workers to step
  programs, and
- ECF's which have passed the desk top.

3.2 Database System

ProMinanD employs the relational database
system TransBase [6], originally developed at the
TU of Munich. TransBase is a "multiple" distribu-
ted system. It means that applications may
access more than one database in a session and
those databases may reside not only on a local
host but also on a remote host. TransBase is a
distributed system guaranteeing the consistency
of distributed data by implementing a distribu-
ted transaction concept. However, TransBase
is not a distributed system in the classical
sense, since it does not provide location
transparency.

TransBase databases are autonomous. Espe-
cially, each database has its own set of tables,
its own name space and its own files and
directories. They are concurrently accessible by
local and remote clients. Databases are named by
a logical name which is unique on the local host.
For a remote client it is identified by the host
name of the database server and the local name.

For ProMinanD especially the distributed
transaction concept is essential. It guarantees
that an action consisting of multiple updates on
different databases is performed either com-
pletely or not at all. In the migration system
there are many situations in which the consist-
tency of different databases is vital: When
transferring an ECF from one server to the other
it avoids loss or duplication of ECF's, when
maintaining the global migration management
information the GMS can be sure that the infor-
mation about the situation in the network is
correct.
4. Intelligent Usage of Databases by ECF's

The main problems with ECF's are that
- ECF's have to live with different system environments at different workstations (e.g. different path names of documents),
- the time between migration specification definition and interpretation is arbitrarily long, and
- organizations almost always change.

To solve these problems ProMinanD follows a twofold approach: on the one hand, keeping information by the GMS database as described above, on the other hand, performing evaluations as late as possible by means of an extended slot mechanism.

4.1. Extensions of Slots

A slot as described in [7] is an instance variable of an object which either has a value or knows where to get its value from when its value is needed. Values of slots are always computed at the time they are needed.

This mechanism is fine as long as the problems can be solved inside an object-oriented program. Then, at the time the value is needed, an a priori named message is sent to another object in order to get the requested value. An obvious first extension is that the used object as well as the parameters of the message sent to that object may be values of slots. For example: At the time of delivery what is known for an ECF and all of its documents is where they are stored at the current workstation. In order to get the path name of a certain document one may send that document the message "path".

This mechanism does not hold if, for example, the ECF needs to know what is the value of a certain field of a certain work part document. Then, an application-dependent decision program has to be started which looks up that document. To that end, a second extension of slots provide the start of independent programs, catching their results from "standard out".

The third extension is the possibility to run arbitrary sequences of select and join statements against different databases.

Slots in ProMinanD combine these three extensions. Their necessity is shown in the following two sections.

4.2. Start of Application Programs

It was an early design decision not to re-invent the wheel: ProMinanD should concentrate on cooperation support, but not rebuild office tools such as text and graphics editors. Thus, a mechanism was needed to run any application-dependent program without user intervention, and only based on information given by migration specifications and database information:
- The global database knows of the programs used by the ProMinanD system, of the assignment of program type and document type to concrete programs, and which program may run on which type of or on which individual machine.
- The migration specification of an ECF knows which type of program to use in which step, which documents and which additional parameters are necessary. Additional parameters are, e.g., views on forms describing which field is readable, writable and changeable. All parameters are specified by means of slots.

Thus, the call string for the invocation of a step program can be constructed as follows: Take the path name of the program which is apt for the required type of program and the required type of document, and append it with the additional parameters and the path of the document.

This mechanism makes sure that any new application-dependent program (step programs, decision programs, dependent ECF initialization and join programs) can be made known to the ProMinanD system only by entering appropriate information into the global database.

4.3. Resolution of Office Workers

Office workers are defined by one of the following ways:

For the initiator of an ECF nothing is defined by the migration specification, but the GMS database knows about which office worker may start which type of ECF in which role. At the time of instantiation of the ECF all information about the instantiator (his/her identification,
name and role) are filled into the migration specification.

Sometimes office workers having coordination functions are addressed, e.g., coordination for procurement of EDP equipment. These roles are unique inside a specific organization. Thus, a migration specification contains the identification of this role (by means of a slot having a certain value).

In some cases, names of office workers are contained in forms carried by an ECF in its work part. In these cases, the next office worker is resolved by looking up that form with an application-dependent decision program as described above.

The fourth alternative is that the next office worker is defined relative to another one who has performed a previous step. An example is the head of department of the initiator of an ECF. To that end, the organizational structure description inside the GMS database has to be looked up: (1) what is the organizational unit of the initiator of the ECF, (2) what is the superior organizational unit, (3) what is the manager post of the found organizational unit, and (4) what is the identification of the corresponding role to that post. The resolution procedure, again, is defined by slots. The late resolution of roles makes sure that all organizational changes are considered which have taken place after the instantiation of an ECF and before the delivery to the head of department.

5. Summary and Outlook

The system functionality and structure correspond to the state in the middle of 1990. Since then, several additions have been integrated which make the system more powerful:
- generalization of the two level system structure to N-level structure by connection of a GMS to a higher level GMS,
- support of additional network types,
- consistent roll back of office tasks by a compensation method [8],
- coexistence of different versions of a migration specification.
To these additions and a number of additional tools, 17 masters theses at the TU of Munich have substantially contributed.

ProMinanD, unlike other systems, relies extensively on global database information. Together with an extended slot mechanism it is, therefore, possible to resolve, e.g., addresses of office workers to whom ECF's are to be delivered just before delivery. In this way the description of the organization is set apart from the migration control algorithms. Specified and even instantiated ECF's are not affected by changes of the organization. Finally, the employment of a distributed database system guarantees that no ECF can get lost.

6. Acknowledgements

Many thanks to Prof. Rudolf Bayer and his students of the Technical University of Munich who have contributed with much work to the development and implementation of ProMinanD.

7. References

2. C.A. Ellis, M. Bernal: OfficeTalk-D - An Experimental Office System, First SIGOA Conference on Office Informations Systems, 1982