University of Wollongong

Research Online

Department of Computing Science Working Paper Series

Faculty of Engineering and Information Sciences

1986

A model of a distributed operating system

A. Goscinski
University of Wollongong

J. Indulska
University of Wollongong

Follow this and additional works at: https://ro.uow.edu.au/compsciwp

Recommended Citation

Goscinski, A. and Indulska, J., A model of a distributed operating system, Department of Computing Science, University of Wollongong, Working Paper 86-7, 1986, 17p.
https://ro.uow.edu.au/compsciwp/36

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

A Model of a Distributed Operating System

A. Goscinski, J. Indulska*

ABSTRACT

In this paper a logical model of a distributed operating system has been presented. This model of a distributed operating system contains a set of processes managing resources, connections between these processes, and mappings of events controlling this distributed operating system into processes managing resources. The fundamental types of resources introduced by the architecture of local computer networks, i.e., messages and data structures describing the location of resources in the network, have been defined. Operations on these resources and connections between the processes managing them and processes managing other resources of the distributed operating system have been presented. Addressing processes have been discussed.

The model has been constructed in such a way that a synthesis of different simulation tools (models) to study distributed operating systems can be carried out. In particular, this model makes it possible to construct simulation tools to study the effectiveness of distributed operating systems with processes managing resources defined in different ways.That means that the model has been developed in such a way to be both a concept and a tool like the model developed by A. K. Jones. The later was treated by us as a background model.

*/ J. Indulska is with the Institute of Computer Science, St. Staszic University of Mining and Metallurgy, Al. Mickiewicza 30, 30-059 Krakow, Poland

1. INTRODUCTION

An operating system should control resource allocation and provide the user with a virtual computer that serves as a convenient environment. In the case of local computer networks, the construction of an operating system fufilling the requirements of the definition given above implies design and implementation problems not known in the area of operating systems for centralized computer systems. The partial answer of this question could be found on the basis of an informal definition of a distributed operating system as being an extension of the definition given above. A distributed operating system should control network resource allocation to allow their use in the most effective way, provide the user with a convenient virtual computer that serves as a high-level programming environment and hide the distribution of the resources This means that in a given network node there is the possibility of a demand for access to a resource not known in that node. On the basis of these two definitions one can say that problems mentioned above are implied by the geographic distribution of resources, access to and management of resources, protection and reliability of the system as a whole and ways of distribution of the operating system between several nodes of the network.

There is a diversity of approaches to distributed operating systems design and implementation. The distributed operating systems are constructed as extensions of existing centralized operating systems into operating systems which allow remote access to network resources [Bro 82, Col 82, Deg 80, Lin 82] or as new operating systems [Cab 79, Des 80, Fri 83, Jen 84, Leb 85, Tan 84, Woo 82, Wat 80, Zim 81].

It should be pointed out that the latest do not solve in an optimal way all construction problems presented above and use many solutions for centralized operating systems. There are hypotheses that the distribution of the control of a computer system (e.g., resource management) will improve the effectiveness of the distributed operating system. This factor is used to point out the importance of research in distributed operating systems [Dav 81, Jen 81, Jen 84] to create a good base for design and implementation. These efforts are very closely connected with the development of adequate tools to describe and carry out research in distributed operating systems.

This paper presents a logical model of a distributed operating system constructed as a base model to carry on a synthesis of simulation models of distributed operating systems. The existing simulation tools making possible the comparison of centralized operating systems developed [Dav 79, Mad 80] have been constructed on the basis of well known logical models of centralized operating systems [Bri 73, Pet 85]. There is a lack of logical models to guide development of simulation tools for distributed operating systems.

The presented logical model of the distributed operating system is oriented towards problems of extending centralized operating systems as well as problems of searching for an effective structure of newly constructed distributed operating systems. When starting a project on the construction of distributed operating systems it was not clear what approach could be the most general and constructive. It has been decided to use ideas of the object model for centralized operating systems developed by A. K. Jones [Jon 78] because it is both a concept and a tool. In this object model, each resource - object of an operating system is connected with a set of admissible operations on that object and conditions of synchronization of these operations. Certain operating system resources have a direct physical implementation (e.g., I/O devices, processors, memory), other resources are logical ones (e.g., processes, files, semaphores) invoked by the operating system for effective management of physical resources. The set of admissible operations could be implemented by a process managing a resource [Lag 78, Wat 80].

The object approach used to develop a logical model of a distributed operating system simplifies a synthesis of simulation models - tools to study distributed operating systems. The construction of a simulation tool based on the logical model presented here for a given research problem statement is as follows: (i) choose from the logical model processes managing resources, and their connections and (ii) define operations for the chosen processes.

2. NEW RESOURCES OF A DISTRIBUTED OPERATING SYSTEM

Let us assume that the model of a distributed operating system is constructed for a heterogeneous local computer network with N nodes, where a heterogeneous network is a network which connects different computers, different peripherals, and there are different types of admissible operations on resources of an operating system. No restrictions have been imposed on a topology of the network. It has been assumed that the topology does not impose restrictions on several nodes, i.e., all nodes are equaly privileged and can carry on any functions of the operating system.

Based on definitions of the object model of a centralized operating system [Jon 78], the construction problems of a distributed operating system:
- access to remote resources,
- management of network resources,
- process synchronization,
- protection and reliability of an operating system

mentioned above can be stated as follows:
- new logical resources, which should be defined to develop an effective distributed operating system, are not known;
- an effective structure of the distributed operating system, i.e., connections between processes managing resources and distribution of processes in the network, are not known.

The problems given above are complicated by the fact that a set of admissible operations on a resource could be implemented as a set of connected concurrent processes located in different nodes of a network.

In the complex problem of defining an effective distributed operating system, it is possible to exhaust much more basic problem? Some new resources of the distributed operating system are known and a definition of managing processes for them makes it possible a development of a model of a distributed operating system which could be treated as a basis for further research.

Such a new type of a logical resource introduced by a computer network are messages used in communication and interprocess synchronization [Moo 82, Tan 85]. The communication could be carried out between operating system processes, between user processes and between an operating system process and a user process. Message passing requires managing additional physical resources, which are not known in centralized operating systems, i.e., communication interface, and creation (maybe) of additional logical resources to perform that message passing in an effective way.

Messages are sent in the network between two logically addressed units (e.g., processes, ports connected to processes). The reliability requirements and a need for a dynamically ballanced load of a network can imply that addresses of communicating processes (addresses of network nodes where these communicating processes run) are not constant. So, management of message passing requires system information about the present locations of message receivers. That need generates the second new type of logical resources of the distributed operating system. This type is data structures describing the location of resources in the network. These data have to describe the location of all logical and physical resources (processes managing resources) known by user processes and / or processes of the operating system. The following could be treated as example solutions of the problem of the distribution of data structures [Gos 84]: centralized (known in one network node) description of the location of resources in the network, distributed description according to classes of resources, local resources known in each network node, the location of all resources known in each node. Models of different methods of resource addressing will be presented in Section 4.

The addition of the new types of resources discussed above to the operating system requires:

(i) the definition of the logical representation of the resources, i.e., the definition of the data structures describing the location of all resources and the distribution of these data structures in the network,

(ii) the definition of operations on resources and synchronization of operations,

(iii) the implementation of these operations by managing processes which we call addressing processes, and

(iv) the definition of methods of attaching addressing processes into the the system of connection processes managing the message passing.

The method of the definition of these new resources has an influence on the effectiveness of the distributed operating system (measured in a sense of given performance indices such as reaction time on an event, service time of an event, etc.). Searching for the definition of the addressing processes is a part of the much more general problem of the construction of the effective distributed operating system (in particular a choice of suitable logical resources necessary to allocate physical resources among competing processes in such a way that it is possible to use them effectively.

3. A MODEL OF A DISTRIBUTED OPERATING SYSTEM

To construct the logical model of a distributed operating system it has been assumed that there is a set R of functionally connected processes managing resources. In the set R it is possible to distinguish different classes R_i, $i = 1, n$, containing processes managing one type of a resource such that

$$R = \cup_{i=1}^{n} R_i$$

where

$$R_i = \{R_{i1}, R_{i2}, ..., R_{im}\}, \ m = m_i \geq 1,$$

and m_i is a number of resources of class R_i.

It has been assumed that the distributed operating system is controlled by interrupts which inform about events which should be served. Let Z is a set of events of the distributed system and

$$Z = \cup_{i=1}^{r} Z_i$$

where Z_i is a class of events distinguished by the type of events.

The set of events of a computer system (hardware and an operating system) which control this system could be divided into two subsets:

a) events generated by the environment of the computer system, which present demands of that environment to the system;

b) events generated by hardware and processes of the operating system when serving events belonging to class a).

The above implies that for each operating system the set Z of possible demands of an environment is defined. These demands can be included in user processes and / or are generated by nondeterministic sources of events (e.g., operator, technological process controlled) characterized by stochastic parameters independent from the the operating system construction.

An architecture of a local computer network is characterized by a distribution of (i) computer facilities, (ii) processes managing resources, and (iii) environment demands. The addressing function defining the allocation of the processes, events and demands in the network (the connections with the nodes) in the time horizon T, $T \cup \mathbb{R}$, is a stochastic process

$$a_T : Q \times T \rightarrow N$$

where $Q = R \cup P \cup Z \cup X$, P is a set of user processes and X is a set of demands of an environment.

At each time t, $t \in T$, the addressing function

$$a : Q \rightarrow N$$

such that

$$a(q) = a_T(q, t), \quad q \in Q$$

defines the address of the node as a function of a process, an event, and an environment demand.

Let η is an equivalance relation of a node defined as follows:

$$Q \times Q \supset \eta \ , \ q_1 \, \eta \ q_2 \ \Leftrightarrow \ a(q_1) = a(q_2)$$

Then

$$Q / \eta = \{[q], \ q \in Q\}$$

defines a set of network nodes.

If N is a number of nodes in the network at time t:

$$N = a_t (Q)$$

then

$$X / \eta = \{X^i\}_{i = 1,N} , \quad X^i = \{x \in X: \ a(x) = i\}$$

$$Z / \eta = \{Z^i\}_{i = 1,N} , \quad Z^i = \{z \in Z: \ a(z) = i\}$$

$$R / \eta = \{R^i\}_{i = 1,N} , \quad R^i = \{R_{lk} \in R: \ a(R_{lk}) = i\}$$

$$P / \eta = \{P^i\}_{i = 1,N} , \quad P^i = \{p \in P: \ a(p) = i\}$$

The behaviour of the operating system serving demand $x \in X$ is defined by the function which attaches a serving process to an environment demand as follows

$$f : X \rightarrow R$$

In the centralized operating system the mapping presented above is carried out in such a way that each environment demand is linked with an event of the operating system what could be described by the two formulas

(i) $$k : X \rightarrow Z$$

and another one being a part of an interrupt system which links events and managing processes

(ii) $$k' : Z \rightarrow R$$

The choice of an event which should be served as first from a set of events arrived at the same time could be described by the formula

$$s : 2^Z \rightarrow Z$$

defined by a type of an interrupt system.

The description of the service of demand $x \in X$ could be presented by the superposition of two mappings k and k' as follows

$$f(x) = (k' \bullet k)(x), \qquad x \in X$$

The distributed operating system realizes a mapping

$$f : X \rightarrow R$$

where

$$X = \cup_{i=1}^{N} X^i, \qquad R = \cup_{i=1}^{N} R^i$$

and for the admissible distribution in the network of all environment demands and managing processes; that means that the following relations could be true

$$\exists\, x \in X : \ \sim (x \ \eta \ f(x))$$

So

$$\exists\, i, j : \ 1 \leq i, j \leq N, \ \ i \neq j :$$

$$x \in X^i \ \text{ and } \ f(x) \in R^j$$

The operation of several nodes of the network is controlled by the interrupts.

That means that $\forall i, 1 \le i \le N$, the following functions are defined

$$k^i : X^i \rightarrow Z^i$$

$$k'^i : Z^i \rightarrow R^i$$

$$s^i : 2^{Z^i} \rightarrow Z^i$$

The case, when $x \in X^i$, $f(x) \in R^j$, $i \neq j$, requires passing a demand to another node of the network, i.e., sending a message containing this demand to process f(x).

The demand of the message passing means that event $z \in Z^i$, $z = k^i(x)$, must be received by the process managing message exchange (message passing), which performs send operation. In the distributed operating system the connection between processes and network nodes could vary, i.e., the dynamic change of the process address is possible while the computer network works. That implies that each demand should be served as a start by the process managing a message passing*.

Even in that case when at any time of an event arrival the condition $x \, \eta \, f(x)$ is fulfilled.

So, if R_h describes a class of processes managing message passing, $1 \le h \le n$, and R_{hi} is the process managing message passing in node i ($R_{hi} \in R^i$) then

$$\forall x \in X^i : (k'^i \bullet k^i)(x) = R_{hi}$$

For the discussion to continue one can assume that the set of admissible operations of the process managing message passing contains:

(i) the operation of receiving the demand for the messsage passing,
(ii) the operation of message synthesis (message construction),
(iii) the operation of sending the message.

If for process R_{hi} a case described by relation $x \, \eta \, f(x)$ is taken into consideration, which requires the message passing to local process of the network node, then for the process managing message passing an additional operation has to be defined:

(iv) the operation of checking if process f(x) is local or remote.

---

*/ Each demand $x \in X$ is a demand of communication between processes. In the case $x \, \eta \, f(x)$ demand x could be treated also as a demand of communication implemented on the basis of message passing. It does not depend on the implementation of the interprocess communication in the node according to the duality of operating systems with message passing and operating systems based on the communication with shared variables [Lau 79].

The result of that operation performed determines the action of the process managing message passing:

- passing of a local demand to process $f(x)$ when $x \eta f(x)$, or
- message passing to another node when $\sim(x \eta f(x))$.

Let us consider the case when $\sim(x \eta f(x))$ as a new one when compared with centralized operating systems. For each event $z \in Z^i$, $z = k^i(x)$, accepted for a service in process R_{hi} managing message passing in the i-th network node, it is possible to define in synonymous way:
- the process which generates demands of an access to a resource,
- the process which manages that resource, and
- the demanded operation on that resource.

This implies that event z fulfiles the following relation χ

$$z \ \chi \ (q_1, op, q_2)$$

where $q_1, q_2 \in R \cup P$

$op \in O$, O - the set of admissible operations on the resources of the distributed operating system

and relation χ is defined in the following way

$$z \ \chi \ (q_1, op, q_2) \Leftrightarrow \exists \, x \in X^i, \ z = k^i(x), \ x \in X^i \cap q_2, \ f(x) = q_1, \ op \in f(x)$$

Message M constituting a new type of resource of the distributed operating system is a data structure sent between communicating processes. The message sent to process $f(x)$ must contain information on demand x.

The operation of message exchange between the processes managing message passing in the network is carried out according to a protocol of message passing. Independently of any specific protocol, the message contains the destination address, i.e., the address of process R_{hi} managing message passing in node j, $j \neq i$.

The method of the determination of the destination address of the receiver of the message containing the service demand in process $f(x)$ depends on:

(i) the chosen protocol of the message passing (the possibility of a determination of one receiver or many receivers of the message)

(ii) the modes of the distribution of the data structures describing the location of the resources in the network (of each new type of the resource introduced by the distribution of the computer system).

The message could be presented as follows

$$M = (q_1, \ op, \ q_2, \ j)$$

where $M \in (R \cup P \times O \times R \cup P \times N)$
and j is the address of the process of class R_h receiving the message.

Process R_{hi} performing the operation of message sending lets a determination of an address of a message receiver to the addressing process which manages in node i the data structures containing the location of the resources in the network.

The address is determined on the basis of the function

$$g^i : R \cup P \rightarrow N$$

which depends on the amount of information about the location of the resources in a given node, i.e., the definition of the addressing process. So, the service of event $z = k^i(x)$ in process R_{hi} implies

- the mapping of that event into new event z^* in the node $g^i(f(x))$ which sygnalize the arrival of the message

$$M = (q_1, op, q_2, g^i(f(x))), \quad \text{or}$$

- the passing of demand x to the managing process $f(x)$ if it is a local one

$$\varphi_{f(x)}^i : R^i \rightarrow Z \cup R^i$$

where

$$\varphi_{f(x)}^i (R_{li}) = \begin{cases} z^*, & z^* \in Z^j,\ j = g^i(f(x)),\ j \neq i \\ f(x), & \text{if } a(f(x)) = i \end{cases}$$

The family of functions $\{\varphi_{f(x)}^i\}$, $i = 1, N$, $f(x) \in R$ is defined in the synonymous way by functions g^i.

If a given event z^* does not fulfil relation $z^* \eta f(x)$, then the message containing the demand is sent by process R_{hi} according to addressing function g^j used in node j

$$M = (q_1, op, q_2, g^j(f(x)))$$

The method of the passing of the demands of access to a resource has been presented in Fig. 1.

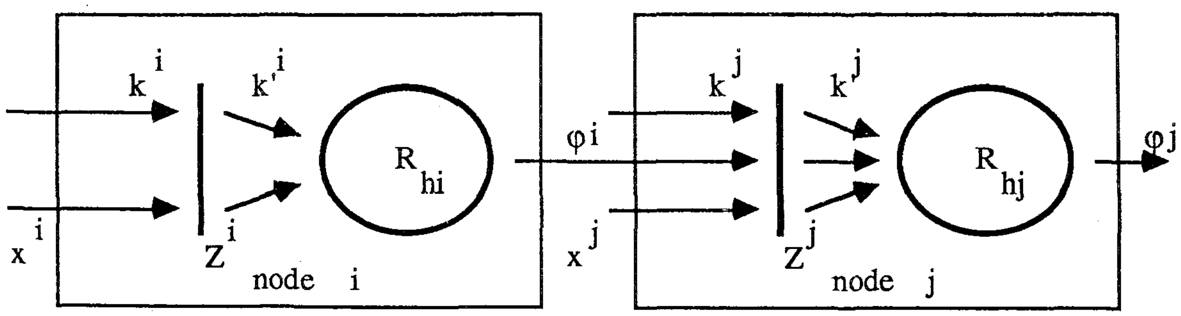

Fig. 1. The passing of the "access to resources" demand.

The sequence of nodes involved in the service of demand x, $x \in X^i$, defined in the way presented above, in the boundary case could be an one-element sequence (e.g., if the location of all resources is known in each node, then $g^i(f(x)) = a(f(x))$). In general, the realization of demand x, $x \in X^i$, has the following form

$$f(x) = (\psi^{sp} \bullet \dots \bullet \psi^{s1} \bullet k^i)(x)$$

where

$$\psi^{si} = \varphi_f{}^{si}(x) \bullet k'^{si}$$

and si is defined in a recursive way

$$s1 = i$$

$$s2 = g^i(f(x))$$

$$\vdots$$

$$s(j+1) = g^j(f(x))$$

and ψ^{sp} is the first element of a sequence $\psi^{s1}, \dots, \psi^{sp}$ that fulfiles the following condition

$$\psi^{sp}(z) = f(x).$$

Taking into account the remote access to the resources, the action of the distributed operating system, Σ, could be define by the structure

$$\Sigma = (Q,\ f,\ G,\ K',\ K,\ S)$$

where	$G = \{g^i\}_{i=1,N}$	is the family of addressing functions,
	$K' = \cup k'^i$	is the sequence of functions linking events and managing processes,
	$K = \cup k^i$	is the sequence of functions linking demands and events,
	$S = \{si\}_{i=1,N}$	is the family of functions defining passing demands of access to a resource.

4. ADDRESSING PROCESSES

In the distributed operating system the access to a remote resource requires a message to be passed to the process managing that resource. The message contains the address of the message receiver and that address is defined by the addressing process. The value of the address defined depends on the definition of addressing processes, i.e., on the distribution of data structures defining the location of resources.

In Section 3 presenting the logical model of the distributed operating system the choice of addressing processes is carried out on the basis of the family of functions

$$G = \{g^i\}, \qquad i = 1, N$$

where $g^i(q)$ defines the address of the receiver of the message sent from the i-th node and containing the request of the access to process q, $q \in R \cup P$.

It is possible to distinguish the following methods of resource addressing, i.e., the following functions g^i could be introduced [Gos 84]:

Hierarchical access (the access to the resource through intermediate addressing processes)

a) one intermediate addressing process - centralized addressing (Fig. 2a)

$$g^i(q) = \begin{cases} i, & \text{if } a(q) = i \\ a(R_{k1}), & \text{if } a(q) \neq i \text{ and } a(R_{k1}) \neq i \\ a(q), & \text{if } a(q) \neq i \text{ and } a(R_{k1}) = i \end{cases}$$

where R_{k1} is this addressing process which contains addresses of all the network resources

b) division into resource classes / n addressing processes - centralized addressing (Fig. 2b)

$$g^i(q) = \begin{cases} i, & \text{if } a(q) = i \\ a(R_{k1}), & \text{if } a(q) \neq i \text{ and } q \in R_j \text{ and } a(R_{k1}) \neq i \\ a(q), & \text{if } a(q) \neq i \text{ and } q \in R_j \text{ and } a(R_{k1}) = i \end{cases}$$

The location of all resources known in each node (Fig. 2.c)

$$g^i(q) = a(q), \qquad \forall q \in R \cup P$$

Distributed access to resources (Fig. 2d)

a) 1 : 1 mapping (message passing to one receiver)

$$g^i(q) = \begin{cases} i, & \text{if } a(q) = i \\ i \bmod N + 1, & \text{if } a(q) \neq i \end{cases}$$

b) 1 : N mapping (message passing to many receivers)

$$g^i : R \cup P \rightarrow \underbrace{N \times \ldots \times N}_{(N - 1 \text{ times})}$$

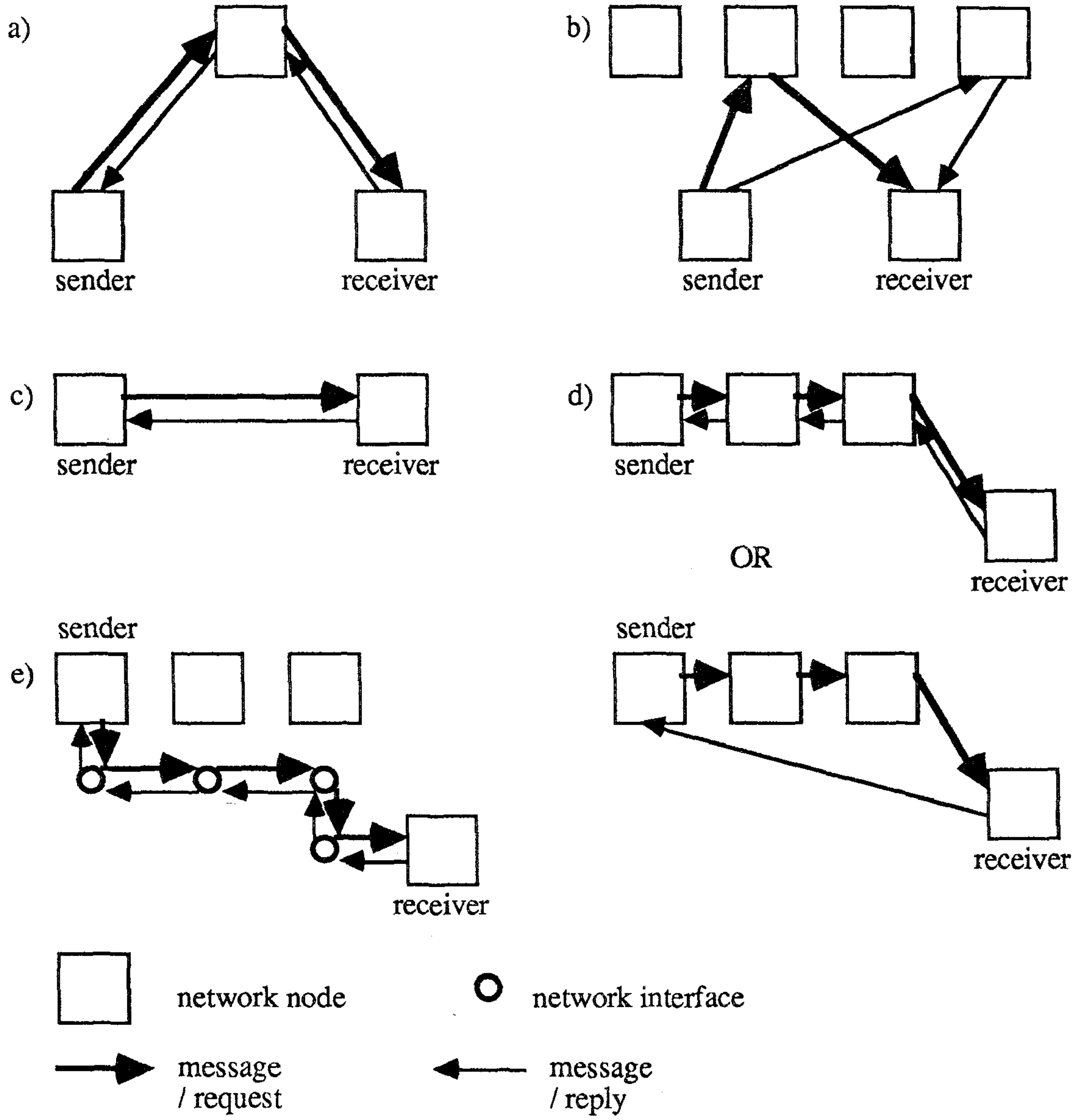

Fig. 2. The functional connections of processes communicating through message passing; a) centralized addressing, b) centralized addressing with division into resource classes, c) addresses known in each node, d) local resources known in each node, e) local resources known in network interfaces.

$$g^i(q) = \begin{cases} i \bmod N + 1,\ (i+1) \bmod N + 1, \ldots, (i+N-2) \bmod N + 1, & \text{if } a(q) \neq i \\ (i, 0, \ldots, 0), & \text{if } a(q) = i \end{cases}$$

Distributed access to resources when the network interface performs functions connected with the addressing of demands (Fig. 2e)

Let R_j, $j \neq 1, j \neq k$, be the class of processes managing message exchange in network interfaces and

$$g'^i : R \cup P \rightarrow N$$

be the addressing functions embodied in the processes of the class R_j, then the message in the i-th node is addressed by

$$g^i(q) = \begin{cases} i, & \text{if } a(q) = i \\ a(R_{ji}), & \text{if } a(q) \neq i \end{cases}$$

and in the network interface (in the R_{ji} -th process) respectively by

$$g'^i(q) = \begin{cases} i, & \text{if } a(q) = i \\ i \bmod N + 1 & \text{if } a(q) \neq i \end{cases}$$

5. THE MODEL EXTENSIONS

The service of demands for access to resources could be simplified [Ind 85] if there are some relations linking a demand and a managing process. If the demand and the managing process are in the same node, i.e., $x\ \eta'\ f(x)$, then it should be considered if the process managing message passing has to be involved in a service of that demand.

Let δ, $\delta \in R \cup N$, is a relation of connection with a node defined as follows

$$q\ \delta\ i \iff \forall t \in T,\ a'(q,t) = a(q) = i$$

and ε, $X \times X \supset \varepsilon$, is a relation of equivalance making possible to distinguish classes of demands addresses to several processes managing resources:

$$x_1\ \varepsilon\ x_2 \iff f(x_1) = f(x_2)$$

In that case, for demand $x \in X^i$, such that the following conditions are fulfilled

1. $x\ \eta\ f(x)$ and $f(x)\ \delta\ i$

2. $\forall x' \in X : x'\ \varepsilon\ x \Rightarrow x'\ \eta\ f(x)$

mapping $f(x)$ could be presented in the form

$$f(x) = (k'^i \bullet k^i)(x)$$

which is equivalent to the demand performed in the centralized operating system. That means that if the process does not change its location in the network and all demands are local, then communication could be performed locally, i.e., the process managing message passing is not involved in the communication.Moreover, the above form presents the influence of definitions of addressing processes on communication indices.

The operating system, except events signalizing environment demands, serves events generated when serving these first ones, e.g., the completion of the I/O transmission, clock interrupts. The demands fulfiling conditions 1. and 2. presented above and their service by adequate managing processes could be done without the process managing message passing.

In the network there is a possibility for the distribution of the service of demand $x \in X$ between several nodes. It concerns a case when new, additional logical resources* and new processes managing them are defined to have a comfortable and effective service of a resource. In that case the subset of managing processes is connected with the request of the environment.

If in set X there is the possibility of distinguishing different classes of request X_i,

$$X = \cup_{i=1}^{s} X_i$$

then $\forall x \in X_i$ the following set of processes managing request x is defined

$$\{f_1(x), \ldots, f_n(x)\} \in 2^R, \quad n = n_i$$

where $f_i : X \rightarrow R, \quad 1 \leq i \leq n_i$

and n_i is the number of processes managing resources involved in the service of requests of class X_i.

---

*/ The example of the construction of new logical resources, in the centralized operating system, to create a better virtual computer and to achieve an effective access to resources is a file system. The multilayer structure of the service of I/O demands (the recognition of the organization and identification of a file, the creation of an independance of data and facilities, the connection of logical channels and physical ones) could be mapped into managing processes. In the case of a similar construction of a file system in the network, there is the possibility of the distribution between several network nodes of managing processes performing requests of I/O [Alb 79, Kie 79]. In general, the problem of the distribution in the network of the service of requests if not well recognized [Dav 81].

The distribution of managing processes $f_i(x)$, $i = 1, n_i$, in the network can increase the effectiveness of a real-time distributed operating system (reaction time, time of the service of requests). The method of allocating a service request among managing processes $f_i(x)$ and the method of their distribution (location) in the network is an open problem requiring additional research.

6. CONCLUSION

In the paper the logical model of the distributed operating system has been presented. This model introduces the fundamental new types of resources implied by properties of local computer networks.These new types of resources are messages and data structures defining the location of resources in the network. In the model the operations on these resources and the connections of processes managing the new types of resources with other processes managing the resources of the distributed operating system have been defined.

Thanks to a good structuring based on the object model of the centralized operating system which makes possible a choice of any level of abstraction to define a set of admissible operations on resources, the model could be treated as a basis for construction of simulation tools to carry out the performance study of distributed operating systems with different processes managing resources. The simulation tools could be used to search the effective service modes in processes managing new types of resources as well as resources known from centralized operating systems. The method of the effective service of the last, because of the distribution of resources in the network, is not known also.

The logical model of the distributed operating system presented in this paper has been used to construct some simulation tools. The first construction was a tool to study the effectiveness of different addressing processes [Ind 85, Ind 86]. In the implementation of that model the structuring of the distributed operating system used in the logical model has been kept out. That made possible the relatively simple modification of that model to carry out simulation studies for different goals, e.g., the effectivenes of process synchronization in the distributed operating system. The simplicity of all modifications has been implied by the possibility of utilization of many modules implementing processes managing resources in the newly constructed tools. It concerns these processes which have the same operations on resources and the same levels of abstraction when defining these operations. As a result the cost of the construction of each next simulation tool is decreased and that feature confirms the applicability and utility of the logical model of the distributed operating system developed.

ACKNOWLEDGEMENT

We wish to thank G. S. Stafford for his helpful comments.

REFERENCES

[Alb 79] Albrecht H. R., Thomasos L. C., I/O Facilities of the Distributed processing programming executive (DPPX), IBM Syst. Journal, Vol.18, No. 4, pp. 526 - 546, 1979.

[Bri 73] Brinch Hansen P., Operating Systems Principles, Prentice - Hall, Englewood

Cliffs, New Jersey, 1973.

[Bro 82] Brownbridge D. R., Harshal L. F., Randell B., The Newcastle Connection of UNIXes of the World Unite!, Software - Practice and Experience, Vol.12, pp. 1147 - 1162, 1982.

[Cab 79] Cabanel J. P., Marouane M. N., Besbes R., Sasbon R. D., Diarpa A.K., A Decentralized OS Model for Aramis Distributed computer System, IEEE 1979.

[Col 82] Collinson R. P. A., The Cambridge Ring and UNIX, Software - Practice and Experience, 1982, Vol. 12, pp. 583 - 594, 1982.

[Dav 79] David Ch., Madaule F., Mendelbaum E., Building a Simulation Model for Real -Time Systems, SOCOCO' 79, Prague 1979.

[Dav 81] Davis D. W., Holler E., Jensen E. D., Kimbleton S. R., Le Lann G., Turber K. J., Watson R. W., Distributed Systems - Architecture and Implementation - An Advanced Course, Springer - Verlag, Berlin, Heidelberg, New York, 1981.

[Deg 80] Degenhardt K. H., Wiesnerg., Woletz W., Distributed Control and Data Processing with Modified Real - Time Operating System, Real - Time Data Handling and Process Control, North -Holland Publishing Company, Brussel and Luxemburg, 1980.

[Des 80] Deschizeaux P., Ladet P., Real - Time Structuration Language for Decentralized Process Control, Real - Time Data Handling and Process Control, North -Holland Publishing Company, Brussel and Luxemburg, 1980.

[Fri 83] Friedrich G. R., Eser F. W., Management Units and Interprocess Communication in DINOS, Simens Forsch. - und Entwickl., -Ber., Bd. 12, No. 1, pp. 21 - 27, 1983.

[Gos 84] Goscinski A., Indulska J., An Object Approach to Network Operating System Model Construction, IEEE Technical Committee on Distributed Processing Newsletter, Distributed Operating Systems, Vol. 6, No. SI - 2, 1984.

[Ghe 85] Ghertal F. F., Mamrak S., An Optimistic Concurrancy Control Mechanism for an Object Based Distributed System, Proc. of the 5-th International Conference on Distributed Computing Systems, Boulder, Colorado, 1985.

[Ind 85] Indulska J., Studies of Real - Time Distributed Operating Systems, Ph.D. Thesis, The St. Staszic University of Mining and Metallurgy, Krakow, 1985.

[Ind 86] Indulska J., Goscinski A., A Simulation Tool and a Study of Methods of Addressing of Resources of a Distributed Operating System, in preparation,1986.

[Jen 81] Jensen D. E., Distributed Control, Distributed Systems - Architecture and Implementation - An Advanced Course, Springer - Vrelag, Berlin, Heidelberg, New York, 1981.

[Jen 84] Jensen D. E., ArchOS: A Physically Dispersed Operating System, IEEE Technical Committee on Distributed Processing Newsletter, Distributed Operating Systems, Vol. 6, No. SI - 2, 1984.

[Jon 78] Jones A. K., The Object Model: A Conceptual Tool for Structuring Software, Operating Systems: An Advanced Cource, Springer Verlage, pp. 7 - 16, 1978.

[Kie 79] Kiely S. C., An Operating System for Distributed Processing - DPPX, IBM System Journal, 1979.

[Lag 78] Lagally K., Synchronization in a Layerd Systems, Operating Systems: An Advanced Cource, Springer Verlage, pp. 252 - 278, 1978.

[Lau 79] Lauer H. E., Needem R. M., On the Duality of Operating Systems Structures, Operating Systems Reviev, Vol. 13, No. 2, 3 - 19, 1979.

[Leb 85] LeBlanc T. J., Friedberg S. A., Hierarchical Process Composition in Distributed Operating Systems, Proc. of the 5-th International Conference on Distributed Computing Systems, Boulder, Colorado, 1985.

[Lin 82] Lin M. T., Tsay D. P., Lian R. C., Design of a Network Operating System for the Distributed Double - Loop Computer Network (DDLCN), Local Computer Networks, Ravasio P. C., Hopkins G., Naffah N. (Editors), North - Holland Co., IFIP, 1982.

[Mad 80] Madey J., Oprogramowanie wspomagajace. Modelowanie systemow liczacych, Projekt OS Kit, PWN 1980.

[Moo 82] Moore L. C., Bukys L., Heliotis J. E., Design and Implementation of a Local Network Message Passing Protocol, 7-th Conference on Local Computer Networks, Minneapolis, Minnesota, 1982.

[Pet 85] Peterson J. L., Silberschatz A., Operating Systems Concepts, Addison - Wesley Publishing Co., 1985.

[Sal 84] Saltzer J. H., Reed D. P., Clark D. D., End - To - End Arguments in System Design, ACM Transactions on Computer Systems, Vol. 2, No. 4, 1984.

[Sha 74] Shaw A. C., The Logical Design of Operating Systems, Prentice - Hall, Englewood Cliffs, New Jersey, 1974.

[Tan 84] Tanenbaum A. S., Mullender S. J., The Design of a Capability - Based Distributed Operating System, Rapport nr. IR-88, Vrije Universiteit Amsterdam, 1984.

[Tan 85] Tanenbaum A. S., van Renesse R., A Survey of Current Research on Distributed Operating Systems, Proc. of the Eighth Australian Computer Science Conference, Melbourne, 1985.

[Wat 80] Watson R. W., Network Architecture Design for Back - End Storage Networks, Computer, pp. 32 - 48, February 1980.

[Woo 82] Wood B. J., Thompson D. R., Rogers L. D., Bryant D. M., A local - Area Network Architecture Based on Message Passing Operating System Concepts, 7-th Conference on Local Computer Networks, Minneapolis, Minnesota, 1982.

[Zim 81] Zimmerman H., Banino J. S., Caristan A., Guillenment M., Basic Concepts for the Support of Distributed Systems, The CHORUS Approach, Proc. of the Second International Conference on Distributed Computing Systems, France, April 1981.