TheATLAS

MODULE SERIES

TRANSDISCIPLINARY ENGINEERING & SCIENCE

COMPOSITE SERVICE MODEL FOR THE PERFORMANCE DESIGN OF COLLABORATIVE SYSTEMS

Tod Gonsalves
Kiyoshi Itoh
Ryo Kawataba

ISSN: 1933-5423

TAM-Vol.3-No.1, 2007

The Academy of Transdisciplinary Learning & Advanced Studies
TheATLAS Publications

THEATLAS BOOK SERIES ON TRANSDISCIPLINARY ENGINEERING & SCIENCE

SERIES EDITOR-IN-CHIEF
A. ERTAS

TRANSDISCIPLINE: Integrating science and engineering principles

"...Today, complexity is a word that is much in fashion. We have learned very well that many of the systems that we are trying to deal with in our contemporary science and engineering are very complex indeed. They are so complex that it is not obvious that the powerful tricks and procedures that served us for four centuries or more in the development of modern science and engineering will enable us to understand and deal with them. We are learning that we need a science of complex systems, and we are beginning to construct it..."

Nobel Laureate Herbert A. Simon
Keynote Speech, 2000 IDPT Conference

TheATLAS Publishing

ISSN: 1933-5423

Published in the United States of America by

www.theatlas.org

Abstract

The conventional waiting line analysis theory deals with the long term steady state performance characteristics of single and multiple server structures that can be used to model the individual components of a queuing network. The single and multiple server structures, though powerful mathematical models in the waiting line analysis theory, do not address a wide range of real-life systems. These models represent, at best, dedicated servers. They cannot represent distributed or composite types of services that are encountered in collaborative systems. In this paper, we introduce a composite service model that can represent both distributed and composite types of services. The major advantage of the composite service model is the cost-effectiveness since it predicts the correct form of server allocation for collaborative jobs. Rather than having the personnel or equipment fixed at one particular service station, it is more cost-effective to have them shift among the nearby service stations to provide service.

We use the novel composite service model in the performance design and improvement of collaborative systems. Performance design of collaborative systems proceeds through the stages of modeling, performance evaluation and performance improvement. System modeling uses a descriptive approach that represents the workflow in the collaborative system in detail. System performance is simulated by a discrete event system simulator. Finally, system performance is improved by means of a Qualitative Reasoning Knowledge-Based System. The descriptive modeling and Qualitative performance design approach is intuitive. It circumvents the mathematical complexity that would appear if one were to attempt a performance analysis and design through a purely quantitative approach. The greatest advantage is that it is speedy. The designer can make a quick analysis of the system performance via simulation and the Knowledge-Based System proposes multiple scenarios which can be used to design and fine tune the system performance.

Keywords

Collaborative systems, performance design, performance evaluation, performance improvement, Qualitative Reasoning, composite service.

1. Introduction

System development goes through the well-known phases of requirement analysis, design and implementation [1]. While testing and evaluating the performance of the system is an important stage in the development of the system, it is often ignored for lack of time or lack of tools or both. Cooling [2] observes that while designing systems "designers are (blindly) optimistic that performance problems – if they arise – can be easily overcome". The designers test the performance of the system after the completion of the design and implementation stages and then try to remedy the problems. The main difficulty with this "reactive approach", Cooling states, is that problems are not predicted, only discovered. We have proposed a systematic performance design approach that can ***predict*** the operational problems and can suggest a viable solution at the requirement analysis stage of system development [3]. Briefly stated, the performance design method consists of a modeling stage (descriptive modeling), a performance evaluation stage (discrete event simulation) and a performance improvement stage (Qualitative Reasoning knowledge-based system).

This paper focuses on the application of the Qualitative performance design method to the performance tuning of collaborative systems. Hospitals, banks, universities, etc., are some examples of practical collaborative systems. The distinctive features of collaborative systems are exploited in the modeling, performance evaluation and performance improvement stages of the system development. Collaborative systems, for instance, exhibit the server-client property. The collaborators collaborate with one another to provide service to customers who enter the system. In other words, the collaborators constituting the collaborative system become providers of service and the clients become recipients of service. The client-server attribute of the collaborative systems can be efficiently modeled by the Multi-Context Map (MCM) technique. The MCM model of the system is a descriptive model that describes the workflow in collaborative systems in great detail. It captures the collaborators in the state of collaboration and gives an overall view of the collaborative tasks and activities in the system. The basic entity in the MCM is the "context". Within a given context the requestor ("Left-hand Perspective") requests the service providing unit or the performer ("Right-hand Perspective") to perform the collaboration activity. There exists an interface between the two through which **Token**, **Material** and **Information** (**TMI**) pass. An aggregate of contexts related to each other through the exchange of TMI flows gives rise to the MCM [4].

From the operational point of view, collaborative systems are discrete event systems. Discrete event systems, as opposed to continuous systems, are categorized as systems in which the phenomenon of interest or the state of the system changes in discrete steps of time [5]. A discrete event simulator simulates the operation of

the system and readily outputs the performance data of the system. Performance evaluation of the system is done by analyzing the performance data. The General Purpose System Simulator (GPSS) is a well-known discrete event simulator that we have used as a tool for the performance evaluation of collaborative systems [6]-[8].

The final stage in performance design is the use of the knowledge-based system (KBS) to evaluate the performance of the system and to suggest ways of fine-tuning it. Performance improvement is via the Qualitative Reasoning (QR) technique. Pioneered by de Kleer, Forbes and others, QR has been applied to a diversity of problems exhibiting qualitative behaviors such as electronic circuits, mechanical systems and economic systems [9] – [13]. We use the techniques of QR in establishing the qualitative rules that drive the diagnostic and bottleneck-resolving KBS.

The distinctive features of collaborative systems are exploited in the above-mentioned modeling, performance evaluation and performance improvement stages of the system development. These three stages are indispensable for system development, so much so that we have defined a generic core life cycle for the development of collaborative systems and have proposed an integrated environment in which they are seamlessly linked [14]. This approach has certain advantages. First of all, the MCM model is descriptive. The system designer need not formulate rigorous mathematical network equations. He/she can easily draw the MCM model of the system by analyzing the collaboration and workflow. The MCM model presents the bird's eye view of the entire collaborative system. The collaboration and the workflow are included down to the last detail. A converting tool semi-automatically converts the MCM draft into the GPSS program that simulates the operation of the system. The simulation data produced by the GPSS program are accessed by the QR-KBS, which diagnoses the problems in the system operation and suggests a viable improvement plan. The rules that make up the knowledge base of the KBS are qualitative in nature and are derived from the heuristics of the domain expert. The advantage of the QR approach is that it produces an improvement plan by overcoming the mathematical complexity that would occur if one were to write all the equations expressing the interactions among all the three-input and three-output contexts of the queuing network, and to solve them simultaneously.

The bottom line of the descriptive modeling-qualitative improvement approach is that it quickly provides a performance estimate of the system at the requirement analysis stage of the system, much before the implementation stage. The method is therefore "predictive" and not "reactive" as stated in the opening paragraph.

In this paper, we discuss the extension of the performance design method to include a myriad of complex scenarios that are encountered in collaborative systems operation. The focus of this paper is the new "composite service model" that we have developed to represent certain situations in collaborative systems operation

that cannot be represented by the existing conventional models. In collaborative systems, there is often a *distributed* type of service, i.e., there are situations in which the collaborator moves from place to place providing service; in addition, there is a *composite* type of service, i.e., there is an overlap of service times of different collaborators. For instance, a nurse in a small dental clinic may receive new patients, help out the dentist for a certain portion of time required to treat each patient and then move towards the accounts section. The single and multiple-channeled server models of the queuing theory cannot accurately depict this situation. They cannot represent the fact that the nurse is not present in the clinical room when she is busy providing service elsewhere.

The composite and distributed nature of the above type of service is accurately represented by the composite server model. In the queuing theory, the M/M/1 or M/M/c servers are treated as black boxes. Given the input, the server produces an output without giving any information about its internal state and dynamics. Further, the server is invariably identified with the service station. We distinguish the 'service stations' as the physical or logical places (in a collaborative system) where service is being provided and the 'sever' as a person or a piece of equipment engaged in the act of providing service. This enables the modeling of distributed and composite service. With this scheme, time sharing by the servers in the adjacent service stations becomes a possibility. The advantage of this model is that it enables us to predict the correct form of personnel allocation at the time-sharing proximate service stations. Rather than having personnel or equipment fixed at one particular service station, it is more cost-effective to have them shift among nearby service stations to provide service.

This paper is organized as follows: In section 2, we define the concept of composite service and develop the composite server model. In section 3, we introduce the composite service model into the MCM of the system and define the time-sharing or Perspective allocation scheme. In section 4, we derive the analytic expressions for the service time and for the elongation of the service time that is inevitable in Perspective allocation. In section 5, we present the validation of the composite service model. In sections 6 and 7, we discuss the performance evaluation and performance improvement scenarios, respectively. In section 8, we discuss the application of the composite service model to the performance design of a medium-sized computer shop floor. In section 9, we present the final conclusion of the paper.

2. Composite service model

The conventional waiting line analysis theory deals with single-channel and multiple-channel server models and their long-term operational measures [15]. These two structures are, at best, *simple* - the overall service time of each server cannot be broken down into individual components, and *dedicated* - each server does

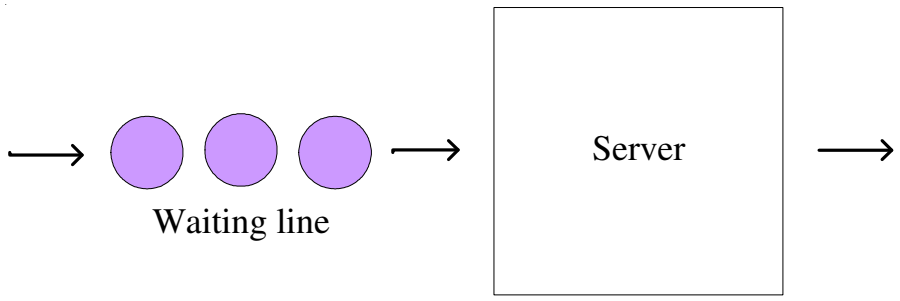


Fig. 1 Single-channel simple & dedicated server.

precisely one particular activity throughout the service time. Owing to these limitations, the server models of the conventional queuing theory are not sufficient to represent some of the complex scenarios that are encountered in collaborative engineering. We propose the composite service model to represent and simulate such complex scenarios. In our scheme, we separate the *server* from the *service station*. Having done this, we (re)define *dedicated* service, *distributed* service and *composite* service.

In the following sub-sections, we briefly introduce the characteristics of the well-known single-channel and multiple-channel server models. We then introduce the novel concept of composite service model and explain how this model can be used to represent server distribution and time-sharing. The section closes with a couple of real-life examples that further illustrate the composite service model.

2.1 Dedicated service

The single-channel model of the conventional queuing theory is represented in Fig. 1. Customers enter the system and if the server is busy, they enter a queue and wait for their turn to receive service. The customers depart from the system on receiving service from the server. Every queuing is characterized by a particular queuing discipline – First-In-First-Out (FIFO), Last-In- First-Out (LIFO), etc. A detailed list of queuing disciplines is found in [16].

If there are c number of identical servers in parallel, then the structure is referred to as a multiple-channel server with capacity c. Such a structure is shown in Fig. 2. Note that each of the c number of parallel servers has an *identical* service time.

These two models have two basic limitations. The first limitation is that the servers are entirely *dedicated*, i.e., no server can do more than a single activity throughout the system operation. This limitation can be overcome by our 'distributed model' described in section 2.2. The second limitation is that the servers are *simple*, i.e., the service time of the server cannot be broken down into individual components.

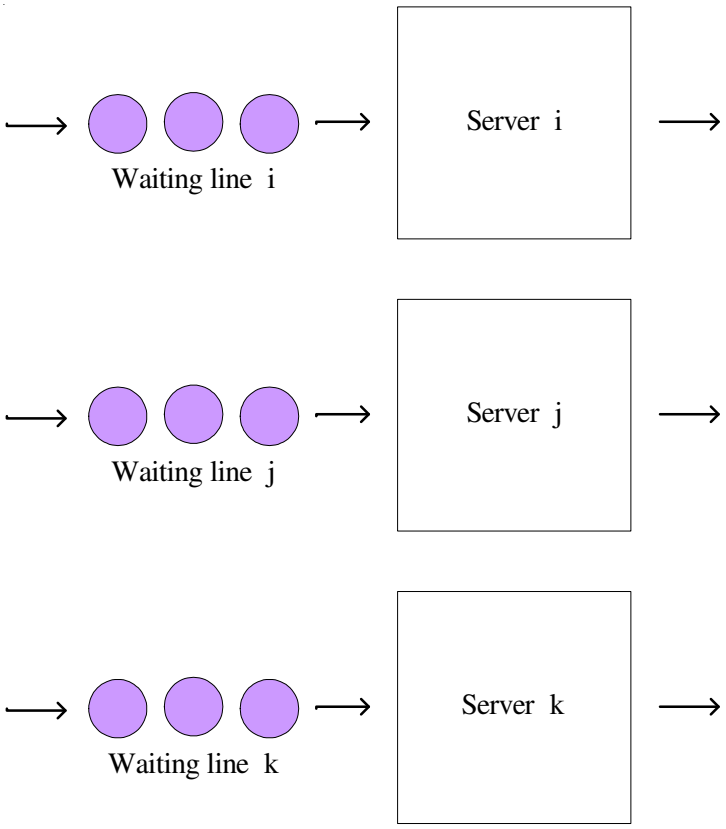


Fig. 2 Multiple-channel server.

The composite service model described in section 2.3 overcomes both these limitations. It can represent distributed service and composite service. In particular, we have used it in a time-sharing scheme.

2.2 Distributed service

In general, a server is anything/anyone (a machine, a robot, a computer, a human workers, etc.) that provides service to customers. Customers, in turn, are humans, calls, orders, punctured tires, etc., that enter a system seeking service. However, implicit in the activity of the server servicing a customer is the *place of activity*. This place of activity can be physical (doctor's clinic) or logical (attending to incoming calls or orders).

Queuing theory does not make a distinction between the server and the place of activity (or "service station"). The two are considered identical. The result is a

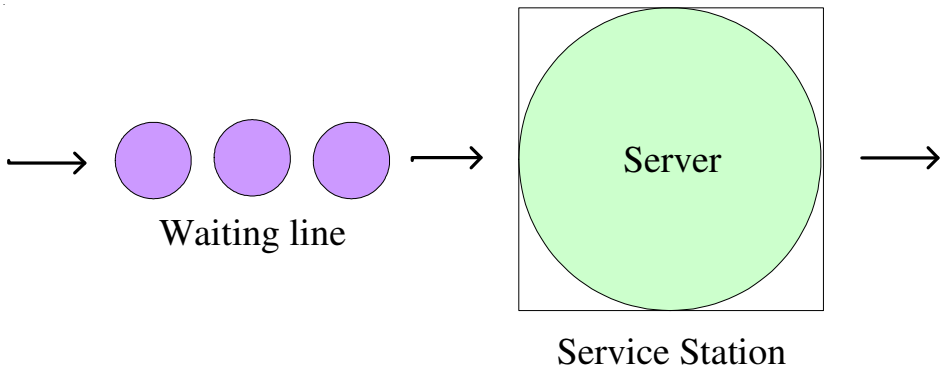


Fig. 3 The server is different from the service station.

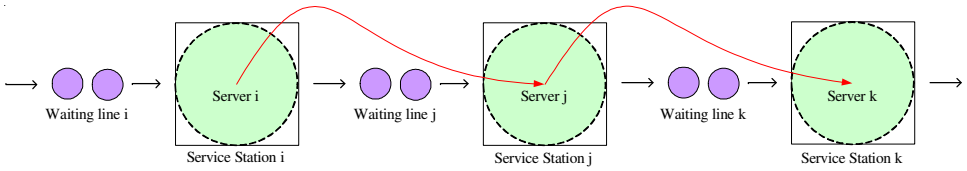


Fig. 4 Server distribution.

dedicated server. With such a dedicated server, the movement of the server from one place of activity to another cannot be represented. In our scheme, we separate the server from the service station. The service station represents the physical or the logical place or the situation in which service is provided. The server (person or machine or combination of both) is the agent that provides service at the service station. Fig. 3 shows our model where the server is treated different from the service station. In this diagram, the square represents the service station, and the circle in the middle represents the server or, very broadly, the service time of the server.

The advantage of representing the server as being separate from the service station is that server distribution can be modeled as shown in Fig. 4. The same server moves from one service station to another. The dotted circle indicates the absence of the server at a given point of time. A worker ("server") at the gas station, moving from car to car ("service stations") filling gas ("providing service") is an example of distributed service. This scenario of the server not being dedicated to one particular service station, but being engaged in providing service at different service stations is more conveniently depicted by Petri nets [17].

2.3 Composite service

Another limitation of the server models of the conventional waiting line analysis is that they can represent only a *simple* type of service. The service time is drawn from a suitable probability distribution and the dedicated server is engaged in service from the beginning till the end of the service time. In collaborative systems, however, we come across complex variations of services and service times, say for instance, when a team consisting of a doctor, a nurse and an anesthetist is simultaneously attending to a patient. This scenario cannot be represented by a simple multi-channel server, because the service times of the three specialists are neither identical nor synchronized.

The composite service model that we are going to describe in this section is capable of representing composite service and complex overlap of service times. Composite service, in general, can be defined as a service providing situation in which at least two different types of servers differing in specialization, provide a common service at least for some portion of the service time. A team consisting of a doctor, a nurse and an anesthetist working together on surgery, for example, is providing composite service. The three collaborators providing the common service of performing surgery have different specializations; they work together for at least a part of the entire surgery time. They could, in addition, have separate service times over and above the overlapping service time.

The composite service, in general, is made up of a varying *degree of contribution* by different specialists. The degree of contribution is determined by the duration of the service time. To keep our analysis simple, we consider only three degrees of contribution in our model, viz., first, second and third. The first degree of contribution is by the specialist whose service is indispensable to the service station; the other two types of specialists offer second and third degree contribution. In the above example, the doctor, the nurse and the anesthetist respectively offer first, second and third degree of contribution. The doctor is the primary server, while the nurse and the anesthetist are auxiliary servers. The duration of the service time of each of the three specialists is proportional to his or her degree of contribution.

Composite service may be schematically represented as in Fig. 5. Server1 is the main contributor to service; His/her/its service time is represented by the largest circle in the service station. Server2 and server3 are auxiliary; their service times, represented by inner circles, overlap with the service time of the main contributor.

2.4 Composite service time

When the service time is composite, each of the servers contributes towards the service time depending on his/her level of profession. S1 offers first degree of contribution and is indispensable for the service at the service station. S2 and S3 offer second and third degrees of contribution, respectively, and are auxiliary in

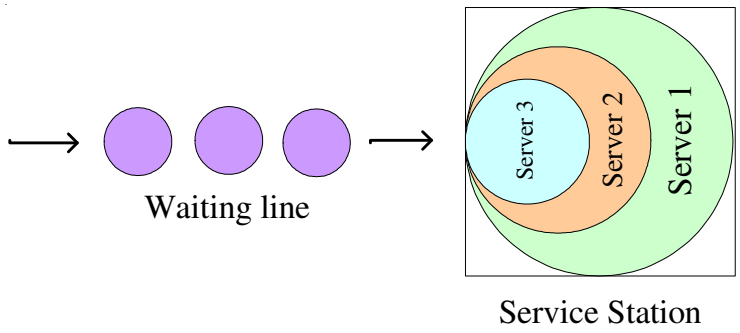


Fig. 5 Service station with composite service.

terms of service. The composite service time for each of these servers is illustrated in the following diagram (Fig. 6).

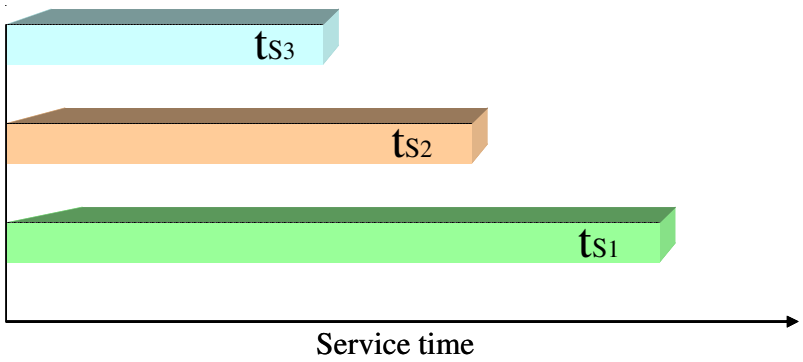


Fig. 6 Composite service time.

The symbols t_{S1}, t_{S2} and t_{S3} are respectively the service times of S1, S2 and S3. Since S1's role is primary, it follows that t_{S1} is the longest of the three service times (we assume it is equal to the service time necessary for the activity at the physical or logical place which the service station represents). Further, to keep the analysis of the composite service model simple, we assume that $t_{s_1} \geq t_{s_2} \geq t_{s_3}$.

2.5 Time-sharing by auxiliary servers

To keep the analysis simple, we further assume that the auxiliary servers need be present for service at a service station only for that part of the service time which overlaps with the service time of the principal server. That part of the time

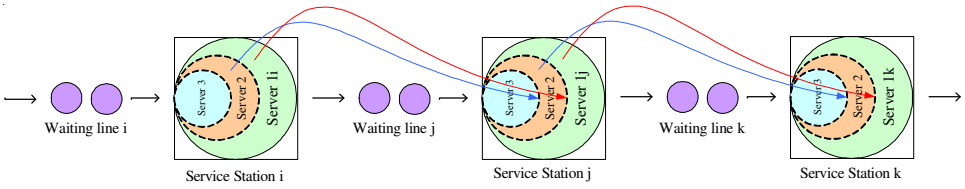


Fig. 7 Time-sharing by auxiliary servers.

which does not overlap with the service time of the principal server, is the free time of the auxiliary server. The auxiliary servers can spend their free time providing service at the neighboring service stations. This time-sharing scheme of the composite servers is illustrated in Fig. 7. At the service station i, for instance, that part of $S1_i$ which lies outside S_3 is the free time of S_3; similarly, that part of $S1_i$ which lies outside S_2 is the free time of S_2. Thus, the auxiliary servers possess some free time when $S1_i$ is still engaged in service. The auxiliary servers can spend their free time in providing service at the neighboring servers. $S1_i$, $S1_j$ and $S1_k$ are the primary servers at the service stations i, j, k, respectively. Their service time is represented by a solid circle, since they are dedicated servers. They do not move away from the service station to which they are assigned. The service time of S_2 and S_3 are represented by dotted circles. This implies that these servers are not dedicated to these service stations.

2.6 Examples of composite service

Composite service is very common in collaborative systems. As collaborators get together to provide service, it is natural that personnel with different specializations will work on a common task, each with a different degree of contribution. Table 1 lists real-life small-scale collaborative systems whose service is composite in nature. These systems can be modeled as composite servers. In the operation theatre, the surgeon, the nurse and the anesthetist work together for some portion of time that is required to perform the surgery. In our model, we assume that the doctor, the nurse and the anesthetist begin their service more or less at the same time (although, in practice, the anesthetist may begin his/her job of administering anesthetic to the patient before the surgery begins) and work together on the patient for a certain period of time. As the surgeon continues to perform the remainder of the surgical operation, the anesthetist finishes his/her job and can move on to perform other independent tasks. The nurse continues to work with the surgeon till the end of the surgery. However, she can intermittently attend to other tasks when the doctor is occupied with the surgery. The surgeon, the nurse and the anesthetist respectively offer first, second and third degrees of contribution to the surgery in terms of their skills and their average service times. The gasoline stand scenario is easier to understand. When a customer pulls in to fill his or her car with gas, the gas-filler

acts as the principal server; his/her service time lasts till the amount of gas ordered by the customer is filled. Meantime, the cleaner cleans the windshield, the windows, ashtrays, etc., and the casher attends to the payment transaction. These three operations require service times in decreasing order. This being the case, the auxiliary servers, namely, the cleaner and the cashier will have some free time when the principal server, namely the gas-filler is attending to a given car. These auxiliary servers can share their free time with the neighboring gas-filler attending to another car. This auxiliary time sharing scheme is cost-effective, since the management need not hire a cleaner and a cashier to accompany every gas-filler. Similar is the case with the cashier, the goods inspector and the goods wrapper working at the cash counter of a department store (refer to section 8 for details). The computer classroom composite service is an actual example practiced by our University Department while conducting computer programming classes to undergraduate students. Each student sits in front of the PC, listens to the instructions of the teacher and performs the programming task. The teacher proceeds with the lesson and those students who need additional assistance request it from the teaching assistants. Note that in this case there is only one type of auxiliary server ($S_2 = S_3$). Since the teaching assistants have enough free time during the lecture, they can move to the neighboring PC room and assist the teacher there (although the *time-sharing scheme* is not currently followed by the university). The last example in Table 1 is also a special case where S_3 is missing. The dentist is the principal contributor, while the nurse is the auxiliary contributor. The nurse works for some time together with the dentist per patient; she may assist another dentist during the remaining time of the composite service or attend to some other tasks[1].

The basic idea in this kind of composite collaborative service systems is that different professionals can concentrate on the same task for a certain amount of time and then move on to other related tasks while the main collaborator continues with the original task during each operation cycle. The advantage of this time-sharing is that fewer workers can be allotted to multiple tasks.

3. Composite service model in MCM model of collaborative systems

In this section, we show how to incorporate the composite service units into the larger model of collaborative systems, known as "Multi-Context Map" (MCM). The MCM is a network of contexts and junctions. Each context represents the collaborative place of activity. The servers working at the contexts are known as "Perspectives". Perspectives could be either principal or auxiliary. We group three neighboring contexts in the MCM so that the auxiliary Perspectives can do time-sharing. The time-sharing by the auxiliary Perspectives is known as Perspective

[1] The examples cited here are based on the business practices common in Japan; they may be different or may not be applicable in other countries.

Table 1 Examples of composite service.

Collaborative system	S_1	S_2	S_3
Surgery	surgeon	nurse	anesthetist
Gasoline stand	gas-filler	cleaner	cashier
Dept. store cash counter	cashier	goods inspector	goods wrapper
Computer classroom	teacher	teaching assistant 1	teaching assistant 2
Dental clinic	dentist	nurse	----

allocation. Section 3.1 describes time-sharing, while section 3.2 gives an example of time-sharing or Perspective allocation in MCM.

3.1 MCM model of collaborative systems

Our target systems are collaborative systems. By collaboration, we mean the coming together of individuals or groups to provide service to end-users. Business firms collaborating to enhance their business prospects, doctors and nurses in a clinic providing medical service to patients, teachers and staff members in a school providing educational service to students and manager and tellers in a bank providing financial service to customers are examples of collaborative systems.

The authors have proposed "Multi-Context Map" (MCM) as a modeling technique to model collaborative systems [4]. In this model, the basic unit of collaborative activity is called the "context". The context stands for a logical or a physical place in which the collaborators are engaged in collaborative activity. In each context, there is a collaborator who makes requests for service and a collaborator who responds to the request by way of performing the service. In the MCM terminology, the service requestor is called the "Left-Hand Perspective" (LHP) and the service performer is called the "Right Hand Perspective" (RHP). The RHP of a context then becomes the LHP to the related context(s) that follow it. Consequently, MCM represents a collective map of request-perform collaborative activities.

The term "Perspective" is used for the collaborator because as a performer of the activity the concerned collaborator usually has a particular standpoint in the entire collaborative process and usually has his/her own perspective in carrying out the assigned collaborative work. In practice, however, the MCM Perspective could be personnel or equipment or a combination of both. In addition, there are junctions that direct the workflow among the contexts. MCM is a topology of individual contexts inter-connected according to the logic of the real system which it represents.

Each context has three inputs and three outputs - *Token*, *Material* and *Information*. Material is anything (normally having weight and/or other tangible

physical properties) that enters the system seeking service. Patients in a hospital, customers in a supermarket, raw materials or semi-finished products on an assembly line, etc., are all 'Materials' in the MCM terminology. Each piece of Material has Information associated with it. The context server (RHP) processes the Material and the Information at the context and, through the outputs of the context, passes them on to the interconnected contexts. Token is the transmission signal necessary for communication among the interconnected contexts and for co-ordination of collaborative work. It may be an electronic signal, a computer message or a spoken word. At times, Token may not be explicit, but only implicit.

The sub-division of the workflow into three distinct categories of Token, Material and Information (collectively, TMI) is necessary for the detailed representation of the workflow in the system. Depending on the nature of the collaborative system, there may be cases in which three distinct flows simply do not exist. Take, for instance, a school teacher correcting the exam papers of her students. Although the papers have tangible dimensions and mass, they are not processed as material; hence there is no Material in this system; the contents of the papers and the written remarks and assessment of the teacher are clearly Information. Later, when the teacher delivers the scores to the exam office requesting the latter to prepare the mark-sheets, the message that she passes on by email or by a memo or by spoken words, is the Token that connects the collaborative activity of the teacher and of the exam office. It could be that the system is automated and the exam office immediately compiles the students' mark-sheets on receiving the students' scores. In such a case the Token is implicit.

The MCM model of a typical clinic system, for instance, consists of the reception, diagnosis, medical tests, prescription, accounts, etc., contexts (Fig. 9). In the MCM, these clinical contexts are interspaced with duplication, decomposition, serialization, branching, and synchronizing junctions that direct the TMI traffic inside the clinical system. The patient who enters the system for treatment is the Material, the patient's case papers with their contents are the Information. Token is the signal normally transmitted by word of mouth from a context that has finished treating the patient and processing the patient's Information at that particular context, to the next context, indicating that the latter may begin its service.

The distinct advantage of the MCM is that, being descriptive in nature, it provides an overall view of the collaboration that is taking place in the system. Further, this way of descriptive modeling lends itself to the application of Qualitative Reasoning (QR) in carrying out the performance improvement and design of the collaborative system. In simulating the system operation by the GPSS discrete event simulation language, the MCM contexts are treated as queuing service stations and the Perspectives are treated as servers. In [3], we have already proposed a performance design scheme of the MCM system model consisting of *simple* servers. In this

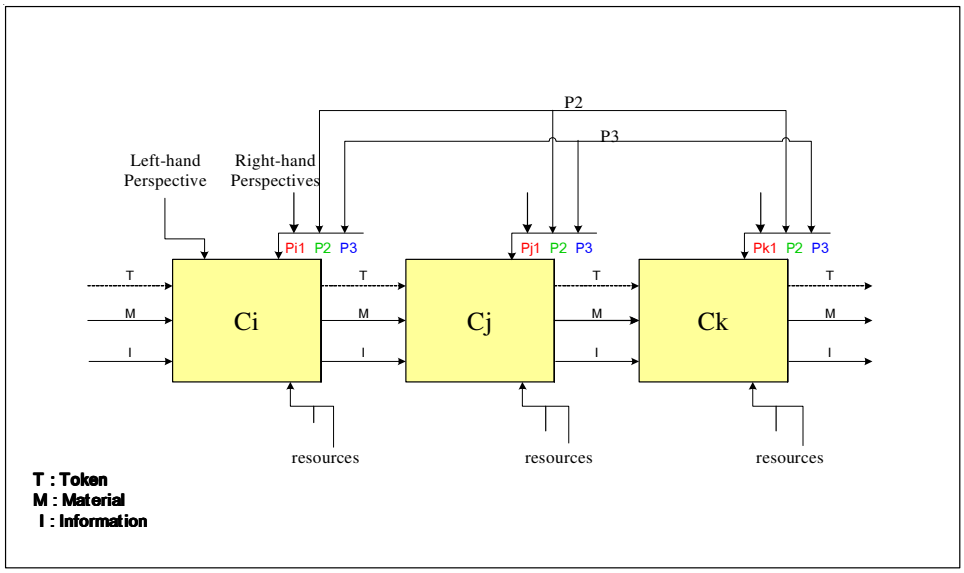


Fig. 8 Time-sharing or Perspective allocation in MCM contexts.

paper, we have attempted to carry on the performance design of collaborative systems that have composite form of service.

3.2 Perspective Allocation in MCM

Recall that the principal server (P1) is dedicated to the service station to which it is assigned, while the auxiliary servers (P2 & P3) are non-dedicated and posses a certain amount of free time. This free time can be utilized in time-sharing among the neighboring contexts. We group three contexts that have "proximity in function" and "proximity in distance" to engage in time-sharing by the auxiliary Perspective (Fig. 8).

Proximity in distance

The contexts are physically located so close to one another that the time spent by the auxiliary Perspectives in moving from one context to another in this group is negligible compared to the service time at each of the contexts. For example, the diagnosis, medical tests and prescription contexts in a clinic (Fig. 9) would not be very far from one another.

Proximity in function

The Auxiliary Perspectives are qualified for, or are capable of providing service specified by each of the three contexts. In the above example, nurses are qualified to assist at diagnoses and medical tests and to give prescriptions. The three contexts, viz. diagnosis, medical tests and prescription, are therefore proximate in function.

In the time-sharing scheme described in the preceding sub-sections, it is implicitly assumed that the service time of the primary server is not less than the service time of each of the auxiliary servers. This assumption ensures that the auxiliary servers have some amount of free time to provide service at the neighboring service stations while the primary server is still occupied at the main service station. The same assumptions hold for the time-sharing scheme or Perspective allocation in MCM. If C_i, C_j and C_k are the three contexts in MCM (Fig. 8) that are proximate in distance and/or function and $tP1_i$, $tP1_j$ and $tP1_k$ (refer to appendix for the rest of the notations) are respectively the service times of their primary Perspectives, then we assume that:

$$tP1_i \geq tP2 \geq tP3$$
$$tP1_j \geq tP2 \geq tP3$$
$$tP1_k \geq tP2 \geq tP3$$

The above assumptions guarantee that the auxiliary servers have free time to make time-sharing possible. The auxiliary Perspectives P2 and P3 initially spend their service times providing service at the C_i context. The remaining service time is spent at the proximate contexts C_j and C_k. In this way, the three proximate contexts participate in Perspective allocation by means of time-sharing.

3.3 Example of Perspective Allocation in MCM

Perspective allocation is illustrated by the MCM of a small clinic (Fig. 9). The three contexts participating in Perspective allocation are Diagnosis, Medical tests and Prescription contexts. These contexts are not far from one another in a small or medium-sized clinic; moreover, the nurses are qualified to assist in the jobs dealt with at each of these contexts, thus making the contexts proximate in distance and function. The respective primary contributors (P1) at these contexts are the doctor, the medical technician and the pharmacist. P2 could be a senior nurse and P3 a junior nurse. The senior and junior nurses assist the doctor during diagnosis. However, they need not be present in the clinical room during the entire diagnosis process; as the doctor is finishing the final stages of his/her diagnosis, the nurses can in the meantime assist in the neighboring medical tests and prescription positions. As noted above, the advantage of this model is that it enables us to predict the correct form of Perspective allocation at the contexts that share their service times. Rather than having personnel or equipment fixed at one particular context, it is more cost-effective to have them shift among nearby contexts to provide service.

4. Analytic expressions for service time in Perspective Allocation

There is give-and-take in service time as the auxiliary Perspectives participate in time-sharing among the proximate contexts. Any context that has extra auxiliary service time "lends" it to the proximate neighbors. The time-sharing scheme or Perspective Allocation functions efficiently as long as the auxiliary service time

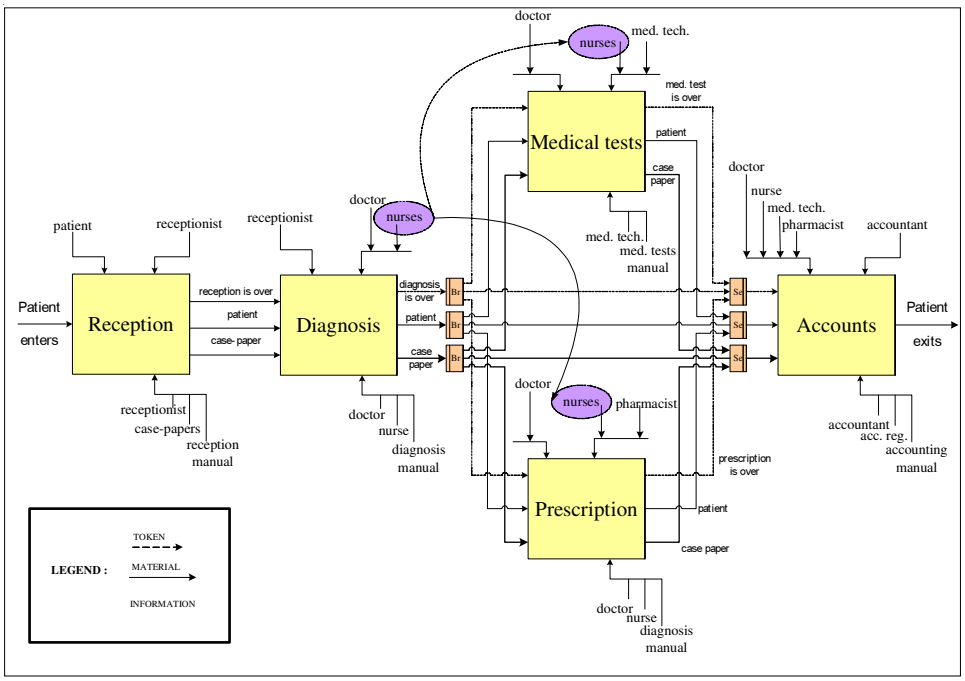


Fig. 9 Perspective allocation in MCM of a general clinic.

lending-borrowing balance is maintained. However, when the borrowing need becomes more than the capacity to lend, delays are introduced and we run into the time-prolongation problem. In this section, we examine the conditions under which delays occur in the time-sharing scheme and derive analytical expressions for those delays. In practice, a time-sharing scheme can be either in a serial or a parallel configuration. We derive analytical expressions for the time delays in each of these configurations.

4.1 Delays introduced due to auxiliary time-sharing

Let C_i, C_j, and C_k (refer to the appendix for notations) be the three proximate contexts among which the auxiliary Perspectives share their service times (Fig. 8). When the auxiliary servers, P2 and P3 are engaged in time-sharing among proximate contexts, prolongation of the composite service time may take place in the following way. Initially, all the servers start their service at C_i. After some time, P2 and then P3 move on to C_j. Similarly, after finishing service at C_j, they move on to C_k. When P3 is providing service at C_j and C_k, it may, however, spend more time at these contexts than what is available at C_i. In the meantime, P2 has moved back to C_i, and both P1 and P2 have to wait for the return of P3 to begin their composite service at C_i. The over-generosity of P3 thus introduces a delay δ_{3_i} in the total service time of

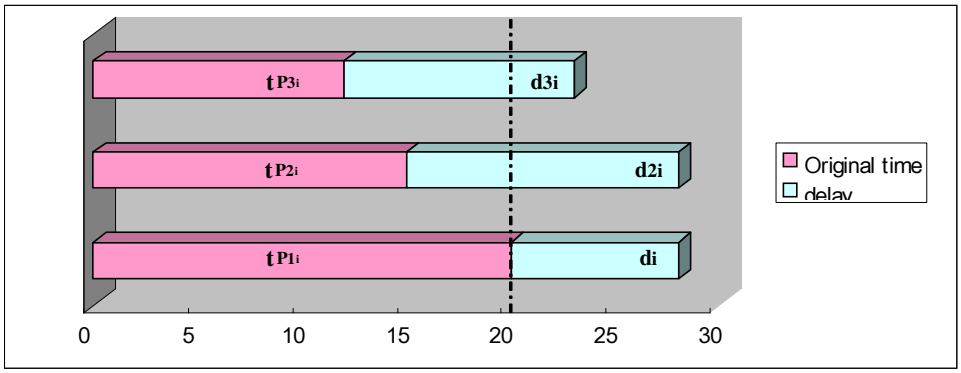


Fig. 10 Prolongation of composite service time (at C_i).

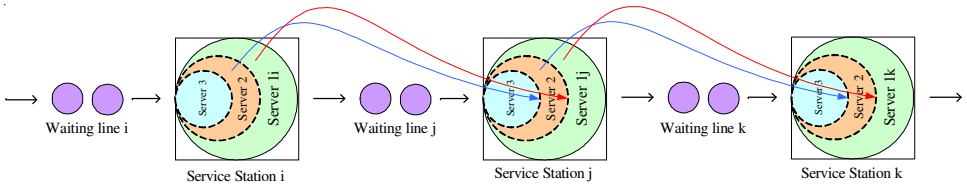


Fig. 11 Serial configuration of time-sharing servers.

C_i. Similarly, P2 spends its surplus time at C_j and C_k and if it overshoots the extra available time, it introduces a delay δ_{2_i} in the composite service time of C_i. The effective delay in the composite service time of C_i is, $\delta_i = \max(\delta_{2_i}, \delta_{3_i})$. The delays caused by P3 and P2 are shown in Fig. 10. The delays result in the prolongation of the service time of the main contributor P1 and, consequently, of the total service time of the context C_i. There are similar delays at C_j (δ_{2_j}, δ_{3_j}) and C_k (δ_{2_k}, δ_{3_k}) whenever the auxiliary free time is insufficient for time-sharing. In the next section, we shall derive analytic expressions for the delays at all the three contexts in the time-sharing group.

4.2 Time-sharing composite servers in series

In the serial configuration of time-sharing, the three contexts doing time-sharing are arranged serially as seen from the point of view of the arriving customers (Fig. 11).

Serial time-sharing takes place under the following context and inter-context assumptions:

Context assumptions

These assumptions follow from the definition of the principal and auxiliary service times.

$$tP1_i \geq tP2_i \geq tP3_i$$
$$tP1_j \geq tP2_j \geq tP3_j$$
$$tP1_k \geq tP2_k \geq tP3_k$$

Inter-context assumptions

In serial time-sharing, we consider C_i to be the mother context with the largest service times; C_j and C_k are sub-contexts with service times in the decreasing order. Hence, the following inequalities:

$$tP1i \geq tP1j \geq tP1k$$
$$tP2i \geq tP2j \geq tP2k$$
$$tP3i \geq tP3j \geq tP3k$$

In the serial configuration, we may say that C_i lends its extra auxiliary service to C_j and C_k; C_j lends its extra auxiliary time to C_k; finally, C_k lends its extra auxiliary service time back to C_i. However, if the time that is lent exceeds the free time available, then delays occur as follows:

Delay at Context C_i

$\delta 2_i$= (time spent by P2 at C_j + time spent by P2 at C_k) – (free time of P2 at C_i)

$$= (\varphi P2_j \, ts_j + \varphi P2_k \, ts_k) - (\varphi P1_i \, ts_i - \varphi P2_i \, ts_i) \quad (1)$$

$\delta 3_i$= (time spent by P3 at C_j + time spent by P3 at C_k) – (free time of P3 at C_i)

$$= (\varphi P3_j \, ts_j + \varphi P3_k \, ts_k) - (\varphi P1_i \, ts_i - \varphi P3_i \, ts_i) \quad (2)$$

Delay at Context C_j

$\delta 3_j$= time spent by P3 to C_k - free time of P3 at C_j

$$= (\varphi P3_i \, ts_i + \varphi P3_k \, ts_k) - (\varphi P1_j \, ts_j - \varphi P3_j \, ts_j) \quad (3)$$

$\delta 3_j$= time spent by P3 to C_k - free time of P3 at C_j

$$= (\varphi P3_i \, ts_i + \varphi P3_k \, ts_k) - (\varphi P1_j \, ts_j - \varphi P3_j \, ts_j) \quad (4)$$

Delay at Context C_k

$\delta 2_k$= time spent by P2 at C_i - free time of P2 at C_k

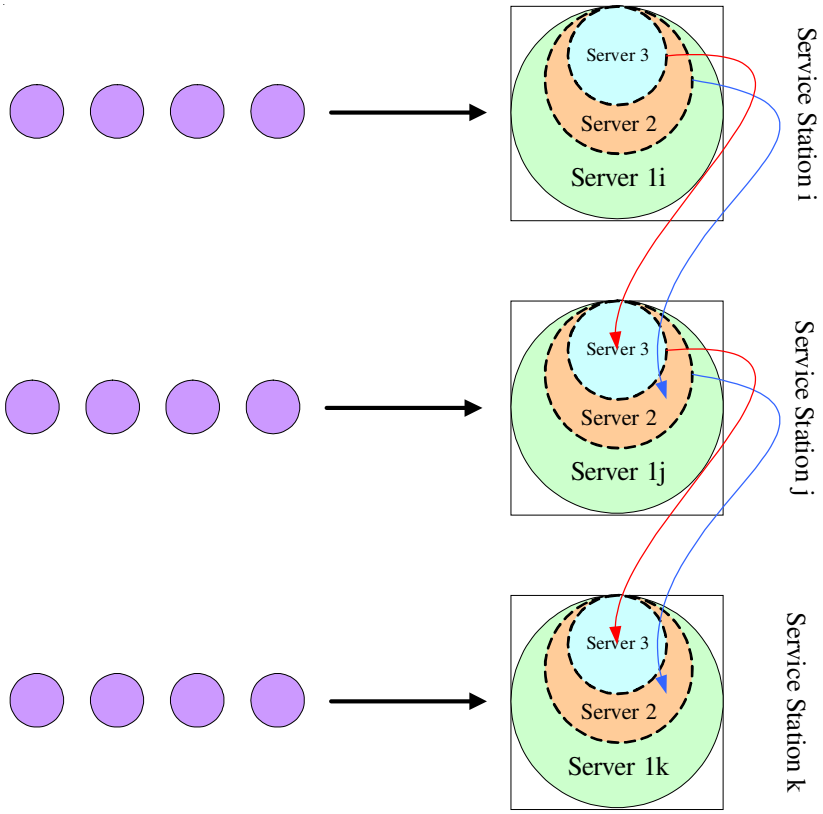
$$= \varphi P2_i \, ts_i - (\varphi P1_k \, ts_k - \varphi P2_k \, ts_k) \quad (5)$$

$\delta 3_k$= time spent by P3 at C_i - free time of P3 at C_k

$$= \varphi P3_i \, ts_i - (\varphi P1_k \, ts_k - \varphi P3_k \, ts_k) \quad (6)$$

4.3 Time-sharing composite servers in parallel

In the parallel time-sharing scheme, the context assumptions as well as the inter-context assumptions are relaxed, since the arrivals are totally independent of each other.



Since the servers are independent of one another as far as the arrivals are concerned, each server lends its extra auxiliary service time to the other two in the group. The delays are as follows:

Delay at Context C_i

δ_{2_i}= (time spent by P2 at C_j + time spent by P2 at C_k) – (free time of P2 at C_i)

$= (\varphi_{P2_j}\, ts_j + \varphi_{P2_k}\, ts_k) - (\varphi_{P1_i}\, ts_i - \varphi_{P2_i}\, ts_i)$ (7)

δ_{3_i}= (time spent by P3 at C_j + time spent by P3 at C_k) – (free time of P3 at C_i)

$= (\varphi_{P3_j}\, ts_j + \varphi_{P3_k}\, ts_k) - (\varphi_{P1_i}\, ts_i - \varphi_{P3_i}\, ts_i)$ (8)

Delay at Context C_j

δ_{2_j} = (time spent by P2 at C_i + time spent P2 at C_k) – (free time of P2 at C_j)

$= (\varphi_{P2_i}\, ts_i + \varphi_{P2_k}\, ts_k) - (\varphi_{P1_j}\, ts_j - \varphi_{P2_j}\, ts_j)$ (9)

δ_{3_j} = (time spent by P3 at C_i + time spent P2 at C_k) – (free time of P3 at C_j)

$= (\varphi_{P3_i}\, ts_i + \varphi_{P3_k}\, ts_k) - (\varphi_{P1_j}\, ts_j - \varphi_{P3_j}\, ts_j)$ (10)

Delay at Context C_k

δ_{2_k} = (time spent by P2 at C_i + time spent P2 at C_j) – (free time of P2 at C_k)

$= (\varphi_{P2_i}\, ts_i + \varphi_{P2_j}\, ts_j) - (\varphi_{P1_k} ts_k - \varphi_{P2_k}\, ts_k)$ (11)

δ_{3_k} = (time spent by P3 at C_i + time spent P3 at C_j) – (free time of P3 at C_k)

$= (\varphi_{P3_i}\, ts_i + \varphi_{P3_j}\, ts_j) - (\varphi_{P1_k} ts_k - \varphi_{P3_k}\, ts_k)$ (12)

4.4 Composite service times with delays

In certain cases, it could happen that the time available for auxiliary service (being borrowed from other contexts) is more than the time required for that auxiliary service. In such cases the δs in the above formulae are negative. The overall result would be a reduction in the total service time, making the latter shorter than the original service time. Since this is unrealistic, we should impose the condition that δ is strictly a non-negative term. Further, the overall delay is due to the auxiliary Perspective that is unavailable for service for a time period that is longer of the two. With these conditions, we have:

$$\delta = \max(\delta_2, \delta_3); \quad \delta_2 \geq 0, \delta_3 \geq 0 \tag{13}$$

The overall composite service time at each of the contexts is given below.

Service time at Context C_i

$$\delta_i = \max(\delta_{2_i}, \delta_{3_i}); \quad \delta_{2_i} \geq 0, \delta_{3_i} \geq 0 \tag{14}$$

$$tP1_i = \varphi_{P1_i}\, ts_i + \delta_i \tag{15}$$

$$tP2_i = \varphi_{P2_i}\, ts_i + \delta_{2_i} \tag{16}$$

$$tP3_i = \varphi_{P3_i}\, ts_i + \delta_{3_i} \tag{17}$$

Service time at Context C_j

$$\delta_j = \max(\delta_{2_j}, \delta_{3_j}); \quad \delta_{2_j} \geq 0, \delta_{3_j} \geq 0 \tag{18}$$

$$tP1_j = \varphi_{P1_j}\, ts_j + \delta_j \tag{19}$$

$$tP2_j = P2_j\, ts_j + \delta_{2_j} \tag{20}$$

$$tP3_j = \varphi_{P3_j}\, ts_j + \delta_{3_j} \tag{21}$$

Service time at Context C_k

$$\delta_k = \max(\delta_{2_k}, \delta_{3_k});\ \delta_{2_k} \geq 0, \delta_{3_k} \geq 0 \quad (22)$$

$$t_{P1_k} = \varphi_{P1_k}\, ts_k + \delta_k \quad (23)$$

$$t_{P2_k} = \varphi_{P2_k}\, ts_k + \delta_{2_k} \quad (24)$$

$$t_{P3_k} = \varphi_{P3_k}\, ts_k + \delta_{3_k} \quad (25)$$

4.5 Utilization of auxiliary Perspectives • CP1 and P2

Since the auxiliary Perspective P2 works at all the three related contexts during the entire cycle of operation, its total utilization would be the sum of the individual utilizations at each of the contexts, provided the sum is not greater than 1. If the sum exceeds 1, we coerce it to 1.

$$\begin{aligned} \rho_{P2} &= \rho_{P2_i} + \rho_{P2_j} + \rho_{P2_k}; \qquad \rho_{P2} \leq 1 \\ &= t_{P2_i} / ta_i + t_{P2_j} / ta_j + t_{P2_k} / ta_k \\ &= (\varphi_{P2_i}\, ts_i) / ta_i + (\varphi_{P2_j}\, ts_j) / ta_j + (\varphi_{P2_k}\, ts_k) / ta_k \\ &= \varphi_{P2_i}\, (ts_i / ta_i) + \varphi_{P2_j}\, (ts_j / ta_j) + \varphi_{P2_k}\, (ts_k / ta_k) \\ &= \varphi_{P2_i}\, \rho_i + \varphi_{P2_j}\, \rho_j + \varphi_{P2_k}\, \rho_k \end{aligned} \quad (26)$$

where, ρ_i, ρ_j and ρ_k are the utilizations of context C_i, C_j and C_k, respectively. Similarly, the utilization of the auxiliary Perspective, P3 is given by:

$$\rho_{P3} = \varphi_{P3_i}\, \rho_i + \varphi_{P3_j}\, \rho_j + \varphi_{P3_k}\, \rho_k \quad (27)$$

5. Composite Service Model Validation

As seen in section 4, composite servers engaged in time-sharing can be arranged either in a serial or a parallel configuration. In this section, we discuss the validation of the composite service in both of these configurations by comparing the simulation results with the theoretical expected results. The simulation scenarios cover the real-life situations in collaborative systems. The service times in collaborative systems may typically go from a minute to over an hour. In the variety of scenarios presented in the following section, we randomly select the service times from a minimum of 1 minute to a maximum of 100 minutes. In each case, the average inter-arrival time is randomly selected to be 102% - 140% of the maximum average service time, since stability conditions for an M/M/1 server requires that the average inter-arrival times be greater than the average service time.

We consider three different statistical distributions of the average inter-arrival time and average service time random variables, viz., constant, uniform and exponential, since these distributions closely represent real-life situations modeled

Table 2 Validation of the composite service model in serial configuration.

Distribution	Sim Num	Ci							Cj							Ck							P2 PARAMETERS								P3 PARAMETERS							
		ta	ta ± offset	tP1i	tP1i± offset	tP2i	tP2i± offset	tP3i	tP3i± offset	tP1j	tP1j± offset	tP2j	tP2j± offset	tP3j	tP3j± offset	tP1k	tP1k± offset	tP2k	tP2k± offset	tP3k	tP3k± offset	ζ2i Cal	ζ2i Sim	ζ2j Cal	ζ2j Sim	ζ2k Cal	ζ2k Sim	ζP2 Cal	ζP2 Sim	ζ3i Cal	ζ3i Sim	ζ3j Cal	ζ3j Sim	ζ3k Cal	ζ3k Sim	ζP3 Cal	ζP3 Sim
	1	60		52		43		39		33		25		21		21		18		17		34	34	10	10	40	40	1	0.99	25	25	5	5	35	35	1	0.99
C	2	89		74		65		57		39		38		31		30		29		22		58	58	28	28	64	64	1	0.99	36	36	14	14	49	49	1	0.99
O	3	55		45		28		26		41		26		25		36		18		17		27	27	3	3	10	10	1	0.99	23	23	1	1	7	7	1	0.99
N	4	28		22		18		15		16		11		10		11		7		7		14	14	2	2	14	14	1	0.99	10	10	1	1	11	11	1	0.99
S	5	73		58		43		38		37		33		29		26		10		10		28	28	6	6	27	27	1	0.99	19	19	2	2	22	22	1	0.99
T	6	93		79		4		3		60		4		1		59		4		1		0	0	0	0	0	0	0.13	0.13	0	0	0	0	0	0	0.05	0.05
A	7	23		17		5		4		15		2		1		14		2		1		0	0	0	0	0	0	0.39	0.39	0	0	0	0	0	0	0.26	0.26
N	8	26		18		15		11		12		2		1		11		2		1		1	0	0	0	6	2	0.74	0.73	0	0	0	0	1	0	0.5	0.5
T	9	59		52		15		14		12		12		4		6		1		1		0	0	1	0	10	10	0.5	0.47	0	0	0	0	9	4	0.33	0.32
	10	26		22		11		8		3		3		2		1		1		1		0	0	1	0	11	0	0.58	0.57	0	0	0	0	8	0	0.42	0.42
	1	22	5	17	5	15	5	14	4	15	6	15	6	13	5	15	6	7	4	7	4	20	20	7	7.4	7	7.1	1	0.99	17	17	5	5.6	6	5.9	1	1.00
U	2	49	5	41	2	29	2	26	2	29	7	21	4	19	3	27	3	18	6	14	5	27	27	10	10	20	20	1	0.99	18	18	4	4.9	13	12	1	1.00
N	3	46	5	39	7	30	5	29	4	32	2	30	2	27	2	32	8	13	2	11	2	34	34	11	11	11	11	1	0.99	28	28	6	6.2	8	7.9	1	1.00
I	4	39	2	31	5	20	1	19	2	26	2	18	2	17	2	22	4	13	3	12	2	20	20	5	5	11	11	1	0.99	17	17	3	3.1	9	8.9	1	1.00
F	5	72	5	57	5	55	5	35	4	54	6	43	6	35	6	45	6	37	4	24	5	78	78	26	26	47	47	1	0.99	37	37	5	5.5	14	13	1	1.00
O	6	76	3	61	1	25	2	23	5	60	7	10	7	9	2	14	7	5	4	5	4	0	0	0	0	16	0	0.53	0.53	0	0	0	0	14	0	0.49	0.49
R	7	99	1	84	3	64	7	53	3	80	3	7	3	4	3	60	7	7	1	3	1	0	0	0	0	11	0	0.78	0.78	4	0	1	0	0	0	0.61	0.61
M	8	64	4	45	7	14	7	8	1	17	4	14	5	7	2	16	2	10	2	6	5	0	0	7	0.6	8	2.5	0.62	0.59	0	0	0	0	0	0	0.33	0.33
	9	47	1	34	5	32	4	25	1	32	9	4	2	3	1	18	1	2	1	2	1	4	0.9	0	0	16	1.2	0.83	0.81	0	0	0	0	9	0.1	0.64	0.64
	10	31	4	23	7	6	5	4	2	8	7	4	3	3	1	7	2	3	1	2	1	0	0	0	0.6	2	0.6	0.44	0.42	0	0	0	0.2	0	0	0.29	0.29
	1	41		30		22		21		29		21		20		17		15		12		30	30	8	8.1	21	21	1	0.99	25	25	4	4	17	17	1	0.99
E	2	33		24		21		17		14		11		9		10		9		8		19	19	7	7	21	21	1	0.99	12	12	4	4.1	16	16	1	0.99
X	3	95		81		55		53		39		38		32		31		24		22		38	38	24	24	49	49	1	0.99	28	28	16	16	45	45	1	0.99
P	4	21		19		17		9		15		14		9		15		9		9		23	23	9	9	12	12	1	0.99	10	10	4	4	4	3.9	1	0.99
O	5	70		63		55		42		56		45		39		45		42		21		81	81	32	32	53	53	1	0.99	41	41	5	5	19	19	1	0.99
N	6	89		79		40		39		57		14		11		31		13		1		0	0	0	0	23	0	0.78	0.78	0	0	0	0	10	0	0.6	0.6
E	7	70		47		45		39		14		9		7		5		1		1		10	0.1	0	0	42	0.1	0.81	0.81	2	0	0	0	36	0.1	0.71	0.71
N	8	69		51		15		11		15		14		11		13		5		4		0	0	5	0.2	8	1	0.54	0.53	0	0	1	0	3	0.2	0.43	0.41
T	9	99		92		25		13		35		21		11		31		20		3		0	0	7	0	15	1.4	0.69	0.68	0	0	0	0	0	0	0.29	0.3
	10	93		77		39		38		43		20		8		27		6		5		0	0	0	0	19	0.1	0.72	0.72	0	0	0	0	17	0	0.57	0.57

by the queuing theory [18]. We find that the values of the time-sharing parameters closely match the expected theoretical values.

5.1 Time-sharing composite servers in series

In this configuration, the three service stations engaged in time-sharing are arranged in series as seen from the point of view of the incoming customers. The customers wait in queues for service in front of the first service station. Upon receiving service at this station, they proceed to stations 2 and 3 in order (Fig. 11). This configuration can, in practice, be realized in a manufacturing setup. P2 can be the larger robotic arm moving from stations 1, 2, 3 in succession; similarly, P3 can be likened to the minor arm serving for a relatively shorter time at the three service stations.

The validation results are shown in Table 2. The average inter-arrival times, and the average service times for each of the servers are drawn from a random probability distribution. Viewed vertically, the table is divided into three sections, with each section containing ten sets of arrival times (t_a) and service times (t_s) culled from a particular distribution – constant, uniform and exponential. Viewed horizontally, the table consists of three different sections: the leftmost section represents the randomly selected t_a and t_s for the three time-sharing contexts Ci, Cj and Ck. The t_a " offset and t_s " offset values in this section correspond only to the uniform distribution. The second section to the right represents the calculated and simulated time-sharing parameters (delays and utilizations) of the auxiliary Perspectives P2. Similarly, the rightmost section represents the calculated and simulated time-sharing parameters of the auxiliary Perspectives P3.

The first 5 simulation runs in the P2 parameters section of each of the distribution

show a time delay (δ_2 Cal) at each of the time-sharing contexts, calculated by using the analytical expressions presented in section 4. These values closely match with the corresponding simulated values of P2 (δ_2 Sim). The last two columns in the P2 parameters section compare the expected (ρ_{P2} Cal) and the simulated (ρ_{P2} Sim) values of the utilization of P2. Since the delays in the first five simulation runs are quite long, the utilization is nearly 1. In the last five simulation runs, when the delays are nearly zero[2], the calculated and the simulated values of the utilization are nearly the same. Similar results are presented for the P3 parameters (δ_3 Cal, δ_3 Sim; ρ_{P3} Cal, ρ_{P3} Sim) in the rightmost section. In each of the distributions section, the first five simulation runs highlight the *delays*, while the last five simulation runs highlight the *utilizations*.

The P2, P3 delays and utilizations have a confidence level of 95% within a width of 0.5 (refer to appendix B for the estimation of confidence intervals). As seen in Table 2, the simulated values closely match the theoretical expected values, thus validating the composite service model for collaborative systems.

5.2 Time-sharing composite servers in parallel

As seen from the point of view of the incoming customers, the three servers are arranged in parallel. There are three separate queues in the system. The queues are independent (Fig 12). Table 3 shows different set of observations for three different distributions. Table 3 is similar to Table 2, except that there are three independent arrivals. Here, too, the simulation results match the expected theoretical results with 95% confidence level.

Table 3 Validation of the composite service model in parallel configuration.

Distribution	Sim Num	Ci							Cj							Ck							P2 PARAMETERS						P3 PARAMETERS					
		t_{ai}	t_{ai} + offst	t_{P1i}	t_{P1i} + offst	t_{P2i}	t_{P2i} + offst	t_{P3i}	t_{P3i} + offst	t_{aj}	t_{aj} + offst	t_{P1j}	t_{P1j} + offst	t_{P2j}	t_{P2j} + offst	t_{P3j}	t_{P3j} + offst	t_{ak}	t_{ak} + offst	t_{P1k}	t_{P1k} + offst	t_{P2k}	t_{P2k} + offst	t_{P3k}	t_{P3k} + offst	$\bar{\omega}^{i}$ Cal	$\bar{\omega}^{i}$ Sim	$\bar{\omega}^{j}$ Cal	$\bar{\omega}^{j}$ Sim	$\bar{\omega}^{k}$ Cal	$\bar{\omega}^{k}$ Sim	ρ^{P2} Cal	ρ^{P2} Sim	$\bar{\omega}^{i}$ Cal	$\bar{\omega}^{i}$ Sim	$\bar{\omega}^{j}$ Cal	$\bar{\omega}^{j}$ Sim	$\bar{\omega}^{k}$ Cal	$\bar{\omega}^{k}$ Sim	ρ^{P3} Cal	ρ^{P3} Sim
	1	93		77		49		37		89		77		85		45		98		76		35		29		92	92	92	93	93	1	0.99	34	34	34	34	35	35	1	0.99
C	2	70		47		25		19		66		65		28		23		98		83		69		57		75	75	57	57	39	39	1	0.99	52	52	34	34	16	16	1	0.99
O	3	98		94		54		2		89		85		69		64		88		80		50		40		79	79	88	88	93	93	1	0.99	12	12	21	21	26	26	1	0.99
N	4	99		90		44		34		73		71		45		36		86		73		65		54		64	64	83	83	81	81	1	0.99	34	34	53	53	51	51	1	0.99
S	5	26		18		16		12		21		20		18		12		8		7		4		4		20	20	18	18	31	31	1	0.99	10	10	8	8	21	21	1	0.99
T	6	95		72		3		2		95		94		17		13		78		53		15		1		0	0	0	0	0	0	0.39	0.39	0	0	0	0	0	0	0.17	0.17
A	7	39		27		8		1		79		76		19		18		99		68		31		24		31	1.7	0	0	0	0	0.76	0.75	16	0.15	0	0	0	0	0.49	0.49
N	8	30		22		3		2		93		87		11		8		28		24		9		3		1	0	0	0	0	0	0.54	0.54	0	0	0	0	0	0	0.26	0.26
T	9	96		69		17		2		54		52		5		3		87		77		59		34		12	0.55	29	1.37	4	0	0.94	0.94	0	0	0	0	0	0	0.46	0.46
	10	64		52		2		1		79		73		16		3		36		24		18		13		0	0	0	0	12	0.71	0.74	0.73	0	0	0	0	0	0	0.41	0.41
	1	88	8	81	6	51	7	50	5	90	1	79	5	69	4	25	1	99	3	97	2	58	5	39	4	97	97	99	99.2	81	81.1	1	0.99	33	32.8	35	35	17	16.9	1	0.99
U	2	71	3	51	7	43	1	39	2	37	4	35	1	33	4	14	5	33	4	26	6	21	4	19	1	46	46	62	62.1	71	71.1	1	0.99	21	20.9	37	37	46	46	1	0.99
N	3	85	9	68	5	54	6	51	1	82	10	78	7	63	6	56	4	62	5	48	8	25	2	18	3	74	74	64	63.6	94	94	1	0.99	57	57	47	47	77	77.1	1	0.99
I	4	37	2	27	2	23	4	21	5	56	1	53	1	34	3	28	4	59	5	41	4	15	7	13	3	45	45	19	19	31	31	1	0.99	35	34.9	9	8.9	21	21	1	0.99
F	5	70	2	59	10	54	2	49	5	26	7	25	6	19	1	17	1	21	3	19	8	14	1	11	3	28	28	62	62	68	68	1	0.99	18	17.9	52	52	58	58	1	0.99
O	6	95	9	80	7	33	1	6	1	53	4	51	5	6	2	3	1	99	5	89	4	30	6	7	3	0	0	18	0.17	0	0	0.76	0.76	0	0	0	0	0	0	0.19	0.19
R	7	59	8	26	9	16	2	14	2	47	5	31	1	14	3	9	2	93	5	64	8	7	3	3	2	11	0.64	6	0.03	0	0	0.62	0.64	0	0.07	0	0	0	0	0.46	0.46
M	8	91	7	77	3	47	5	11	5	91	2	82	7	6	4	5	3	85	5	62	9	20	2	13	2	0	0	0	0	11	0.41	0.82	0.82	0	0	0	0	0	0	0.33	0.33
	9	52	6	38	5	15	5	9	4	76	7	73	6	9	3	8	3	57	1	52	10	3	2	2	1	0	0	0	0	0	0	0.45	0.45	0	0	0	0	0	0	0.31	0.31
	10	99	6	84	7	16	1	10	2	91	5	64	4	19	6	5	4	87	3	67	9	15	2	14	1	0	0	0	0	0	0.54	0.54	0.54	0	0	0	0	0	0	0.32	0.32
	1	4		16		14		14		83		77		46		35		40		37		35		34		81	81	20	20.1	60	60.1	1	0.99	69	69.1	8	8	48	48.1	1	0.99
E	2	71		61		33		26		36		34		32		25		70		59		24		19		30	30.1	57	57.1	32	32.1	1	0.99	11	11	38	38	13	13	1	0.99
X	3	73		66		44		24		86		80		51		41		97		74		41		37		72	72.1	58	58.1	64	64.1	1	0.99	38	38	24	24	30	30	1	0.99
P	4	96		87		59		45		71		67		43		42		57		43		38		15		55	55.1	75	75.1	99	99.1	1	0.99	17	17.1	37	37	61	61.1	1	0.99
O	5	91		83		39		37		60		48		25		21		77		64		33		32		16	16	51	51	35	34.9	1	0.99	9	9.16	44	44	28	28.1	1	0.99
N	6	95		68		17		5		84		82		14		7		96		87		10		8		0	0	0	0	0	0	0.47	0.47	0	0	0	0	0	0	0.24	0.24
E	7	84		51		11		2		93		46		5		3		86		51		29		5		0	0	1	0	0	0	0.55	0.55	0	0	0	0	0	0	0.14	0.14
N	8	51		34		10		9		28		27		6		2		30		25		6		4		0	0	0	0	0	0	0.68	0.68	0	0	0	0	0	0	0.45	0.45
T	9	88		73		4		3		23		22		14		12		76		59		5		5		0	0	3	0	0	0	0.76	0.76	0	0	0	0	0	0	0.66	0.66
	10	35		29		7		6		39		30		6		5		61		46		12		10		0	0	0	0	0	0	0.61	0.61	0	0	0	0	0	0	0.52	0.52

[2] Under our assumptions, when at least one of the expected delays becomes nearly zero, the rest of the delays cannot be calculated theoretically. The simulated delays depend entirely on the arrival time.

6. Performance Evaluation

The system MCM is semi-automatically converted into the General Purpose Simulator System (GPSS) program. GPSS is a well-known discrete event simulator. The GPSS program produces a simulation data sheet of system performance. The KBS refers to the simulation data sheet and diagnoses the bottlenecks arising in the system operation. The different performance metrics and the different types of bottlenecks in the system operation, together with the bottleneck classification scheme, are presented in this section.

6.1 Performance metrics and bottleneck definition

MCM is a queuing network consisting of contexts and junctions. The contexts act as servers and the junctions act as workflow controllers among the contexts. If there is a problem with the flow of the entities in the system, it could lead to bottlenecks. In addition, the inappropriate allocation of auxiliary Perspectives to the proximate contexts engaged in time-sharing could also lead to bottlenecks. We select queue-length (q) and server utilization (ρ) as the performance indicators of the system. A context is defined to be a bottleneck if it satisfies any one of the following inequalities:

$$\rho \geq 0.7 \tag{28}$$

$$\rho \leq 0.3 \tag{29}$$

$$q \geq 1.0 \tag{30}$$

$\rho \geq 0.7$ is an indication that the system operation is becoming loaded. It is a sign that the personnel are being kept occupied for over seventy percent of the total operation time of the system. On the other hand, $\rho \leq 0.3$ is an indication of the low efficiency of the system; it implies that the system resources are underutilized. As for the length of the queues, $q \geq 1.0$ implies that in the long run customers will keep joining the queues to such an extent that the system will never be able to cope up with the demand for service. A context with normal operation would be one with ρ within the above fail-safe limits and with queue less than unity.

There are practical reasons for selecting the above values for the performance metrics. As the arrival rate increases, the utilization approaches 1 and the number of jobs in the system and response time approach infinity. This infinite response time is the key reason for not subjecting a server to 100% utilization [19]. Since the arrivals are random, there could be times when the inter-arrival rate suddenly shoots up and overwhelms the system. In such a critical period, the server response time will be infinite and the system will not be able to cope with the incoming arrivals. Hence the experts' heuristics suggest that 0.7 be the fail-safe landmark [20]. However, $\rho = 0.7$ is only a relative landmark; it could vary from system to system (and from one management policy to another). Some executives could even raise it

to 0.8 or 0.9. What we give here is a theoretical yardstick that should hold for any general type of collaborative system operation.

6.2 Bottleneck classification

The following is a detailed list of all the possible bottlenecks in the system operation when a group of three proximate contexts, C_i, C_j and C_k participate in time-sharing.

$\rho_{P1_i} \leq 0.3$; OR	$\rho_{P1_i} \geq 0.7$;	AND/OR	$q_i \geq 1$	(31)
$\rho_{P1_j} \leq 0.3$; OR	$\rho_{P1_j} \geq 0.7$;	AND/OR	$q_j \geq 1$	(32)
$\rho_{P1_k} \leq 0.3$; OR	$\rho_{P1_k} \geq 0.7$;	AND/OR	$q_k \geq 1$	(33)
$\rho_{P2} \leq 0.3$; OR	$\rho_{P2} \geq 0.7$;			(34)
$\rho_{P3} \leq 0.3$; OR	$\rho_{P3} \geq 0.7$;			(35)

The bottlenecks may appear single or in various combinations. They may be classified as simple, compound and complex. If only one context in the group of proximate time-sharing contexts is a bottleneck, then it is a simple bottleneck; if there are bottlenecks in two contexts, the group is a compound bottleneck; if bottlenecks are present in all the three contexts, then the group is a complex bottleneck. This scheme of classification is helpful in executing the bottleneck resolving strategy employed by the KBS.

7. Performance Improvement

The performance improvement scenario is provided by the QR-KBS. The rules that make up the knowledge base are Qualitative in nature. They are the representation of the heuristics of the domain experts. The qualitative rules are classified into two categories, namely, local rules and global rules. The local rules control the local tuning of parameters so as to resolve a (local) bottleneck, while the global rules guide the invoking of the local rules. In providing a tuning plan to the designer, the global and the local rules work together to implement a "think globally, but act locally", policy.

7.1 Local Qualitative rules

The local rules concentrate on the resolving of the local bottlenecks one at a time. They are not concerned with the global performance of the system under consideration. They are classified as Type I, Type II and Type III rule groups to facilitate the reasoning mechanism of the KBS.

Type I Qualitative rules

These are the rules (Table 4) that the KBS would refer to when faced with the task of resolving a *simple* bottleneck, i.e., only one context (denoted by C_i) in the

Table 4 Type I qualitative rules.

Rule	SYMPTOM				SOLUTION							
					C_i				C_j		C_k	
	C_i		C_{ijk}		External		Internal		Internal		Internal	
	q_i	ρ_{P1i}	ρ_{P2}	ρ_{P3}	t_{ai}	t_{si}	φ_{P2i}	φ_{P3i}	φ_{P2j}	φ_{P3j}	φ_{P2k}	φ_{P3k}
1	H	H	H	H	↑	↓	↓	↓	↓	↓	↓	↓
2	H	H	H	M	↑	↓	↓	○	↓	○	↓	○
3	H	H	H	L	↑	↓	↓	↑	↓	↑	↓	↑
4	H	H	M	H	↑	↓	○	↓	○	↓	○	↓
5	H	H	M	M	↑	↓	○	○	○	○	○	○
6	H	H	M	L	↑	↓	○	↑	○	↑	○	↑
7	H	H	L	H	↑	↓	↑	↓	↑	↓	↑	↓
8	H	H	L	M	↑	↓	↑	○	↑	○	↑	○
9	H	H	L	L	↑	↓	↑	↑	↑	↑	↑	↑
10	H	M	H	H	↑	↓	↓	↓	↓	↓	↓	↓
11	H	M	H	M	↑	↓	↓	○	↓	○	↓	○
12	H	M	H	L	↑	↓	↓	↑	↓	↑	↓	↑
13	H	M	M	H	↑	↓	○	↓	○	↓	○	↓
14	H	M	M	M	↑	↓	○	○	○	○	○	○
15	H	M	M	L	↑	↓	○	↑	○	↑	○	↑
16	H	M	L	H	↑	↓	↑	↓	↑	↓	↑	↓
17	H	M	L	M	↑	↓	↑	○	↑	○	↑	○
18	H	M	L	L	↑	↓	↑	↑	↑	↑	↑	↑
19	L	M	H	H	○	○	↓	↓	↓	↓	↓	↓
20	L	M	H	M	○	○	↓	○	↓	○	↓	○
21	L	M	H	L	○	○	↓	↑	↓	↑	↓	↑
22	L	M	M	H	○	○	○	↓	○	↓	○	↓
23	L	M	M	M	○	○	○	○	○	○	○	○
24	L	M	M	L	○	○	○	↑	○	↑	○	↑
25	L	M	L	H	○	○	↑	↓	↑	↓	↑	↓
26	L	M	L	M	○	○	↑	○	↑	○	↑	○
27	L	M	L	L	○	○	↑	↑	↑	↑	↑	↑
28	L	L	H	H	↓	↑	↓	↓	↓	↓	↓	↓
29	L	L	H	M	↓	↑	↓	○	↓	○	↓	○
30	L	L	H	L	↓	↑	↓	↑	↓	↑	↓	↑
31	L	L	M	H	↓	↑	○	↓	○	↓	○	↓
32	L	L	M	M	↓	↑	○	○	○	○	○	○
33	L	L	M	L	↓	↑	○	↑	○	↑	○	↑
34	L	L	L	H	↓	↑	↑	↓	↑	↓	↑	↓
35	L	L	L	M	↓	↑	↑	○	↑	○	↑	○
36	L	L	L	L	↓	↑	↑	↑	↑	↑	↑	↑

H:High; M:Medium; L:Low; ↑: Increase parameter; ↓: Decrease parameter; o:parameter is OK

group of three is a bottleneck. The causes of this bottleneck could be internal or external or a combination of both. When there is a delay in the service time of any of the Perspectives, the cause is said to be internal, since it arises due to the inappropriate time-sharing among the three contexts. If the cause is in the flow of the entities (customers) for service or in the total service time of the context, then the cause is considered to be external. The KBS determines whether the cause is internal or external and accordingly prescribes the solution. The "IF" part of the rule is the symptom (of the bottleneck), and the "THEN" part is the solution. For instance, rule 1 in Table 4 should be paraphrased as follows:

IF C_i is a context such that:

q_i is High
AND/OR ρ_{P1_i} is High
AND/OR ρ_{P2} is High
AND/OR ρ_{P3} is High

Table 5 Type II Qualitative rules.

Rule	SYMPTOM					SOLUTION								
	C_i					C_i				C_j				C_k
	C_i		C_j		C_{ijk}	External		Internal		External		Internal		Internal
	q_i	ρ_{P1i}	q_j	ρ_{P1j}	ρ_{P2} / ρ_{P3}	t_{ai}	t_{si}	φ_{P2i}	φ_{P3i}	t_{aj}	t_{sj}	φ_{P2j}	φ_{P3j}	φ_{P2k} / φ_{P3k}
1	H	H	H	H	H	↑	↓	↓	↓	↑	↓	↓	↓	↓
2	H	H	H	H	M	↑	↓	○	○	↑	↓	○	○	○
3	H	H	H	H	L	↑	↓	↑	↑	↑	↓	↑	↑	↑
4	H	H	H	M	H	↑	↓	↓	↓	↑	↓	↓	↓	↓
5	H	H	H	M	M	↑	↓	○	○	↑	↓	○	○	○
6	H	H	H	M	L	↑	↓	○	↑	↑	↓	○	↑	↑
7	H	M	H	M	H	↑	↓	↓	↓	↑	↓	↓	↓	↓
8	H	M	H	M	M	↑	↓	○	○	↑	↓	○	○	○
9	H	M	H	M	L	↑	↓	↑	↑	↑	↓	↑	↑	↑
10	L	L	H	M	H	↓	↑	↓	↓	↑	↓	↓	↓	↓
11	L	L	H	M	M	↓	↑	○	○	↑	↓	○	○	○
12	L	L	H	M	L	↓	↑	↑	↑	↑	↓	↑	↑	↑
13	L	L	L	L	H	↓	↑	○	↓	↓	↑	↓	↓	↓
14	L	L	L	L	M	↓	↑	○	○	↓	↑	○	○	○
15	L	L	L	L	L	↓	↑	↑	↑	↓	↑	↑	↑	↑

H:High; M:Medium; L:Low; ↑: Increase parameter; ↓: Decrease parameter; o:parameter is OK

THEN
increase t_{a_i}
AND/OR decrease t_{s_i}
AND/OR decrease φ_{P2_i} AND/OR φ_{P2_j} AND/OR φ_{P2_k}
AND/OR decrease φ_{P3_i} AND/OR φ_{P3_j} AND/OR φ_{P3_k}

BECAUSE
The cause is external
The cause is external
The cause is internal

Each rule, therefore, is actually a combination of several rules.

In order to resolve the simple bottleneck occurring at context C_i, we may have to change a few parameters of the neighboring contexts C_j and C_k. But, doing so may introduce fresh bottlenecks at these neighboring contexts. Before the local rules are fired to resolve a chosen bottleneck, the KBS invokes the global rules (section 7.2) to ensure that the improvement is carried out without worsening the existing bottlenecks and without introducing new bottlenecks.

Type II Qualitative rules

When two contexts in a group of time-sharing proximate contexts become bottlenecks, we have a compound bottleneck. Type II qualitative rules (Table 5) are invoked by the KBS to solve compound bottlenecks. Here, too, the external causes as well as the internal causes of the bottlenecks are considered. In Table 5, contexts C_i and C_j represent the bottleneck contexts. C_k is the normally operating context. Each rule is a combination of several sub-rules.

Table 6 Type III Qualitative rules.

Rule	SYMPTOM							SOLUTION											
	C_i		C_j		C_k		C_{ijk}	C_i				C_j				C_k			
								External		Internal		External		Internal		External		Internal	
	q_i	ρ_{P1i}	q_j	ρ_{P1j}	q_k	ρ_{P1k}	ρ_{P2} / ρ_{P3}	t_{ai}	t_{si}	φ_{P2i}	φ_{P3i}	t_{aj}	t_{sj}	φ_{P2j}	φ_{P3j}	t_{ak}	t_{sk}	φ_{P2k}	φ_{P3k}
1	H	H	H	H	H	H	H	↑	↓	↓	↓	↑	↓	↓	↓	↑	↓	↓	↓
2	H	H	H	H	H	H	M	↑	↓	↓	○	↑	↓	↓	○	↑	↓	↓	○
3	H	H	H	H	H	H	L	↑	↓	↓	↑	↑	↓	↓	↑	↑	↓	↓	↑
4	H	H	H	H	H	M	H	↑	↓	○	↓	↑	↓	○	↓	↑	↓	○	↓
5	H	H	H	H	H	M	M	↑	↓	○	○	↑	↓	○	○	↑	↓	○	○
6	H	H	H	H	H	M	L	↑	↓	○	↑	↑	↓	○	↑	↑	↓	○	↑
7	H	H	H	M	H	M	H	↑	↓	↑	↓	↑	↓	↑	↓	↑	↓	↑	↓
8	H	H	H	M	H	M	M	↑	↓	↑	○	↑	↓	↑	○	↑	↓	↑	○
9	H	H	H	M	H	M	L	↑	↓	↑	↑	↑	↓	↑	↑	↑	↓	↑	↑
10	H	H	H	M	L	L	H	↑	↓	↓	↓	↑	↓	↓	↓	↑	↓	↓	↓
11	H	H	H	M	L	L	M	↑	↓	↓	○	↑	↓	↓	○	↑	↓	↓	○
12	H	H	H	M	L	L	L	↑	↓	↓	↑	↑	↓	↓	↑	↑	↓	↓	↑
13	H	M	H	M	H	M	H	↑	↓	○	↓	↑	↓	○	↓	↑	↓	○	↓
14	H	M	H	M	H	M	M	↑	↓	○	○	↑	↓	○	○	↑	↓	○	○
15	H	M	H	M	H	M	L	↑	↓	↑	↑	↑	↓	↑	↑	↑	↓	↑	↑
16	L	L	L	L	H	M	H	↓	↑	↓	↓	↓	↑	↓	↓	↑	↓	↓	↓
17	L	L	L	L	H	M	M	↓	↑	○	○	↓	↑	○	○	↑	↓	○	○
18	L	L	L	L	H	M	L	↓	↑	↑	↑	↓	↑	↑	↑	↑	↓	↑	↑
19	L	L	H	M	H	M	H	↓	↑	↓	↓	↑	↓	↓	↓	↑	↓	↓	↓
20	L	L	H	M	H	M	M	↓	↑	○	○	↑	↓	○	○	↑	↓	○	○
21	L	L	H	M	H	M	L	↓	↑	↑	↑	↑	↓	↑	↑	↑	↓	↑	↑
22	L	L	L	L	L	L	H	↓	↑	↓	↓	↓	↑	↓	↓	↓	↑	↓	↓
23	L	L	L	L	L	L	M	↓	↑	○	○	↓	↑	○	○	↓	↑	○	○
24	L	L	L	L	L	L	L	↓	↑	↑	↑	↓	↑	↑	↑	↓	↑	↑	↑

H:High; M:Medium; L:Low; ↑: Increase parameter; ↓: Decrease parameter; o:parameter is OK

Type III Qualitative rules

The KBS makes use of these rules in resolving complex bottlenecks. The rules (Table 6) outline the course of action to be taken when all the three contexts in the time-sharing group become bottlenecks. In most cases of compound bottlenecks, the cause is internal – the inappropriate allocation of Perspectives to the time-sharing contexts.

7.2 Global Qualitative rules

The local rules listed in the above section cannot be applied arbitrarily. There are practical conditions that restrict the application of these rules. Often it is not possible to resolve a context bottleneck, because the nature of the system is such that the context parameters that need be changed to resolve the bottleneck cannot be changed. In such cases, the KBS has to try out other alternatives, seeking relevant information from the user. In resolving the bottlenecks, more important than the above *feasibility* condition, is the *advisability* condition. At times, it may be *feasible* to change a parameter, but the KBS may judge that it is not *advisable* to change it for fear of exerting bad influence on the other contexts. If the changes made in the feasible parameters to resolve a bottleneck will end up in creating fresh bottlenecks or worsen the existing ones somewhere else in the MCM network, then the *advisability* condition will prompt the KBS to issue a warning to the user. Advisability means coming up with a sound strategy for resolving bottlenecks such that changes

Table 7 Global Rules.

RULE	IF			THEN	
#	Feasibility	Constraints	Group/downstream context bottleneck state change	Invoke local rule(s)	Advisability Message
1	o	×	--	×	Not feasible
2	×	o	--	×	Not feasible
3	×	×	--	×	Not feasible
4	o	o	No change	o	Highly advisable
5	o	o	New simple	o	Somewhat advisable
6	o	o	New compound/complex	o	Not at all advisable
7	o	o	Simple to compound	o	Somewhat advisable
8	o	o	Simple to complex	o	Not at all advisable
9	o	o	Compound to complex	o	Not advisable
10	o	o	Complex to saturated	o	Not advisable
11	o	o	Compound to simple	o	Advisable
12	o	o	Complex to compound	o	Advisable

o: Yes; ×: No; -- : No change; ×× : Improvement not possible

made in the network are minimum and consequently undesirable influences on junctions and contexts operating in normal conditions are kept to the minimum. The *advisability* condition works at the following two levels:

1. The resolving of a particular bottleneck may not worsen the existing bottlenecks at the other contexts in the proximate group or introduce new bottlenecks in the proximate group.
2. The resolving of the said bottleneck may not worsen the existing bottlenecks at the other contexts outside the proximate group or introduce new bottlenecks outside the proximate group.

The global rules listed in Table 7 take care of the feasible and advisable conditions when resolving a given bottleneck. The global rules state that if the feasibility conditions are met (i.e., if it is possible to change the values of the selected parameters to resolve the given bottleneck) and if the constraints are kept (i.e., the changes are made within the prescribed limits) then the local rules for resolving may be invoked after issuing the appropriate advisability to the user. Rule 4, for instance, states that if the feasibility and constraint requirements are met and if there will be no bad influence on any other context in the network due to the proposed improvement, then the local rules for improvement may be invoked and a 'highly advisable' message be displayed to the user. Rule 8, on the other hand, states that if the feasibility and constraints requirements are met, but an existing simple bottleneck will worsen to a complex bottleneck due to the effect of the proposed improvement, then a 'not at all advisable' message be issued to the user, but the local rules for improvement may still be invoked if the user says 'OK'. This is because the user may want to improve a particular bottleneck even if there is going to be a bad influence somewhere else in the network. The global rules implements the 'think globally, but act locally,' tuning policy.

8. Example of performance design of MCM with the composite service model

In the preceding sections, we defined the composite service model and showed how to use it to design the Perspective allocation in collaborative systems. In this section, we apply the composite service theory to work out the Perspective allocation and performance design of a practical example. This example is the floor of a medium-sized computer store in the city of Tokyo. The MCM description of the system, the GPSS simulation of the system operation and the performance design of the system by the KBS are discussed in the following sub-sections.

8.1 Requirements Analysis

The owner of the store plans to open a new floor to sell computer hardware and software. The floor space can accommodate a total of twelve large shelves. In addition, there are four cash counters, two counters for issuing parking vouchers and two small enquiry counters. A rough estimate of the customer arrivals and their shopping preferences is available. The aim is to design the performance of the system so that its operation is efficient, while maximizing customer satisfaction and, at the same time, minimizing operational cost (i.e. cost of hiring personnel).

To achieve the above threefold objectives, we do the following:

1. Arrange the layout of the floor in such as way that it closely matches the customer preferences. A subtle observation that will help our design is that the majority of the customers will not buy major hardware and major software at the same time, because it is expensive. Grouping the hardware and the software shelves separately (Fig. 13) will save the customers a lot of crisscrossing the shop floor.
2. Group the services such that they are proximate in function and/or distance so that Perspective allocation can be effectively applied. The purpose of this Perspective Allocation is to cut down the costs in employing the sales staff.
3. Simulate the operation of the system with the available "rough estimates" and seek the advice of the KBS in improving the operation of the system.

8.2 Layout Design and Perspective Allocation

Fig. 13 shows the MCM of the layout of the shop floor. There are four cash payment counters, two counters for issuing handwritten receipts and parking vouchers, two enquiry counters, six hardware shelves and six software shelves. Each of these is a "context" in the MCM terminology. Each context has a primary Perspective P1 assigned to it so that he/she works as a dedicated server at that context. In addition, auxiliary Perspectives P2 and P3 are also allocated. A group of two or three proximate contexts doing auxiliary time-sharing is indicated by drawing arcs on the top right hand corner of the individual contexts. For example, the contexts, "Accounts1" (ACN1), "Accounts2" (ACN2), and "Parking vouchers1" (VCH1) form a group

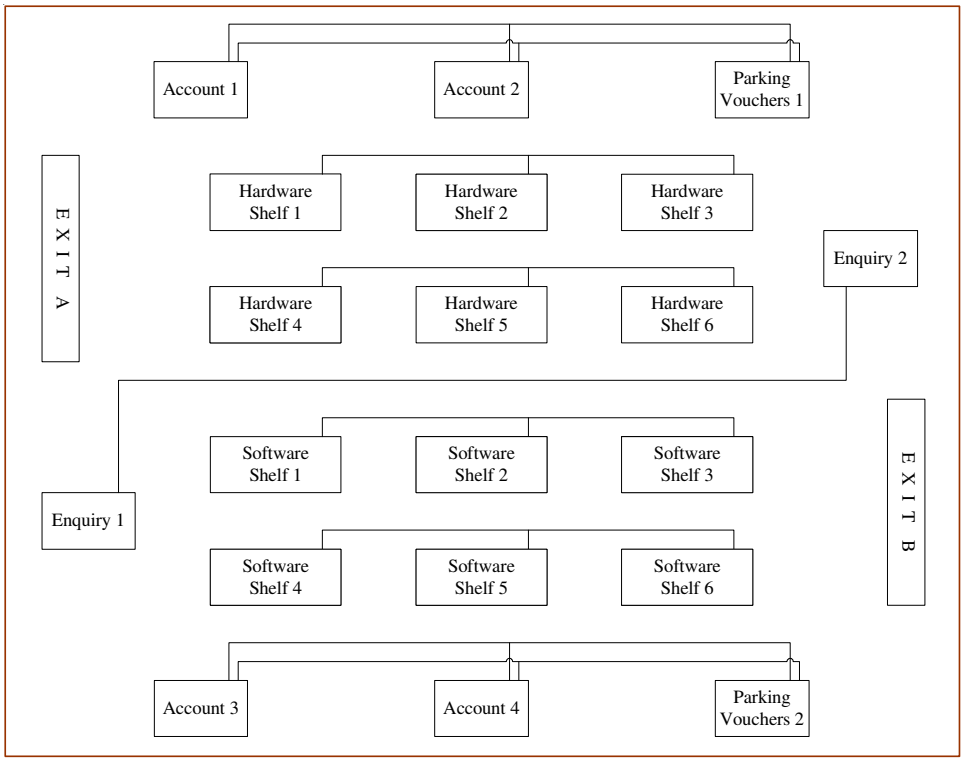


Fig. 13 Computer store example of Perspective Allocation.

of time-sharing proximate contexts. The primary Perspective P1 at ACN1 and ACN2 is the cashier who handles the payments and does not move from context to context. He/she is the principal Perspective. P2 is the goods inspector, who receives the goods from the customers, inspects them before packing and reads the barcode price tags; P3 is the good wrapper, whose job is to wrap the goods. There is no principal Perspective in the VCH1 context. "Accounts3" (ACN3), "Accounts4" (ACN4), and "Parking vouchers2" (VCH2) do a similar time-sharing. Arcs representing the flow of customers are omitted for the sake of visual clarity in the diagram.

As a rule of thumb, P1 is the staff member who, on an average, works at a given context for the longest period of time per customer and *does not move to other contexts*. In other words, he/she provides *dedicated* service. P2 works for a relatively shorter time on an average at a given context and *moves* from context to context providing service. He/she offers a *distributed* type of service. P3's role is similar to that of P2, except that P3's average time is shorter than that of P2.

Table 7 Initial Simulation Scenario.

CONTEXT GROUP	ta	Ci			Cj			Ck			□2i	□2j	□2k	□3i	□3j	□3k	Qi	Qj	Qk	□P1i	□P1j	□P1k	□P2	□P3
		tP1i	tP2i	tP3i	tP1j	tP2j	tP3j	tP1k	tP2k	tP3k														
ACN1 ACN2 VCH1		10	10	9	11	11	10	9	8	7	1.46	3.11	1.04	1.43	1.08	0.57	0.00	0.03	0.01	0.08	0.14	0.12	0.24	0.22
HDW1 HDW2 HDW3		14	13	12	13	12	11	11	10	9	21.53	19.75	22.83	20.39	18.33	21.47	23.83	21.73	25.74	0.99	1.00	0.96	1.00	1.00
HDW4 HDW5 HDW6	8	13	13	11	12	9	8	12	11	7	16.91	16.09	17.85	11.55	10.86	12.59	11.38	17.79	15.78	0.93	0.99	0.95	0.99	0.87
ENQ1 ENQ2	6	8	6	4	5	4	3	-	-	-	0.35	0.31	-	0.00	0.07	-	0.01	0.00	-	0.12	0.09	-	0.16	0.11
SFW1 SFW2 SFW3		11	10	3	12	9	8	14	12	11	10.02	9.78	12.44	0.54	1.27	2.07	1.17	0.84	1.46	0.63	0.68	0.61	0.89	0.62
SFW4 SFW5 SFW6		15	14	8	13	11	10	13	13	10	24.72	26.84	26.83	4.94	5.62	5.27	16.52	5.20	9.14	0.98	0.93	0.93	0.98	0.82
ACN3 ACN4 VCH2		11	10	9	10	10	9	11	11	9	7.48	9.07	3.63	4.19	4.00	2.28	0.04	0.11	0.02	0.16	0.21	0.17	0.31	0.28

The hardware shelves (HDW) and the software shelves (SFW) are arranged in rows of three. Each row of shelves forms a group of time-sharing contexts. In these groups, however, the auxiliary Perspective P3 is absent (P2 is indicated by a single arc). P1 is the main Perspective explaining to the customers the hardware and the software features and responding to their queries. P2 is the auxiliary Perspective moving from shelf to shelf in the time-sharing group and guiding the customers to "what is where". Enquiry1 and Enquiry2 also have only P1 and P2 Perspectives. The Enquiry contexts are proximate in function, although they are not proximate in distance.

8.3 System performance evaluation & performance design

We simulate one particular scenario of the computer shop operation, based on the historical data obtained from similar shops in the vicinity. In this typical scenario, the majority of the customers visit only the software or only the hardware area; only a small percentage of the customers visit both the areas. Roughly 12% of the customers visit 85% of the shelves; 46% visit only the hardware (HDF) shelves. Of this, 87% visit over four shelves. The remaining 42% visit only the software (SFW) shelves. The operational parameters randomly selected for simulation and the designed parameters are given in Table 7.

Table 8 shows the simulation data obtained from the initial simulation with randomly selected operation parameters. The delays in service time due to Perspective allocation are large. This leads to a very high utilization in most of the HDW and SFW contexts. The bottlenecks in these contexts stop the flow of customers to the accounts contexts to such an extent that the utilization of the latter contexts is very low.

The KBS attempts to resolve the bottlenecks in the system operation. Most of the bottlenecks are due to an inappropriate allocation of the auxiliary Perspectives. The cause of these bottlenecks is mainly "interior", as evident from the long delays. The KBS advises the designer to reduce the values of the simulation parameters of the auxiliary as well as primary Perspectives. The result is the intermediate simulation data shown in Table 9. The delays and consequently the utilizations of the Perspectives are somewhat lowered; however, this is not an acceptable solution because the bottlenecks are persistent. The KBS gives a further try. This time the bottlenecks are completely resolved as shown in Table 10.

Table 8 Intermediate Simulation Scenario.

CONTEXT GROUP	ta	Ci			Cj			Ck			$\delta 2^i$	$\delta 2^j$	$\delta 2^k$	$\delta 3^i$	$\delta 3^j$	$\delta 3^k$	Q_i	Q_j	Q_k	$\delta P1^i$	$\delta P1^j$	$\delta P1^k$	$\delta P2$	$\delta P3$
		tP1i	tP2i	tP3i	tP1j	tP2j	tP3j	tP1k	tP2k	tP3k														
ACN1 ACN2 VCH1		10	9	9	11	10	8	9	8	7	6.79	4.51	4.15	3.08	1.47	3.30	0.05	0.03	0.03	0.13	0.14	0.16	0.28	0.25
HDW1 HDW2 HDW3		11	10	9	13	11	11	10	10	9	22.07	20.07	23.26	20.35	18.12	21.14	26.43	21.36	27.31	0.99	0.99	0.97	1.00	1.00
HDW4 HDW5 HDW6	10	13	10	9	12	9	8	12	9	7	16.47	16.64	16.93	10.43	11.13	11.29	3.62	15.25	14.99	0.88	0.99	0.98	0.99	0.87
ENQ1 ENQ2	9	8	6	4	5	4	3	-	-	-	0.29	0.65	-	0.00	0.37	-	0.00	0.00	-	0.09	0.06	-	0.12	0.09
SFW1 SFW2 SFW3		11	9	3	12	9	8	11	10	8	5.26	5.29	6.92	0.54	0.93	1.02	0.13	0.40	0.27	0.42	0.56	0.52	0.82	0.60
SFW4 SFW5 SFW6		15	14	8	13	11	10	13	13	7	24.94	27.21	27.20	4.87	6.78	5.27	14.85	5.87	10.86	0.97	0.93	0.94	0.98	0.76
ACN3 ACN4 VCH2		11	8	8	10	8	8	11	9	7	3.88	7.18	5.85	2.08	3.86	5.03	0.01	0.06	0.05	0.16	0.14	0.17	0.28	0.25

Table 9 Final Simulation Scenario.

CONTEXT GROUP	ta	Ci			Cj			Ck			$\delta 2^i$	$\delta 2^j$	$\delta 2^k$	$\delta 3^i$	$\delta 3^j$	$\delta 3^k$	Q_i	Q_j	Q_k	$\delta P1^i$	$\delta P1^j$	$\delta P1^k$	$\delta P2$	$\delta P3$
		tP1i	tP2i	tP3i	tP1j	tP2j	tP3j	tP1k	tP2k	tP3k														
ACN1 ACN2 VCH1		10	5	5	11	6	5	9	5	4	4.47	4.91	3.74	1.98	1.10	1.47	0.65	0.17	0.65	0.53	0.42	0.55	0.63	0.56
HDW1 HDW2 HDW3		8	2	2	8	2	2	8	2	2	0.02	0.02	0.04	0.03	0.09	0.06	1.34	0.90	1.29	0.81	0.77	0.77	0.80	0.80
HDW4 HDW5 HDW6	12	10	3	3	8	3	3	9	3	2	0.71	2.11	1.65	0.25	0.68	0.45	0.98	9.03	8.71	0.83	0.92	0.94	0.94	0.91
ENQ1 ENQ2	11	12	9	8	11	9	9	-	-	-	3.11	2.10	-	1.90	0.99	-	0.01	0.01	-	0.32	0.31	-	0.45	0.43
SFW1 SFW2 SFW3		12	3	3	11	5	5	10	2	2	0.11	0.29	0.67	0.39	0.39	0.84	1.98	1.26	1.79	0.81	0.85	0.82	0.84	0.85
SFW4 SFW5 SFW6		11	4	4	11	3	3	12	3	3	0.16	0.14	0.04	0.43	0.41	0.09	6.43	0.79	4.28	0.88	0.72	0.89	0.87	0.85
ACN3 ACN4 VCH2		11	7	5	10	5	4	11	6	6	7.46	5.70	4.80	4.45	3.40	3.07	0.31	0.44	0.62	0.40	0.54	0.59	0.64	0.58

We have, however, relaxed the theoretical bottleneck landmark criteria discussed in section 6. In this scenario, the utilization of the individual contexts can have a tolerance of up to 0.94, since the customer arrival, as seen from the historical data does not peak erratically. There is no danger, therefore, that the arrivals will overwhelm the servers. Besides, the auxiliary servers being internal can have relatively higher utilizations, since they are somewhat insensitive to the external fluctuations.

The resolving of bottlenecks in the system improves the efficiency in the system operation. This reduces the waiting time as well as the service time per customer, leading to customer satisfaction. Finally, due to the appropriate Perspective Allocation advised by the KBS, the cost of hiring sales persons is also kept to a minimum.

9. Conclusion

Single and parallel server structures of the queuing theory cannot model a wide variety of services that are found in collaborative systems. In particular, being dedicated servers, they cannot represent distributed and composite types of services. We presented a new composite service model that can represent a wide variety of services. We made use of the composite server model to address the problem of Perspective allocation in MCM. We proposed a Qualitative approach in the diagnosis and resolving of bottlenecks that result from inappropriate Perspective allocation. We applied our method in the performance design of a practical collaborative system. By the appropriate server allocations suggested by the KBS, we managed to design the system so that its performance is efficient, while keeping the cost of operation low.

The qualitative reasoning approach may not always yield a numerically precise solution to the problem. However, it provides a set of feasible solutions under the system operation constraints. Any of these solutions are sufficient to project the performance estimate earlier on in the requirement analysis of the system development life cycle. As an extension to this study, we plan to fine-tune the performance design by introducing other simulation optimization techniques.

10. References

[1] Shelly, G. B., Cashman, T. J., and Rosenblatt, H. J., *System Analysis & Design* (5th ed.). Thomson Course Technology, Boston, 2003.

[2] Cooling, J., *Software Engineering for Real-time Systems*. Addison-Wesley, 2003.

[3] Gonsalves, T., Itoh K., Kawabata R., "Performance Design and Improvement of Collaborative Engineering Systems by the Application of Knowledge-Based Qualitative Reasoning", *Knowledge Based Design Series*, *The ATLAS*, Vol.1, pp. 1-31, 2005. (www.theatlasnet.org/?q=node/58)

[4] Hasegawa, A., Kumagai, S., Itoh K., "Collaboration Task Analysis by Identifying Multi-Context and Collaborative Linkage", *CERA*, Vol. 8, No. 1, pp. 61-71, 2000.

[5] Banks, J., Carson, II, J. S., Nelson, B. L., and Nicol, D. M., *Discrete event System Simulation* (3rd ed.). Prentice-Hall, New Jersey, 2001.

[6] Schriber, T. J., *An Introduction to Simulation Using Gpss/H*. John Wiley & Sons, New York, 1991.

[7] •@Bobillier, P. A., Kahan B., C., and Probst, A. R., *Simulation with GPSS and GPSS V*. Prentice-Hall, Inc., New Jersey, 1976.

[8] Minuteman Software, *GPSS World*, NC, 2001.

[9] Kuipers, J., *Qualitative Reasoning Modeling and Simulation with Incomplete Knowledge*. The MIT Press, Cambridge, Massachussetts, 1994.

[10] Farley, A. M., and Lin, K. P., "Qualitative reasoning in economics", *Journal of Economic Dynamics and Control*, Vol. 14, pp. 465-490, May, 1990.

[11] de Kleer, J., and Forbus, K. D., *Building problem solvers*, MIT Press, Cambridge, Massachusetts, 1993.

[12] Davies R., "Qualitative Reasoning about Physical Systems", *Qualitative Reasoning about Physical Systems*, Bobrow D. G., ed., The MIT Press, Cambridge, Massachusetts, pp. 347-410, 1985.

[13] Rajagopalan R., "Qualitative modeling in the turbojet engine domain", Proc. AAAI, Austin, pp. 283-287, 1984.

[14] Gonsalves, T., and Itoh, K., "Generic Core Life Cycle and Conceptual Architecture for the Development of Collaborative Systems", *Knowledge Sharing in the Integrated Enterprise: Interoperability Strategies for the Enterprise Architect*, Bernus, P., and Fox, M., eds., Vol. 183/2005, pp. 417-426, 2006.

[15] Klienrock, L., *Queuing systems*. John Wiley & Sons, Inc., 1975.

[16] Whitehouse, G. E., and Wechsler, B. L. *Applied Operations Research: A Survey.* John Wiley & Sons, New York, 1976.

[17] Gonsalves, T., Kawabata, R., Tabata, S., and Itoh, K., "Petri net Tools for the Analysis of Collaborative Tasks", *SDPS Journal*, submitted for publication.

[18] Law, A. M., and Kelton, W. D., *Simulation Modeling and Analysis* (2nd ed.). McGrawHill, New York, 1991.

[19] Jain, R., *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley-Interscience, New York, 1991.

[20] Itoh K., Honiden, S., Sawamura, J., and Shida, K., "A Method for Diagnosis and Improvement on Bottleneck of Queuing Network by Qualitative and Quantitative Reasoning", *Journal of Artificial Intelligence* (Japanese), Vol. 5, No. 1, pp.92-105, 1990.

[21] Karian, Z. A., and Dudewicz, E. J., *Modern Statistical, Systems, and GPSS Simulation* (2nd ed.). CRC Press, New York, 1999.

Appendix A

Notations

C_i, C_j, C_k = Proximate contexts sharing auxiliary service times.

ta_i = average inter-arrival time at context C_i

ta_j = average inter-arrival time at context C_j

ta_k = average inter- arrival time at context C_k

ts_i = average service time at context C_i

ts_j = average service time at context C_j

ts_k = average service time at context C_k

φ_{P1_i} = P1's fractional service time at C_i

φ_{P1_j} = P1's fractional service time at C_j

φ_{P1_k} = P1's fractional service time at C_k

φ_{P2_i} = P2's fractional service time at C_i

φ_{P2_j} = P2's fractional service time at C_j

φ_{P2_k} = P2's fractional service time at C_k

φ_{P3_i} = P3's fractional service time at C_i

φ_{P3_j} = P3's fractional service time at C_j

φ_{P3_k} = P3's fractional service time at C_k

t_{P1_i} = P1's service time at context C_i

t_{P1_j} = P1's service time at context C_j

t_{P1_k} = P1's service time at context C_k

t_{P2_i} = P2's service time at context C_i

t_{P2_j} = P2's service time at context C_j

t_{P2_k} = P2's service time at context C_k

t_{P3_i} = P3's service time at context C_i

t_{P3_j} = P3's service time at context C_j

t_{P3_k} = P3's service time at context C_k

δ_{2_i} = delay introduced by the unavailability of P2 for service at C_i

δ_{2_j} = delay introduced by the unavailability of P2 for service at C_j

δ_{2_k} = delay introduced by the unavailability of P2 for service at C_k

δ_{3_i} = delay introduced by the unavailability of P3 for service at C_i

δ_{3_j} = delay introduced by the unavailability of P3 for service at C_j

δ_{3_k} = delay introduced by the unavailability of P3 for service at C_k

δ_i = effective delay introduced at C_i

δ_j = effective delay introduced at C_j

δ_k = effective delay introduced at C_k

ρ_{C_i} = utilization of C_i

ρ_{C_j} = utilization of C_j

ρ_{C_k} = utilization of C_k

ρ_{P2} = utilization of P2

ρ_{P3} = utilization of P3

ρ_{P1_i} = utilization of P1 at C_i (= ρ_{C_i})

ρ_{P1_j} = utilization of P1 at C_j (= ρ_{C_j})

ρ_{P1_k} = utilization of P1 at C_k (= ρ_{C_k})

ρ_{P2_i} = utilization of P2 at C_i

ρ_{P2_j} = utilization of P2 at C_j

ρ_{P2_k} = utilization of P2 at C_k

ρ_{P3_i} = utilization of P3 at C_i

ρ_{P3_j} = utilization of P3 at C_j

ρ_{P3_k} = utilization of P3 at C_k

ρ_{P2} = overall utilization of P2 (at C_{ijk})

ρ_{P3} = overall utilization of P3 (at C_{ijk})

Appendix B
Estimation of confidence intervals

The validation of the composite service model by comparing the simulation results with the theoretical results is presented in section 5. Each simulation run simulates 10 hours (600 minutes) of work. "How long to simulate?" is a difficult question in simulation analysis. According to Dudewitch and Kerien [21], the number of periods, n, for which the simulation should be carried out is given by:

$$n = \left\lceil \left(\frac{\sigma}{d} \Phi^{-1} \left(\frac{1+P^*}{2} \right) \right)^2 \right\rceil$$

where, P^* is the confidence interval,

σ is the variance,

d is the width,

and Φ^{-1} is the inverse normal distribution

$$\Phi(z) = \int_{\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$$

and $\Phi^{-1}(y)=(z \text{ such that } \Phi(z)=y)$

Exponential Distribution

$$f(x) = \begin{cases} \frac{1}{\beta} e^{\frac{-(x-\lambda)}{\beta}} & (x \geq \lambda) \\ 0 & (otherwise) \end{cases}$$

where, λ = locate

and β = scale

variance = β^2

We have chosen $\beta = 1$. For a 95% confidence interval with width 0.5,

$$n = \left\lceil \left(\frac{1}{0.05} \Phi^{-1} \left(\frac{1+0.5}{2} \right) \right)^2 \right\rceil$$
$$= \left\lceil \frac{1}{(0.05)^2} (1.96)^2 \right\rceil$$
$$= 1536.64$$
$$\cong 2,000$$

Discrete Uniform Distribution

$$Variance = \frac{(Max - Min + 1)^2}{12}$$

Since, we have chosen the offset values for each y the service times as ± 10, the variance is

$$\sigma^2 = \frac{\left[\left(\bar{x} + 10 \right) - \left(\bar{x} - 10 \right) + 1 \right]^2}{12} = 36.75$$
$$n = \left\lceil \frac{36.75}{(0.05)^2} (1.96)^2 \right\rceil$$
$$= 56471.62$$
$$\cong 60,000$$