

# Performance Engineering of Human and Computerised Workflows

Gunnar Brataas\*, Peter H. Hughes\*\* and Arne Sølvsberg\*\*\*

Department of Computer and Information Science  
Norwegian University of Science and Technology (NTNU)  
7034 Trondheim, NORWAY

**Abstract.** An integrated approach to performance analysis of workflows in terms of throughput and response time to satisfy customer requirements is investigated. Integration of performance is particularly important for workflow systems where human and computerised processes are intertwined. The proposed modelling framework encompasses both human and computer resources and is based on the Structure and Performance specification method (SP). The framework is explored and illustrated with a case study for a workflow system for handling sales contracts in the natural gas industry. The case study involves performance modelling of both manual and computerised systems, the latter using a Lotus Notes platform.

## 1 Introduction

Organisations increasingly depend on computerised workflow systems which assist case workers in delivering services to customers. An integrated approach to performance analysis of workflows in terms of throughput and response time to satisfy customer requirements is needed [VLP95]. The total information system, consisting of the workflow system and the case workers, must have acceptable *performance* in terms of response time, throughput, and utilisation of the available resources. *Performance engineering* of information systems aims at developing systems with acceptable performance [Smi90]. Performance engineering has until now almost exclusively dealt with the performance of computerised information systems. This paper addresses performance engineering of the total information system consisting of both the computerised workflow system and the case workers who use it. For the organisation, this paper relies on a mechanistic view [Mor86]. A comprehensive theory of organisational performance must also take other views into account, e.g. the social view.

This paper builds on a thesis describing a method for performance engineering of workflow systems [Bra96]. The basic phases of the method are:

---

\* Present affiliation: Telenor R & D, Kongens gate 8, 7005 Trondheim, Norway.  
Email: Gunnar.Brataas@fou.telenor.no

\*\* Also at: Modicum Ltd, Congleton, Cheshire CW12 3HZ, United Kingdom.  
Email: phh@modicum.demon.co.uk

\*\*\* Email: Arne.Solvberg@idt.ntnu.no

- I. Specify system requirements:
  - IA. Determine objectives of performance model.
  - IB. Specify system boundaries.
  - IC. Estimate workload.
  - ID. Determine performance requirements.
- II. Create performance model:
  - IIA. Establish static model.
  - IIB. If necessary, establish dynamic model.
- III. Guide system development.
- IV. Verify and refine overall performance model.

This method was illustrated with two case studies in the thesis. In this paper we use a condensed version of *The Gas Sales Telex Administration Case Study* (GASCASE). The paper is structured as follows: A conceptual framework which governs the approach, the *basic framework* is introduced in Section 2. The underlying modelling paradigm *Structure and Performance* is described in Section 3 and illustrated with respect to manual telex handling. GASCASE is introduced in Section 4, which also completes the description of the original manual solution in GASCASE. The computerised solution in GASCASE is considered in Section 5. Section 6 discusses the practicability of the framework in the context of GASCASE and Section 7 offers some conclusions.

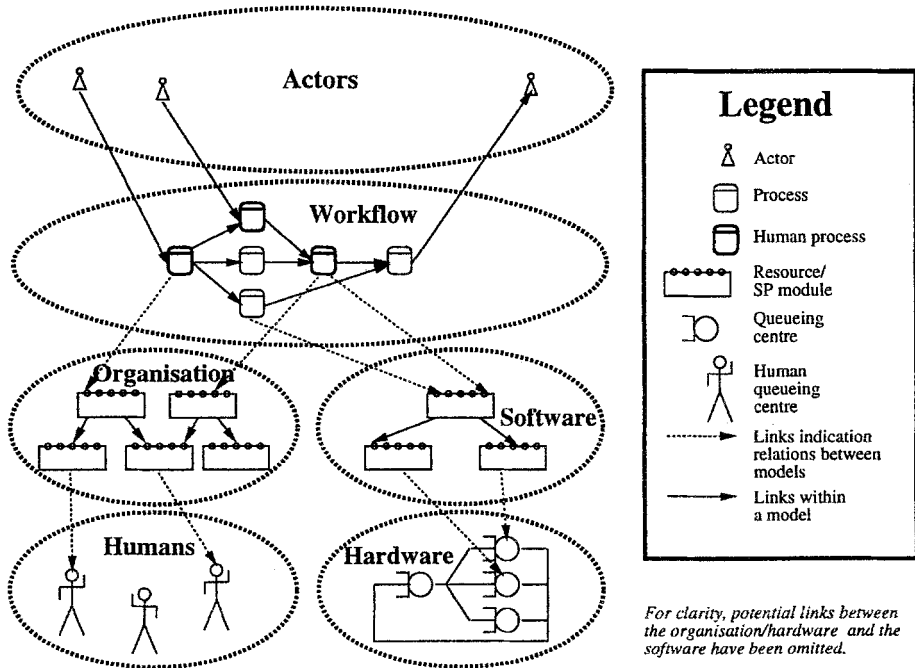


Fig. 1. Basic framework for performance engineering in workflow organisations.

## 2 Basic Framework

Workflows is often centred around (electronic) documents which flow between the environment and the organisation and within the organisation. Probably the most basic difference between workflows is the degree of predictability: (1) *routine workflows* have deterministic resource demands, (2) *template workflows* have statistical resource demands, e.g. some subprocesses may be optional, and (3) *ad-hoc workflows* have unpredictable resource demands, often because the next subprocess is not determined before the current subprocess is finished.

As illustrated in the basic framework in Fig. 1, *workflow processes* use *organisational resources* like departments and sectors which again use *humans*. Workflow processes also use *software resources* like databases or file systems which finally use *hardware resources* like disks, networks and CPUs. *Actors* start workflows and use the results of workflow. Note that the term resource is used more narrowly than the ordinary use of this term. In this paper, a resource is reactive. A resource carries out work which is initiated by an actor. In the figure, the solid lines inside of each ellipse describe relationships between the objects in each ellipse, whereas the dotted lines describe relationships between objects in different ellipses. The six ellipses in Fig. 1 together constitute a *workflow system*.

Each resource offers *operations* to resources at higher levels of abstraction and uses operations from resources at lower levels of abstraction. Resources can either be *existing* or *projected*. In a performance model, the properties of a resource are described via parameters. For existing resources, measurements may be used to provide these parameters. For projected resources, estimation is necessary.

## 3 Structure and Performance (SP)

SP is a method for describing the hierarchical and modular nature of a system and for analysing the resource requirements of its component parts [Hug88]. In SP a resource which provides a related set of services or *operations* is known as a *component*. More formally, a component implements a data type, i.e. a data structure defined by the operations which can be performed upon it.

Until now, SP has almost exclusively been used to model *computerised* information systems. To illustrate SP, we introduce a model of a *manual* information system, the secretary in Fig. 2. The secretary is seen to possess operations for communication, processing and memorising. For example, the secretary acts as a communicator when receiving and sending documents, and is using the mail, `secretary_copy` and `secretary_bus` resources, where the two latter resources are resources internal to the secretary, e.g. it takes secretary time to copy a document. `Secretary_bus` is related to the walking between, e.g. the archive and the desk.

Each link between two components are represented by a *work complexity specification matrix*. An example is shown below:

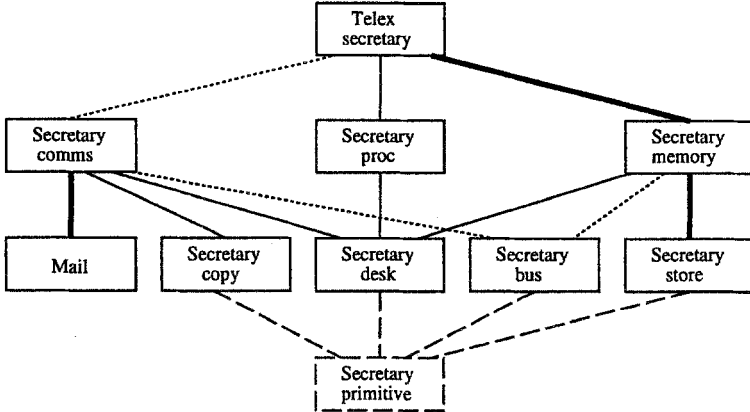


Fig. 2. SP model for the secretary.

$$C_{Secretary\_proc}^{Telex\_secretary} = \begin{matrix} in\_telex \\ out\_telex \end{matrix} \begin{matrix} retype & polish \\ \left[ \begin{array}{cc} 0 & 0 \\ n_p & xn_p \end{array} \right] \end{matrix}$$

In this example, the component `telex_secretary` devolves work on the component `secretary_proc`. This complexity specification corresponds to the link between these two components in Fig. 2. For outgoing telexes (with the operation `out_telex`), the resource `telex_secretary` has to retype  $n_p$  pages written by the contract specialist. There are also  $x$  iterations between the contract specialist and the secretary with draft versions of outgoing telexes, each with  $n_p$  pages which the `telex_secretary` has to polish.

The work devolved by `telex_secretary` to a component at a lower level, e.g. `secretary_desk` is calculated by adding the work for all possible paths between the two components. The work along each path is calculated by multiplying the complexity matrices for each link along the path. Thus, the total resource demand on `secretary_desk` is:

$$C_{Secretary\_desk}^{Telex\_secretary} = C_{Secretary\_comms}^{Telex\_secretary} \cdot C_{Secretary\_desk}^{Secretary\_comms} + C_{Secretary\_memory}^{Telex\_secretary} \cdot C_{Secretary\_desk}^{Secretary\_memory} + C_{Secretary\_proc}^{Telex\_secretary} \cdot C_{Secretary\_desk}^{Secretary\_proc}$$

Using the parameters in Table 1, the result becomes [Bra96]:

$$C_{Secretary\_desk}^{Telex\_Secretary} = \begin{matrix} in\_telex \\ out\_telex \end{matrix} \begin{matrix} split & stamp & read & type \\ \left[ \begin{array}{cccc} 1 & 1 & 0.1(r_i + 3) & 0 \\ 1 & 1 & (1 + x + n_i)n_p + 0.1(4 + r_i + r_e) & n_p(p_s + xp_i) \end{array} \right] \end{matrix}$$

Operations in SP are typed. The basic types of operations in SP are processing operations, memory operations and communication operations. Typing of operations in SP is indicated by the thickness of the links in SP models as shown in Fig. 2. Processing operations have solid lines, memory operations have bold lines and communication operations have dotted lines. A communications component is required whenever a higher level operation is implemented by two or more independent processing and/or memory operations.

The distinction between work and load is important in SP. *Work* is defined as a vector  $w$  of operations applied on a given component or resource in SP. For work, there is no information about time: how long time an operation takes or the sequence of operations. *Load* may be modelled as a transaction rate  $\lambda$ , for an open system, or as a concurrency  $N$ , for a closed system, where  $N$  represents the number of continually active processes, e.g. user terminals. Taken together, work and load define the workload for some system. Section 4 describes the workload of GASCASE. When a load model and a work model are combined with a contention model, we have a performance model. The dynamic load model is only relevant in the contention model and not in the static work model.

The mapping between the devolved work on a component and the time required to carry out the operations is specified by a set of resource demand vectors. As an example of resource demands, the resource `secretary_desk` offers four operations as described below, e.g. to split a telex takes 0.25 minutes, and typing takes 10 minutes per page:

$$D_{\text{Secretary\_desk}} = \begin{matrix} \textit{split} \\ \textit{stamp} \\ \textit{read} \\ \textit{type} \end{matrix} \begin{bmatrix} 0.25 \\ 1 \\ 1 \\ 10 \end{bmatrix}$$

When a complexity specification is combined with a resource demand vector, the total resource demands are calculated, e.g. for the secretary as desk:

$$D_{\text{Secretary\_desk}}^{\text{Telex\_secretary}} = C_{\text{Secretary\_desk}}^{\text{Telex\_secretary}} \cdot D_{\text{Secretary\_desk}}$$

Using the earlier expressions for  $C_{\text{Secretary\_desk}}^{\text{Telex\_secretary}}$  and  $D_{\text{Secretary\_desk}}$  and substituting the parameters with the numbers described in Table 1, we get:

$$D_{\text{Secretary\_desk}}^{\text{Telex\_secretary}} = \begin{matrix} \textit{in\_telex} \\ \textit{out\_telex} \end{matrix} \begin{bmatrix} 1.75 \\ 25.75 \end{bmatrix}$$

Thus, the secretary *as processor* will spend 1.75 minutes on incoming telexes and 25.75 minutes on outgoing telexes, when contention is not considered. For a contention model, the resource demand is required for each independent physical resource, in this case `secretary_primitive`, see Section 4.

Parameter	Value	Description of parameter
$n_t$	2	Number of telexes to be read during receipt handling
$n_p$	2	Number of pages in a telex.
$d_p$	2	Walking distance, in meters, to the mail box.
$r_e$	9	External receivers of a telex.
$r_i$	2	Internal receivers of a telex.
$x$	1	Number of iterations between contract specialist and telex secretary
$p_i$	0.5	Fraction of pages which has to be retyped initially by the telex secretary based on text from the contract specialist.
$p_s$	0.25	Fraction of pages to be retyped during each iteration between between contract specialist and telex secretary

Table 1. Some parameters for the manual solution in GASCASE.

## 4 GASCASE: Manual Subsystem

GASCASE (The Gas Sales Telex Administration Case Study) was part of a project in the Norwegian petroleum company Statoil. In the natural gas industry, telexes are used as a means of communication between stakeholders of the gas production and trade. The central case worker in the Statoil project was the telex secretary, and the focus of the project was improvement of telex handling effectiveness in the sales administration sector because of an anticipated increase in the number of telexes. Based on the objectives of the Statoil project, the performance objective for the performance model was formulated as: *Find out if response time was smaller for the computerised solution than for the manual solution*, expressed mathematically as:

$$\begin{aligned}
 & W_{\text{Today, Normal}}^{\text{T Sales\_Telex\_administration}} \cdot R_{\text{Computerised, Today, Normal}}^{\text{Sales\_Telex\_Administration}} \\
 < W_{\text{Today, Normal}}^{\text{T Sales\_Telex\_administration}} \cdot R_{\text{Manual, Today, Normal}}^{\text{Sales\_Telex\_Administration}}
 \end{aligned}$$

Both for the computerised and for the manual solution, the present normal workload is used, where “normal” refers to a particular workload scenario defined below. The *transposed* workload vector  $W_{\text{Today, Normal}}^{\text{T Sales\_Telex\_administration}}$  is described later.  $R$  is a vector of response times for each operation on the Sales Telex Administration. In addition, the projected system must cope with the anticipated increase in future workload, which leads to a stricter performance requirement.

Figure 3 shows the main subsystems in GASCASE. Lotus Notes together with associated software and hardware plays the role of a simple workflow system. The computerised subsystem is described in more detail in Section 5. The manual information subsystem is described further below. Figure 3 represents two applications of the generic framework in Fig. 1.

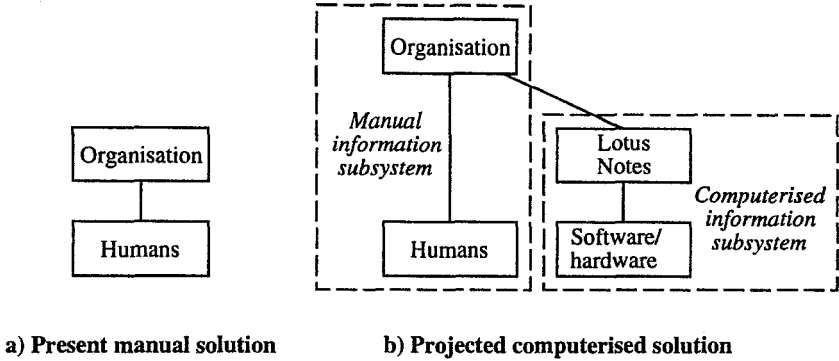


Fig. 3. Two organisational information systems.

The organisation chart for the Sales Administration sector in Statoil is shown in Fig. 4. Note that this organisational chart is an SP model, but some links are missing, e.g. memory links which could represent document centres and archives in Statoil. The postman is used for communication within Statoil, which is shown by the dotted line in Fig. 4. The focus of the paper will be telex handling, which is performed by the resources *telex\_secretary* and *contract\_specialist* in Fig. 4, marked with the **boldface** font. The other secretaries in Contract coordination and Troll were not engaged in telex handling.

For telex handling, two workload scenarios were defined: *normal operation* and *shortfall*, which occur when there is a production stop in the North Sea. Two operations were relevant in this case study, namely (1) *in.telex*, number of incoming telexes to *Sales.telex.administration* and (2) *out.telex*, number of outgoing telexes from *Sales.telex.administration*. When the Statoil project started, normal workload per day for the gas sales telex administration was:

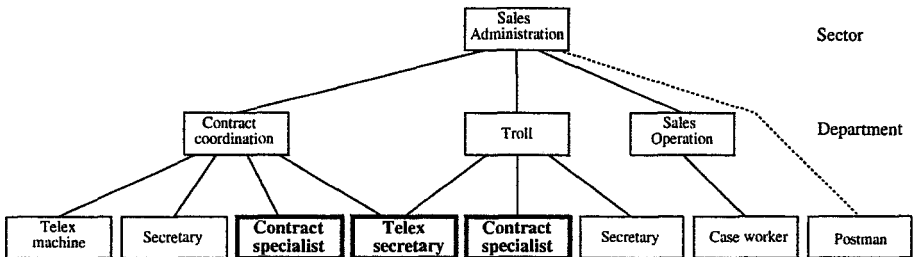


Fig. 4. Rudimentary SP diagram for the sales administration sector.

$$W_{\text{Today, Normal}}^{\text{Sales\_Telex\_Administration}} = \begin{bmatrix} \text{in\_telex} & \text{out\_telex} \\ 20 & 10 \end{bmatrix}$$

Expected future, normal workload per day was 75 incoming telexes and 50 outgoing telexes per day. The workload during shortfall was estimated to be twice the workflow during normal operation.

The workflow for outgoing telexes is shown in Fig. 5 [Bra96]. Based on this workflow, the SP model in Fig. 2 is derived. The resource `telex_secretary` is visible to the outside world. Mail is also used by several other resources. All the other resources in Fig. 2 are visible only to the `telex_secretary`. Internally, the secretary plays *roles* in communication, processing and memory, indicated by placing `secretary_comms`, `secretary_proc` and `secretary_memory` under `telex_secretary` in the figure. On a low level, `secretary_copy`, `secretary_desk`, `secretary_bus`, and `secretary_store` perform the basic operations of the telex secretary. In Fig. 2, these low-level components are implemented by the same person, and are therefore not four independent subsystems. Therefore, a *primitive* subsystem termed `secretary_primitive` is introduced. The four low-level components all delegate work to this primitive subsystem.

In Section 3 the resource demand for the component `secretary_desk` was derived. When the resource demands for the other components: `secretary_copy`, `secretary_bus`, and `secretary_store` are added together, *the total resource demands for the telex secretary under present, normal workload, are estimated to be 3.6 minutes for handling an incoming telex, and 30 minutes for an outgoing telex.* This was confirmed by previously estimated resource demands in an internal Statoil report. In addition, a small amount of time (0.25 minutes) is used by the contract specialists to deliver outgoing telexes.

With the workload  $W_{\text{Today, Normal}}^{\text{Sales\_Telex\_Administration}}$ , the secretary will spend on average  $20 \cdot 3.6 + 10 \cdot 30 = 372$  minutes  $\approx 6.2$  hours each day on telex handling, leading to an average utilisation of 83 % ( $\frac{6.2}{7.5} = 0.83$ ). This corresponds roughly to a busy full-time position (7.5 hours each day), especially if we also consider the context changes which are necessary for each telex, and which are not considered explicitly. An utilisation of 83 % is very high for computers, but is possible for humans because they are more flexible. If the workload is too high, humans

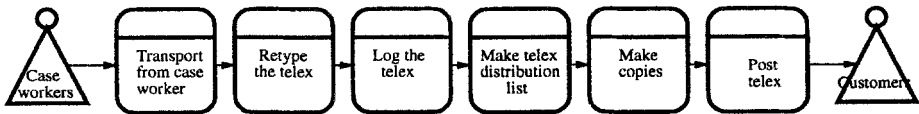


Fig. 5. Workflow for outgoing telexes.



may perform shortcuts to get the work done. During shortfall and future load, estimated resource demands are too much for one person if she is to work with sufficient quality.

The overall conclusion is that the static model produced estimates which were consistent with observed behaviour, in terms of utilisation. The correspondence between the model and what was expected was good enough to be used as a basis for *changes* in the actual system. The parameters of the model could have been refined if the model were to be used for response time estimation, which is more sensitive to parameter quality.

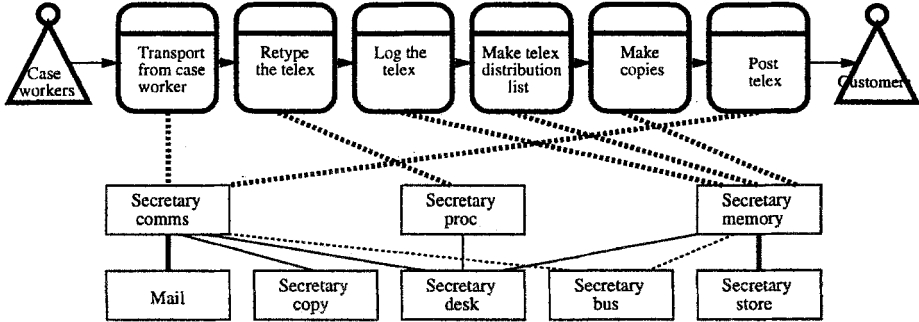
Modelling of contention was not necessary in this case study, since the manual system was not going to be used when the computerised solution in Section 5 came in production. If contention modelling were necessary, the static model could give us the parameters we need. As a simple illustration, an open, two-class queueing network with these parameter values would predict that the secretary on the average had 0.9 ingoing and 3.8 outgoing telexes on here desk waiting to be handled. The average response time for incoming telexes would be 21 minutes. For outgoing telexes, the average response time would be 2 hours and 50 minutes. An open queueing network assumes independent arrival of telexes, which may be questionable. Other assumptions could give a more detailed contention model.

The model is especially sensitive to some parameters, e.g. a change in  $d_p$  and  $x$  in Table 1 may change the secretary resource demands considerably. During modelling we initially used  $x = 2$  and  $d_p = 10$ , which gave an estimated response time of 43 minutes for outgoing telexes! After discussions with Statoil we found  $x = 1$  and  $d_p = 2$  to be more appropriate. Identification of this error and subsequent corrections were simple because of the structured method which had been used.

Some comments are in order for the resource demand vector  $D_{\text{Secretary\_desk}}$ , in the end of Section 3. Here the relationship between the number of pages read and the time taken are linear. This relationship might also be non-linear, i.e. for less than one page, reading is slower than for more than one page, except when the number of pages reaches a certain limit, when reduced attention reduces the speed again. Reading could also be combined with typing, just as the secretary could think while she walks. A possible solution to the latter problem is to model the legs and brain as separate resources. (This level of detail also has its humorous sides...) Some tasks occupy both resources, like sitting by the desk and reading, while other tasks like walking with some documents will only occupy the legs, leaving the brain free for other purposes. Thus, a human agent can perform several functions at a time, each using different aspects of the human agent. This is analogous to multiprogramming in a computer system, i.e. the disk and the CPU may be used by different processes.

### Relationship Between Workflow Model and Resource Model

The workflow model of Fig. 5 describes the process used to implement the `out.telex` operation. If we compare the resource model in Fig. 2 with the



**Fig. 6.** Relationship between the workflow model and the resource model in the manual solution. The links between the workflow model and the SP model are indicated with dashed lines.

workflow for outgoing telexes in Fig. 5, we realise that each of the six *subprocesses* in Fig. 5 uses the three resources `secretary_comms`, `secretary_proc` and `secretary_memory` in Fig. 2. This is illustrated in Fig. 6, where the `telex_secretary` resource has been replaced by the workflow for the `out_telex` operation. The resources `secretary_comms`, `secretary_proc` and `secretary_memory` in Fig. 6 represent specialised operations for the case worker with the specialised task *telex secretary*. The numbers in the complexity specifications between the resource `telex_secretary` and the resources `secretary_comms`, `secretary_proc` and `secretary_memory` in Fig. 2, can be derived via the mappings represented by the dotted lines in Fig. 6. For example, the complexity specification  $C_{\text{Secretary\_proc}}^{\text{Telex\_secretary}}$  in Section 3 can be identified in Fig. 6 as the dashed line between the subprocess **Retype the telex** and the resource `secretary_proc`.

## 5 GASCASE: Computerised Subsystem

This section presents an outline of an overall performance model of Lotus Notes, which forms the basis for the computerised workflows in the gas sales telex administration sector in Statoil. Figure 7 shows how two interacting subsystems can be identified in the Lotus Notes system.

Statoil uses a replication scheme with servers at three levels. The *replication subsystem* takes care of the replication of documents between the hierarchy of Lotus Notes servers, while the *client-server subsystem* deals with the interaction between a `server_level_3_server` and the `client`. To limit the amount of work, replication in Lotus Notes was not modelled. If a model of replication in Lotus Notes had been available, it could have been included in the overall performance model by straightforward extension.

The client server part of Fig. 7 is expanded in Fig. 8. In this case study, the server and the network are shared resources, whereas the client is not shared.

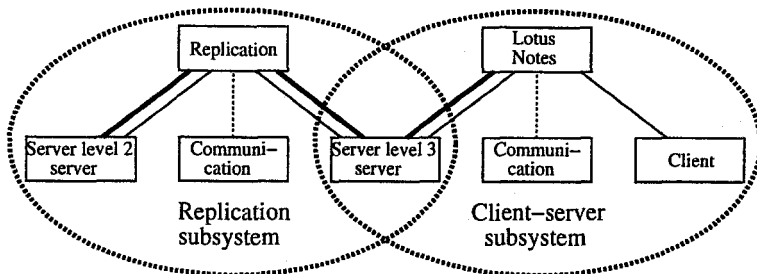


Fig. 7. Separate parts of Lotus Notes modelling

Thus, only the server and the network are potential performance bottlenecks. To limit the complexity, the server (and not the network) is the focal point of the case study.

With appropriate measurement tools, it would be possible to measure each resource of the model shown in Fig. 8. In practice, with the available tool SPM/2 from IBM, it was not possible to distinguish the contribution of the OS/2 resource or to measure usage of the network resource. The SP model was therefore reduced from three layers to two layers, as shown in Fig. 9. In this model the OS/2 layer of the Server has been subsumed within the Lotus Notes layer, attention being focused on the Server.

Three Lotus Notes operations correspond to the routine work of a case worker, were modelled, namely (1) `open_view`, showing the content of a database (2) `open_document` in a database (3) `store_document` in a database. The complexity specification for the relationship between the three user operations and the read sector and write sector operations on the server was [Kow95]:

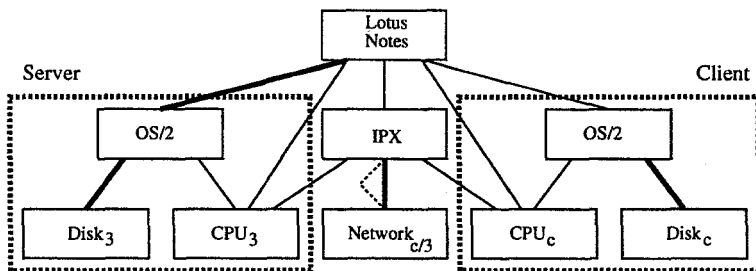


Fig. 8. SP model for the client-server part of Lotus Notes. Lotus Notes devolves work both on the server level 3 server (termed  $Disk_3$  and  $CPU_3$ ) and on the client (termed  $Disk_c$  and  $CPU_c$ ).

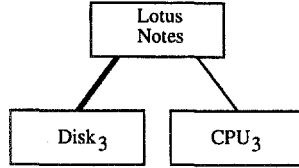


Fig. 9. The reduced SP model of the Lotus Notes server.

$$C_{Disk_3}^{Lotus\_Notes} = \begin{matrix} & & \textit{read\_sector} & \textit{write\_sector} \\ \textit{open\_view} & \left[ \begin{array}{cc} 0.0067s + 0.167n + 86 & 112 \\ 0.0012s + 8.325 & 42 \\ 9 & 0.00404s + 132 \end{array} \right. \\ \textit{open\_doc} \\ \textit{store\_doc} \end{matrix}$$

The parameter  $s$  is the average size of documents stored in the database measured in bytes and the parameter  $n$  is the number of documents stored in the database. Thus, an *open\_view* on a database with 2 000 documents and with a size of 5 kB for each document is equivalent to 450 physical sectors read from disk ( $0.0067 \cdot 5 \cdot 1\,000 + 0.167 \cdot 2\,000 + 86$ ). A similar complexity specification was made for the relation between Lotus Notes and the Server CPU.

To estimate these values, it was necessary to carry out a large number of measurements with a synthetic workload. These were done under controlled experimental conditions using a range of values for parameters such as  $s$  and  $n$ . A linear regression analysis was then performed to arrive at the results given. These results should not be used out of context, since the confidence intervals involved are of varying quality. A full description of the method is given in [Jai91]. Given more time for measurement, confidence intervals could have been improved. However, regression analysis is essentially a "black-box" technique. Close cooperation with software developers would allow knowledge of software structure to guide the measurement approach and produce complexity specifications more efficiently. In an ideal world, software vendors would be expected to provide such specifications for their products as a matter of routine.

## Dependencies

This model depends on the sector size of the particular disk. A refined model, where the operating system OS/2 is modelled as an extra software layer as described in Fig. 8, would allow the Lotus Notes work model to be hardware independent. The measurements may also be sensitive to size of cache and primary memory. Such installation properties could be included as parameters to the complexity specifications. Different versions of the software would require different complexity specifications.

These measurements are state dependent, and care should be taken to control such factors as the fragmentation of the disk and changes in database indexes, e.g. it is known that opening of a database view is more complex in Lotus Notes if there has been a change in database indexes, than if indexes have not been changed. The robustness of the complexity specification to changes in internal Lotus Notes tuning parameters could also be studied more carefully.

### Estimation of Resource Demands

The resource demands for the computerised solution need to be estimated. Reading one sector on the disk was estimated to take 1.6 milliseconds, and writing one sector was estimated to take 1.7 milliseconds [Bra96]. This is a rough estimation which is sensitive to physical placements of files on the disk and to the size of the files which are read and written. Similarly, resource demands were also estimated for the CPU. *When these resource demands are combined with the available complexity specifications and the load, the total resource demand for the computerised solution is 0.4 minutes to handle an incoming telex, and 2.3 minutes to handle an outgoing telex.* This compares with the time spend by the telex secretary and the contract specialist in the *manual* solution, which was estimated 3.6 minutes for incoming telex, and 30 minutes for outgoing telexes. The resource demand in the computerised solution is most sensitive to variations in the human processing. Thus, in Fig. 10, 2.0 minutes out of 2.3 minutes for outgoing telexes is used by the resource `secretary_desk_comp`.

However, in the computerised model some parts of the system were not considered. The network and the client were omitted in the model as shown in the parameterised model in Fig. 9. Unfortunately, there is a more serious omission, namely the effect of replication in Fig. 7. To determine whether the performance requirement at the beginning of Section 4 is satisfied, requires: (1) The addition of a static model for replication. (2) A contention model in which replication and telex handling workloads are combined. The contention model is a routine matter once the static models are available.

### Comparing the Manual and the Computerised Solution

Figure 10 shows an almost complete resource model for the manual and computerised solution, and it is interesting to compare these two SP models. The `customised_Lotus_Notes` resource handles the customisation of the basic operations in Lotus Notes, e.g. a customised operation like `view_mail_list` would use one `open_view` Lotus Notes operation. Note that the contract specialist is missing in the computerised solution. This is because the legwork performed by the component `contract_specialist` in the manual solution, is done by the network in the computerised solution of telex *handling*. In both solutions the contract specialist is heavily involved in the *creation* of telexes, but this is outside of the scope of the model.

Although the manual and the computerised solutions are fairly equal there are some differences: (1) The server (i.e. the resources `OS/23`, `Disk3` and `CPU3`) is

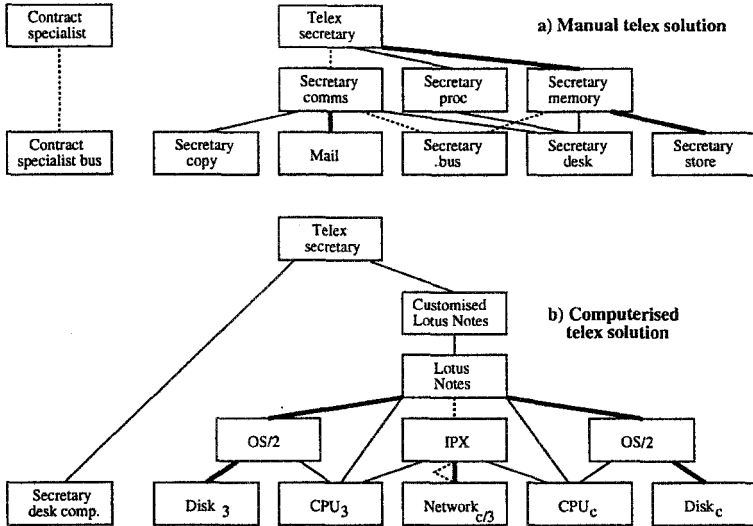


Fig. 10. An SP model for (a) the manual and (b) the computerised solution.

almost missing in the manual solution. However, an equivalent role is provided by the document archives in Statoil. (2) In the manual solution, the secretary performs some of the communication herself, namely the communication which is internal to the sales telex administration sector. The postman is used for external communication. In the computerised solution, all communication is performed via the (complex) network. (3) *Secretary\_bus* is so called by analogy with bus architectures for communication between for example disks and CPUs. This is not modelled explicitly in Fig. 10b). (4) *Secretary\_store* is an (active) interface to storage. This is the same situation as for the disk resource, which represents not only the physical disk, but also the interface logic.

Five of the nine primitive operations which the secretary performs in the manual solution need no manual work in the computerised solution. If the process in Fig. 5 had been designed with the computerised solution in mind it might have been revised, e.g. the **Transport\_from\_case\_worker** process could for example be ignored since it is now taken care of by the network, and is no longer an explicit part of the solution. Of the four remaining operations, the two operations for finding internal and external addresses are easier in the computerised solution. Reading is still manual, but the time consuming job of handling incoming receipts is no longer necessary. However, most of the saving with the computerised solution comes from using cut and paste of incoming telexes from external agents. Some extra work must be performed in the computerised solution, e.g. moving the cursor.

## 6 Practicability

To assess the practicability of the approach on the basis of the case study, three questions may be posed. (We are grateful to C.U. Smith for her suggestion on this point.) Firstly, is it *at all possible* to model human workflows, with no constraints on the size of the model, i.e. without any aggregations of operations or structural variables like  $n_p$  and  $x$ . The cost of making and maintaining such a model would in most cases be too large in practice. Therefore a *practical* model must aggregate the operations and structural variables of the real world. In the context of the basic framework in Section 2, this question can be positively answered for fairly routine workflows. If the workflow is routine or stable, then it is possible to find a mean for each aggregate parameter. The problems in identifying the mean are similar to the problems in estimating mean values for computer system models. Statistical techniques are needed. As an example of the differences and the similarity between manual and computer operations, see the discussion about walking in the end of Section 4. Parameters for the routine subprocesses in a template workflow can also be estimated. For ad-hoc workflows it is more difficult to say anything general about resource demands. At best they require a statistical view over a much longer period.

The second natural question is if the inaccuracy introduced by the chosen aggregations is acceptable. Here it is illuminating to compare alternative aggregations. A detailed model has the potential to be more accurate, if the input data is more accurate. If parameter accuracy is not sufficient, it is easy to aggregate the model before using it, producing a coarser model more suited to the available parameters. At the one extreme the two operations `in.telex` and `out.telex` could have been measured directly, in terms of the time used. This coarse approach is used for the human workflows in Section 5. There are also other possible approaches in between the coarse and the more detailed approach described here. Selecting a suitable granularity for modelling of human workflows is no different from selecting granularity when modelling a computer system.

The level of detail represented by the model was influenced by the use of SP in software modelling. It was chosen partly to explore the correspondence between human and computerised information processing in Section 5. The selected level of detail enables reuse of the static model components, when parameters representing e.g. office layout, skill levels, and information processing requirements vary. A more detailed model is more suited for reuse.

Given that it is possible to create a reasonably correct model with manageable size, the third question is if the accuracy in the computation of the model is reasonable. There are no built in assumptions in the method of solution of the static model, which would give rise to inaccuracy. For contention modelling, the usual considerations for approximate solution methods would apply. In general, development of a performance model is a tradeoff between accuracy and cost. In cases where the accuracy of a general model is too poor, it may be possible to make a customised model. The customisation may hamper the reusability of the model, and will therefore increase the cost of the model, sometimes to unacceptable levels.

## 7 Conclusions and Further Work

It has been demonstrated that, equivalent information processes, the one manual, the other computerised, can be modelled within the same framework. The framework places human beings at the top of the hierarchy, in the sense of "actors" or initiative takers. However, the framework also treats humans in the same way as computers, as "resources" used to process, communicate, store and retrieve information. In applying this, the following differences were noted between the human and the computer as "resources", within the context of this case study:

**Processor mobility** A human processor can temporarily associate his or herself with a passive device such as a filing cabinet, a desk, a copying machine, or a mail system, turning it into an active processing unit.

**Multi-tasking** A human processors exhibit different forms of multi-tasking than do computers, e.g. reading and typing, thinking and walking.

**Adaptive behaviour** Humans adapt to exceptional situations as overload in highly flexible ways, e.g. the secretary may carry several telexes in one go.

Processor mobility was easily handled by the SP framework. A detailed treatment of multi-tasking would require an elaborate contention model. However, as in the present case, simple approximations often suffice. Adaptive behaviour is related to the concept of ad-hoc processes, which requires further study.

Two potential developments of the approach can be identified. Firstly, further work on the Lotus Notes modelling could assist capacity planning of workflow servers. Secondly, an extension of the SP model in Fig. 2 could provide a generic model of case workers relevant for business process reengineering.

## References

- [Bra96] Gunnar Brataas. *Performance Engineering Method for Workflow Systems: An Integrated View of Human and Computerised Work Processes*. PhD thesis, Norwegian University of Science and Technology (NTNU), July 1996. Available from <http://www.fou.telenor.no/brukere/gunnarb>.
- [Hug88] Peter H. Hughes. SP principles. Technical report, STC Technology o59/ICL226/0, July 1988.
- [Jai91] Raj Jain. *The Art of Computer Systems Performance Analysis*. Wiley, 1991.
- [Kow95] Alexander Kowalski. Modelling of Workflow System Software for Performance Evaluation. Master's thesis, The Norwegian Institute of Technology, 1995.
- [Mor86] Gareth Morgan. *Images of Organizations*. Sage Publications Inc., 1986.
- [Smi90] Connie Umland Smith. *Performance Engineering of Software Systems*. Addison-Wesley, 1990.
- [VLP95] Jari Veijalainen, Aarno Lehtola, and Olli Pihlajamaa. Research Issues in Workflow Systems. In *Proceedings of the 8th ERCIM Database Research Group*, Trondheim, NORWAY, August 1995.