Using System Dynamics (SD) simulation to explore and derive self-adaptive techniques for information systems

Derek J. Edwards, Alan D. Pengelly

British Telecommunications plc

BT Labs,
Martlesham Heath,
Ipswich,
Suffolk,
IP5 7RE
United Kingdom

ABSTRACT

Rapidly changing customer needs require greater flexibility from the service provider to maintain or improve quality of service. This is particularly true of the telecommunications industry which is highly competitive and subject to rapid evolution.

To meet the desired flexibility, a company's information systems must be able to adapt quickly to the environment. As manual reconfiguration is becoming increasingly ineffective, a solution is to endow the Information System with the ability to autonomously adapt to change.

Information Systems architecture is essentially driven by the dynamics of market trends and company structure. Simulation techniques, such as SD, that mirror associated causes and effects give essential insight into the critical interactions between the Information System and the environment.

This paper discusses how SD has been used to simulate an adaptive information system and its environment, and ways in which the models can be analysed to derive the principles and criteria for adaption.

Introduction

The management and control of today's network and operational support systems presents a significant challenge to Telecommunications companies. These systems generally have very complex architectures and a high degree of heterogeneity. They are typically geographically distributed and incorporate large databases, such that managing and operating these systems accounts for a large proportion of a company's running costs. Many of the maintenance and support activities involve manually based processes which, as well as being resource intensive, also provide a source of inefficiency and poor quality of service (QoS). Automation of these activities would provide the dual benefits of reduced costs and improved QoS. The question is how?

Over recent years there has been a growth of interest in the area of adaptive and evolving systems (Morecroft & Sterman, 1994). A number of leading academic institutions and companies are active in this field. In addition, there is much interest within the Systems Dynamics (SD) community as regards how SD can be used to model such systems.

The work described in this paper has used SD to model a distributed database system which can adapt itself on the basis of usage and performance criteria so that it is always optimised with respect to its environment. Results from this work will be used to derive novel adaption algorithms* and techniques.

Context

The model comprises three main parts - the information system (fig. 2b), the customer/market influence on the system (fig. 2a), and the effect of the system on the customer/market (upper fig. 2c). These form a feedback loop which models the evolution of the system.

The Distributed Information System

The Information system under study can be considered as either being used directly by customers (e.g. automated service information) or used within a business process by people who service customers (e.g. help desk or fault reporting). In either case, variants of the model can be implemented to suit the situation. The model represented here relates to the latter example which can be used as the general case.

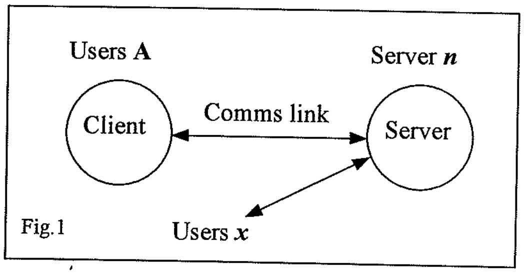

Fig.1

At the time of writing, the model is in the early stages of development and will evolve as work continues. The model shows a simplified Server node and Client of a distributed database system where different nodes would be substituted according to the experimentation and evaluation required. Let us assume that we are interested in the information response time (performance) for a particular group of users 'Users A' (fig. 1). These are connected via a communications link to 'Server *n*'. Also using Server *n* are Users *x* (via different links). Users A and Users *x* will influence the performance seen by Users A as they are using the same server.

Within the SD model (fig. 2b), communications performance (return of data) is represented by the upper path, and server performance (query response time) by the lower path. The separation of the two paths in this way may be thought unusual. We might normally expect to model more literally, i.e. data return to follow on from the query process. However, as the model normalises different query types in order to determine server response times, it would be difficult to separate these out in order to determine the data return pattern for specific users. Instead, since each query type is known at initiation, it is far easier to deal with the communication response separately. This will lead to model sample time lag error, but over the intervals we are interested in (usually several hours), response time samples (at most a few minutes) will average out the affect of these lags. Normally, the overall response time would be the combined outgoing query processing time plus the return data time, hence the model 'sums' these. This form of model allows the individual effects of server and communication link to be easily seen, rather than a server 'backlog' being caused by a slow communication link. It might also be considered a reasonable representation of return data being cached, freeing the server. If required, feedback could be implemented between the communication link and the server flow.

An adaption algorithm is run if system loading causes performance ('Quality of Service') to be reduced beyond a defined threshold for a significant period, in order to determine a possible improved alternative Client / Server connectivity and data distribution. Each suggested Client to Server connection can then be investigated with the model, firstly to validate the proposed configuration, and secondly to improve the algorithm by seeing where anomalies might occur. Hence initially (fig.1), Users A may be connected to Server 1, which also services Users C. The adaption algorithm may suggest that a better configuration is to connect Users A instead to Server 2, which will also service Users D. The different server performance (Server 2) and user loading (Users D) parameters would then be substituted. The intention would be to adjust the system to the new configuration as quickly as desired according to specified criteria - perhaps ready for the same load the next day, or in an emergency within a few minutes.

Customer / Market Influence

The model allows us to shape probable influences on the use of the Information System by taking into account known trends, causes and effects or even simulating less predictable scenarios. The changes in usage may be, for example, simply an altered work-pattern with regard to a particular group of users, fewer or more users, or a major reorganisation. Any of these may be short or long term and will influence the demands on the system which may require a reconfiguration. Within a large organisation, this is typically an upgrade or retune performed locally, rather than a rebalance of the utilisation of the system as a whole across the company. Also, manual correction is often only implemented when the situation has seriously deteriorated. Automated methods allow for timely adjustment and possible fine-tuning maintaining a system as close to optimal performance as possible.

The model attempts to represent these subtle and not so subtle changes and the likely time periods and effects involved. As well as these being input directly to the Information System model, the results can be used to provide usage pattern input scenarios to the adaption algorithm for development and test purposes.

Effect of the System on the Customer / Market

There have been a number of papers and articles referring to the effects of competition on service demand (similar principles to HPS, 1994). This model assumes that the primary goal is to maintain a high perceived quality of service to the customer (compared with competitors) and that this will maintain or generate new demand which in turn will influence the use of systems that support that demand. If the demand cannot be met, quality of service will be perceived to fall. Various factors can influence what we call quality. However, we are assuming that provision of timely and accurate information, together with a sensible price are the key factors. If we can't find the correct information or can't access or provide it quickly enough, then this will affect customer perception. If the service is too expensive it will generally not matter how 'good' it is. Other factors can be plugged-in as desired.

This aspect of the model is based on the assumption that if the response to user required data is too slow, then we will not respond quickly enough to the customer need and hence customer perception will be poor, causing a potential migration away from this service. The effect of this (eventually) is to modify our work-practice (as discussed earlier) requiring a system adaption. Ideally we need early sight of performance problems and rapid rectification in order to minimise the ripple effect through to the customer. In this way, the system will tune itself to the day-to-day variable demands, as well as to the more gradual changes, insulating the customer from the less-desirable effects, as far as is possible.

Price is influenced by the cost of provision of the service. An attempt is made to reflect the difference in the effect of timely and relatively cheap automated adaption methods compared with a slower acting manual approach. Hence both the timeliness and cost of the service are affected by the method used for adaption. Accuracy is affected by the fact that there is a tendency to use incomplete information or to be unaware of the latest details if long delays are incurred in trying to obtain the data. These influences are therefore used as the key quality factors used within the model.

The longer it takes to improve the system, the more likely will customer perception be affected and correspondingly the way the company responds. If the delay is relatively long, the company is more likely to be biased towards a reactive, 'fire-fighting' approach. This can lead to unpredictable demands on the information system which may cause undesirable affects (perhaps quality oscillations). On the other hand, if fine-adjustment improvements can be made in relatively short timescales, then the company can take a more measured strategic approach to change, leading to more predictable information system demands. The aim is to create a closed-loop model that covers the above influences in adequate detail and is sufficiently rigorous to validate the approach, methods and results of adaption.

Conclusions

Work to date has shown that the model can be useful in evaluating the criteria and response time for adaption. Trigger conditions for adaption have in the past been based on the exceeding of service-time thresholds, however experiments suggest that a more 'damped' view of conditions may on occasions be more appropriate depending on the tolerance to fluctuations in system performance. Also, the effect on the end-customer is likely to follow a predictable time-lag. Using these modelling techniques helps with identifying the performance measures that are required, enabling performance monitors to be located appropriately on the actual system for use by the adaption process. Figures derived from these experiments can help to determine the optimal scheduling for adaption and the speed at which this should be undertaken. Development of the algorithm will be based on the SD model experiments and the results applied to the model for validation and improvement.

* **The Adaption Algorithm**. By using existing or derived performance data, non-linear calculations are carried out to determine the most effective Client - Server connectivity to improve overall system and/or individual user-group perceived performance. This will generally result in some users being connected to alternative servers with the need for corresponding data movement or replication. The currently implemented algorithm uses analytical limited combinatorial test calculations on possible connections and data allocation to arrive at a near optimal configuration. If there is little latitude in the system, a direct improvement may not be possible. The results are generally a compromise between calculation time and accuracy. Methods to reduce calculation time and maintain sufficient accuracy are paramount. Other, more esoteric techniques are also being investigated.

References

Elizabeth Oxborrow, (1989). Databases and Database Systems : 2nd Edition. Chartwell-Bratt, ISBN 0-86238-237-8

HPS (1994). Introduction to Systems Thinking and Ithink. High Performance Systems, Inc., Hanover, NH. p 103.

Morecroft, J. and Sterman J. (1994) - Modeling for Learning Organisations. Productivity Press.

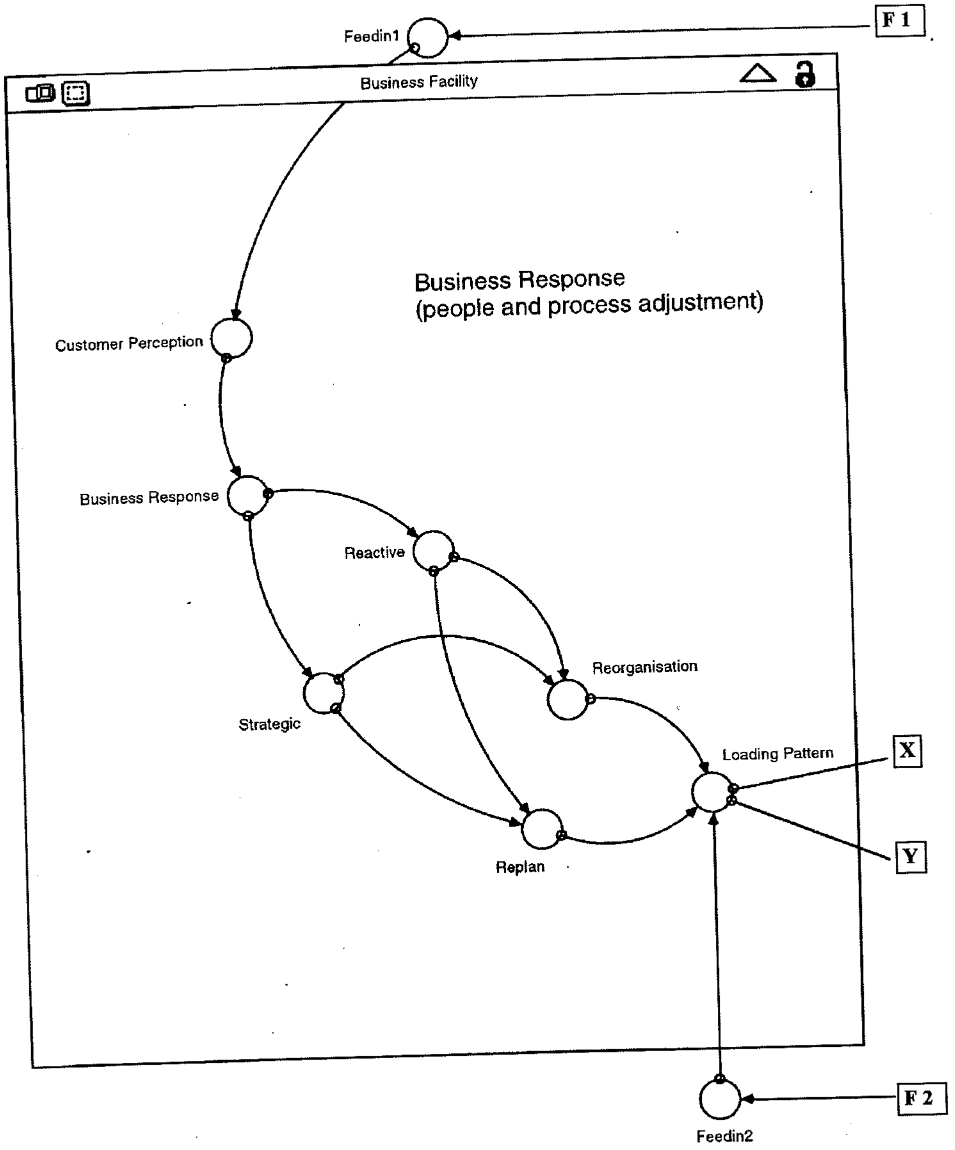

Fig. 2a. This section of the model represents the business response to the need for change and activities which are likely to influence the scheduling and severity of demands on the information system.

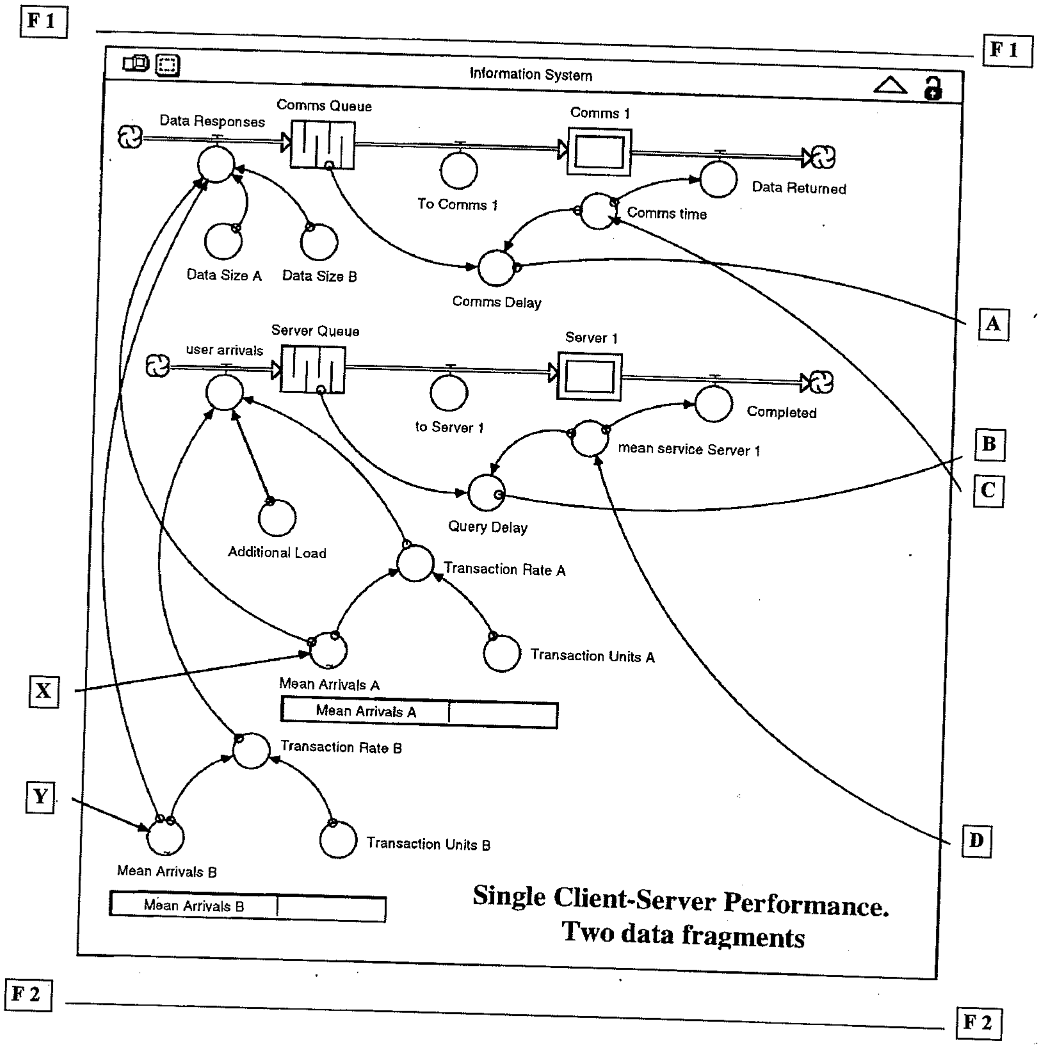

Fig. 2b. This section of the model represents the communication and server performance effects based on Client activity. Mean Arrivals A and B represent differing queries and rates for differing data fragment sizes.

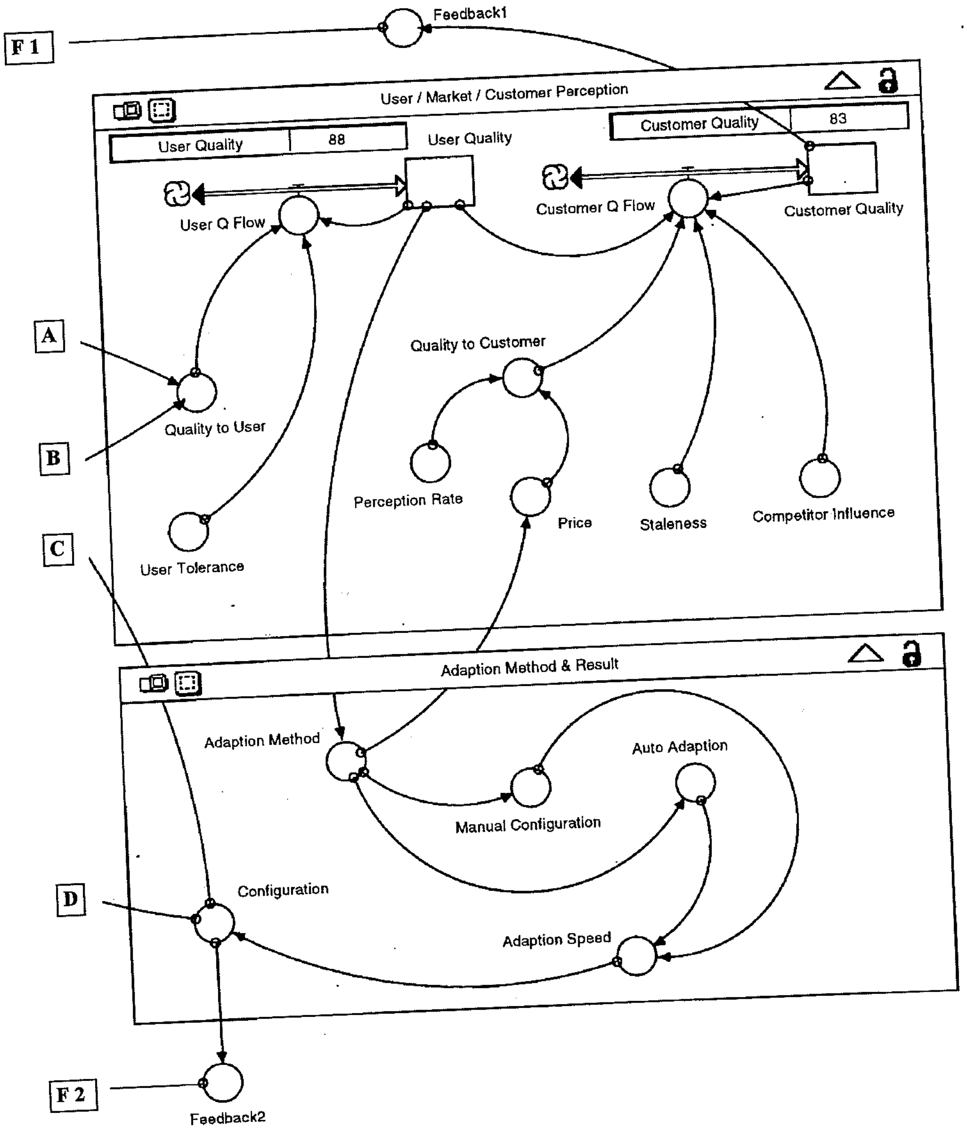

Fig. 2c. This section of the model represents the effects of Information System performance on quality as perceived by users and customers. Customers are also shown influenced by external factors. Changes in quality influence the need for adaption and its timeliness, which in turn affect system performance by reconfiguration.