DSoS

IST-1999-11585

Dependable Systems of Systems

Models of Organisational Failures

By

John E. Dobson & Panayiotis Periorellis

Report Version: Deliverable (PCE4)

Report Preparation Date: 1 December 2002

Classification: Public Circulation

Contract Start Date: 1 April 2000 **Duration:** 36m

Project Co-ordinator: Newcastle University

Partners: DERA, Malvern – UK; INRIA – France; CNRS-LAAS – France; TU Wien – Austria; Universität Ulm – Germany; LRI Paris-Sud - France

Project funded by the European Community under the "Information Society Technology" Programme (1998-2002)

Models of Organisational Failure ....

Abstract ....

1 Introduction..4

2 Design by Configuration..7

3 Two Simple Models 9

3.1 Organisational Structure 10

3.2 Organisational Process...13

4 So what can go wrong 15

5 Post transaction communication management 19

6 Application to the Travel Agent Case Study 22

7 Recovery Strategies..26

8 Application to the Examine Case Study 28

8.1 Introduction to the Case study..29

8.2 EXaMINE and Dependability 30

8.3 Organisational Failures in EPS 30

8.4 What have we learned?..34

9 Scoping Organisational Failures..35

10 Further Reading 37

11 References..38

Models of Organisational Failure

John Dobson and Panos Periorellis

University of Newcastle

1. Introduction

The purpose of this deliverable is to provide a taxonomy of organisational failures, so that there is a basis for analysing some of the possible organisational failure modes resulting from putting together two or more organisational systems, each with its own purpose in its own organisational context, to make a new DSoS. It thus provides a way of examining new emergent failure modes. It also points out possible failure modes that, because they are organisational, cannot be prevented (or tolerated) at the level of the individual technical systems or the technical system that brings them together.

In outline, the approach will be to take two simple conceptual models of an organisation, models that are relevant to the idea that a system plays a role in an organisational context, and use these models to describe a number of organisational failure in which a single system can be implicated. By organisational failure here, we mean to exclude technical failures of the technical system itself, i.e. failures to deliver – for whatever reason– the specified service, but to include those many cases where although the service delivered may be considered in accordance with some specification, it fails to fulfil some organisational purpose it was intended to fulfill. We shall extend this discussion to look at problems emergent from a composition of two organisations.

The purpose of these models is not to express anything formal, in the way that a proof of a result expressed in a process algebra would. Their purpose is to facilitate the creation of new knowledge in the context of a particular situation of organizational failure, so that sense can be made of the situation by those in the organisations that are implicated in the situation and the designers of systems to support those organisations. That is why the question of whether or not the expressive language of the models has a formal semantics is irrelevant. There is a difference between a semiotic account of interpretation (what do these signs signify to those who come across them as a found object?) and a hermeneutic model of interpretation (what did this artifact mean to those present in the moment of its creation?) These models of organisational structure and process are deliberately simplified so that they can be used in constructing a story that is intended to be interpreted hermeneutically. They are not primarily intended for use in a context where interpretation is semiotic, and in this they differ from the other , more formal, models in DSoS.

So these models are to be used in the telling of stories about organisational failure. The stories told, like all stories, assume a certain conceptual framework in terms of which the story is interpreted and which makes sense to all those who hear it. The season why this is important for DSoS is that organisational failure is not the sort of thing whose possibilities can be exhaustively enumerated in advance by some well-defined method (such as HAZOPS for certain kinds of technical system). Rather, the task of the designer is to listen to, or to invent, post hoc stories of organisational failure and to consider what response to those stories, in terms of what form of prevention, or tolerance, or compensation, can be incorporated in the design of a DSoS to support an organization whose components are (systems owned by) individual organisations each of which is potentially subject to organisational failure.

The relationship between models of organisation as used in narrative about organisational failure, and the failure management language and apparatus used in software engineering, is not straightforward. This lack of simple direct relationship is due to the differences in kind between technical and social systems. Technical systems are designed, controlled from outwith the system, and exhibit law-like regularities. Social systems are products of social learning, controlled from within the system and any regularities they exhibit are not really law-like.

So the language in which one talks about the management of failure is very different in the two cases. Indeed, one of the most important distinctions is in the connotations of the word “failure” itself. In technical systems, failure is often taken as 'failure to deliver a specified service' and is therefore treated as if it were a property of the system, whereas in organisations, failure is often a judgement about the organisation. It is true that failure of a technical system is indeed also a judgement, but this is often implicit, sometimes so much so as to be invisible; whereas organisational failure is always a judgement and therefore implies the explicit identification of a judge. In this deliverable, the judge is taken to be the designer or configurer of the DSoS, though of course we recognise the equal validity of other possible judges who may arrive at a different judgement. And when talking about organisational failure, it is the specification which is implicit, sometimes so much so as to be invisible.

Although the concepts of fault, error, failure considered as causal concepts are often as applicable to social systems as to technical ones, structural concepts such as boundary, interface, exception, signalling, state and so on, all of which are clearly defined concepts in a software system, do not have such clear definitions in a social system. To elaborate: a boundary is defined by reference to a space within which it is a boundary, but a multifaceted social system has many spaces in which boundaries can be drawn and it is often unclear which is the best space (or spaces) to choose as elements in a particular story, let alone where to draw the boundaries in each space. Interfaces do not really exist between social systems, their places being taken, at least in some cases, by boundary objects which are interpreted differently on each side of the boundary; but these boundary objects may share but few characteristics with the software concept of an object. Exceptions and signalling can occur in some social systems, but more often they are unplanned and implicit rather than planned and explicit as they are in technical systems. This is because in the lack of design and external control, it is often unclear to whom an exception should be signalled and what constitutes a signal and in any case there is often a failure correctly to interpret the signal ("Not waving but drowning"). Finally, state is a very problematic concept when applied to social systems, except in a very loose and metaphorical manner of speaking.

What this means is that the kind of thinking we have to employ in thinking about organisational failure is very different from the kind of thinking we employ in the presence of failure in a technical system. Only by making extreme simplifications of the concepts of organisational structure and process is it possible to bring discussions of organisational and technical failures into a common discourse. In many cases this will turn out to be an over-simplification, and perhaps only in very simple organizational relationships such as retail or simple brokerage they are good enough. Since this a fairly new and unexplored area we are using the Travel Agent case study [DMS3] to bring a technical and organizational discussion under the same domain, but we will explore the point further by discussing our second case study. During the later stages of the project we looked into the European Electric system (EXaMINE), and we will discuss some of the organizational failures we found in such a system and the impact they may have. Since the second case study has not been discussed before, we introduce it in section 8 prior to applying the concepts presented in this deliverable.

It follows from the distinction made earlier between semiotic and hermeneutic interpretation that the form of this deliverable is different from some other DSoS deliverables. A technical paper with technical apparatus such as formal notations with a formal semantics, learned references to current literature, and an epistemic view of knowledge as abstraction and interpretation as semiotic, is not appropriate. This deliverable is itself more of a story, with a structure which is appropriate to the structure of a story, which presents a set of concepts from which other stories -stories of organisational failure- can be constructed. Its epistemic view is that knowledge is mediation, and in the context of DSoS in particular, that knowledge of organizational failure is mediated between the narrator of organisational failure and the designer of a DSoS using a conceptual framework understood by both. It is an illustrated story, with illustrations not only in the form of pictures but in the form of scenarios that could occur in the context of our travel agent and the European electric system case studies, though the models can be, and indeed in the past have been, used to account for failures in forms of organisation other than a commercial brokerage enterprise.

2. Design by configuration

The theory of design that underlies DSoS is design by configuration - i.e. the process of designing a system to achieve a certain human purpose is by configuring a set of already existing component systems over which the designer has little or no control, and which together might achieve a sufficiently close approximation to that purpose in order to deliver a controlled result (of any kind, including abortion) in the presence of arbitrary failure including organisational failure. Where the designer has little or no control over, or indeed knowledge of, the component or the organisation which is its context, the law of requisite variety [RA1964] suggests that a simplified view has to be taken both of the services(s) offered by the component and the structure and processes of the organisation, and of the management of their various distinct modes of failure. This is equally true of organisational and technical components and failure modes. In this deliverable, we shall look in particular at failures that can be judged to be failures in configuration. This involves looking at the relationships between organisations, and thus concentrates less on failures of intra-organisational configuration (though these can form a starting point) than failures on inter-organisational configuration. One of the conditions of possibility of design by configuration is compositionality - that things can be put together (in some space) in a way that does not perturb or compromise the integrity of each. Indeed, this is the basis of DSoS and its insistence on simple interfaces in the technical domain. However, there is no reason to suppose that organisations -as organisations, not as the services they deliver- can be composed in this way. In particular, a simplistic view of service delivery and its failure will not do in the presence of organisational failure, because the failure to deliver a requested resource because the organisation has none left to provide has to be handled very differently from there being no organisation left to provide the requested resource. Indeed, the decisions in the two cases as to where to place what kind of signalling mechanism and to whom the situation is signalled have probably to be taken at different points in the design process. Another case of having to know more about a configured infrastructure than is visible at a simple interface is when the organisations have combined structurally in some way. For example, when two airline companies share the same route it can be important to know whether they operate a common tariff and joint recognition of each other's tickets (which would imply some co-ordination at the management level of the organisation) or not. Again, this becomes an issue in the presence of failure, for example if a traveller had planned to catch a plane operated by one company which is cancelled, thereby forcing a wait for the next plane, which turns out to be operated by the other company. The reason for taking a simple approach to organisational failure is so as to delineate clearly what sorts of failure can, and what sorts cannot, be allowed for in the design of a DSoS configuration. For example, a change in market positioning on the part of a supplier which does not result in a renegotiation with its customers at the management level of the relationship is likely to lead to inter-organisational failure and misunderstanding which cannot be adapted to by changes in technical interfaces. More generally, the approach adopted in DSoS is to explore whether and to what extent the desired effect of configuration by design is achievable through the combination of very simplified models of technical linkage matched by equally simplified models of business linkage. It is in the nature of business relationships that the concept of a simple interface only facilitates very simple relationships, such as that of a drive-through hamburger outlet. Whenever the relationship is more complex, it is negotiated, and like all human negotiations, what is negotiated is the limits of what may be requested and the limits of what may be offered. It is one of the advantages of the travel agent case study that a DSoS approach is neither so simple that it is prima facie possible (like the combination of two very simple drive-through hamburger organisations) nor so complex that it is equally obviously not possible (like the combination of two hospitals into a single new organisation). We feel that our case study lies neatly between the boundaries between what is possible and what is not, so that it is a research issue which we are investigating whether the simplified business models presented in the next section are adequate for the sense-making that must necessarily precede an account of management of organizational failure in configured systems that span multiple domains of management.

In the case of our second case study, which deals with a much larger and more complex system of systems, the concepts of organization and boundary have to be extended to map onto nations and national boundaries. The underlying theme and common conclusion in both case studies is that since computer systems reflect organizational thinking in terms of goals, strategies, they can bring computer systems under a SoS in conflict; and this is where what we call organizational failures arise.

3. Two simple models

In the following paragraphs we present two simple models to develop our concepts using examples from the travel agent case study. In section 8 we apply these concepts to a real system namely the EXaMINE case study. The two simple models we present show two different but related aspects of an organisation: **structure** and **process**. There are many dozens, perhaps hundreds, of models of organisations in the literature of organisation theory, but the models we present here are simplified abstractions common to most of them in one form or another. Briefly, the structure model is a standard one of dividing the organisation into responsibilities for *direction*, for *management* and for *execution*. The actual structure of any particular organisation is determined –to a greater or lesser degree– by how these responsibilities are mapped onto individual role holders and individuals in the organisation; this can obviously vary from one organisation to another. What is invariant across organisations is the existence of these three types of responsibility, and the fact that they are mapped onto roles and individuals.

The process model divides organisational processes into three types: *scoping* the business (i.e. deciding what the organisation is about), *resourcing* the business (i.e. procuring and managing the resources needed for the organisation to do whatever the scoping process determines it should do) and *delivering* the business (i.e. the actual performance).

It is important to realise that this process model is not just a re-articulation of the structure model. An organisation that simply combined direction with scoping, management with resourcing and execution with delivery would be very naive and not very effective. At the very least, each of the scoping, resourcing and delivery processes would have its own internal D/M/E structure within it, but the actual relationships in practice show a wide variety of configuration possibilities.

3.1 Organisational structure

As indicated earlier, we classify the responsibilities that exist in an organisation into direction, management and execution responsibilities.

Direction responsibilities are for deciding on desired future states of the organisation, for enunciating strategies for achieving those states, and for allocating generic resources (e.g. overall budgets) to enable the achievement.

Management responsibilities are for turning policy objectives and strategies into plans, for transformation of the generic budget into actual resource instances and allocating and deploying them. And of course there are required back channels of reports and accounts.

Similarly execution consumes the resources in fulfilling (or not) the plans and reporting back.

So we have the following simple model:

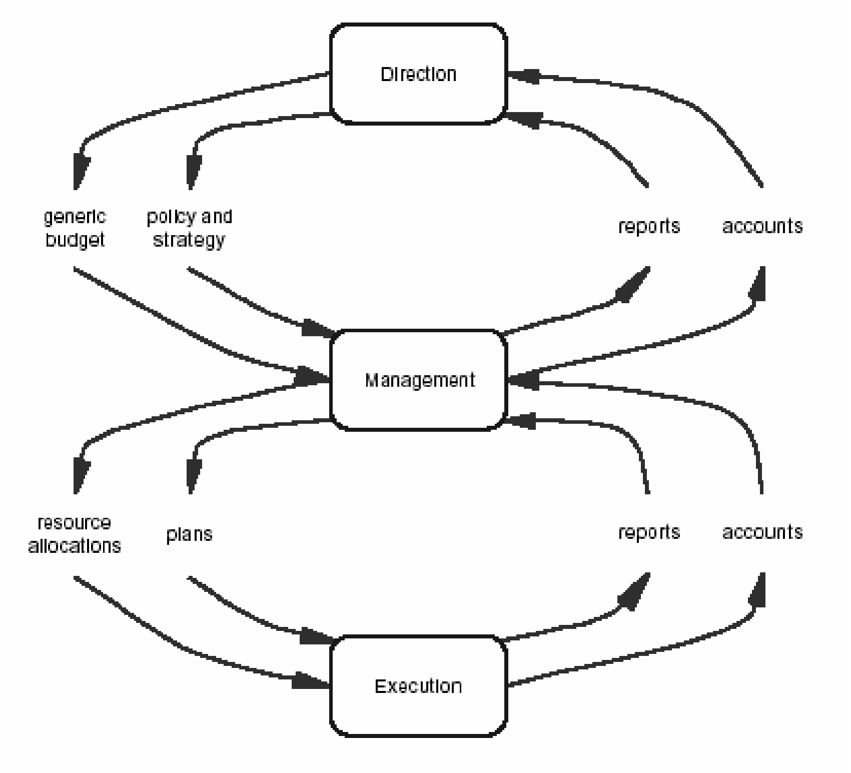


Figure 1

It is important to realise that these structural components can be seen as distinct collections of responsibilities for dealing with distinct units of failure. It enables the distinction to be made between having the wrong policy, having the right policy but an inappropriate set of plans, and having the right plans but failure to execute them correctly. It also allows for explanations of failure couched in terms of inadequate budget, inappropriate allocations, or inadequate or incorrect reports and accounts. How these responsibilities map into roles and actors in the organisation is a matter of configuration - and configurations can be faulty too.

We can hide the information channels to produce the following simplification:

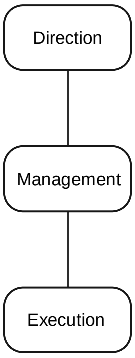


Figure 2

When we compose two organisational structures, there are two levels at which composition can take place:

shared management

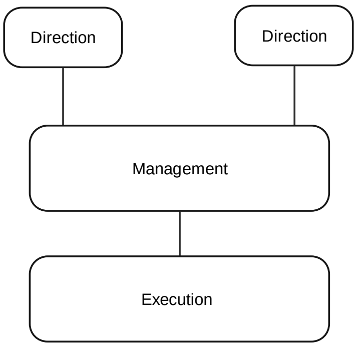


Figure 3

shared execution

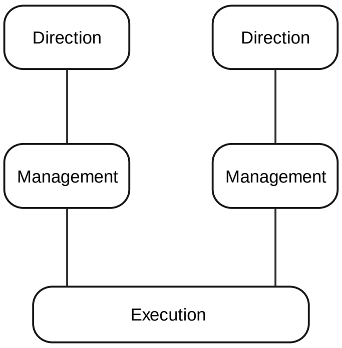


Figure 4

One example of the first is the way Sir Peter de la Billière as commander of the

British troops was able, during the Gulf War, to insert himself in to the overall operational command team led by the American commander. Although he reported to the British Prime Minister rather than to the American President, the British troops saw the same single command structure to whom they were responsible as did the American troops. An example of the second is when an aero engine manufacturer subcontracts out the development of its avionics software to a specialist firm, yet insists on having some of its own staff on the development team in order to protect its interests in the confidentiality of the control laws being implemented, which it regards as intellectual property too valuable to leave in the hands of subcontractors.

We can now start enumerating different failure modes. In the shared execution case, different plans can conflict or interfere, different reports and accounts can be passed upwards on the two different channels, execution failure can result in different recovery actions at the management level, management can disagree on the allocation of responsibilities, and so on.

Similarly, in the case of shared management, there are opportunities for conflicting policies, differing reports and accounts, arguments over managerial responsibilities and so on.

The point of these pictures is to provide a simple pictorial representation so that when a particular failure is analysed, it is clear at what level in the organization exception signaling and recovery mechanisms can be placed.

3.2 Organisational process

We now turn to an equally simple process model. As indicated earlier, this is a model of the *kinds* of process that take place in an organisation.

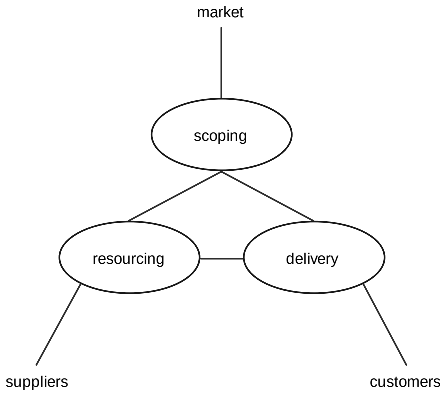


Figure 5

As indicated above, scoping processes are market-facing, dealing with questions such as deciding on position within the market –and indeed which market– monitoring the movement of the market and deciding on appropriate responses, and –for some kinds of business– actually making the market.

Resourcing processes are supplier-facing. They are concerned with acquiring sufficient resources (including of course human resources) to run the business, maintaining those resources, monitoring their quality, managing suppliers and so on.

Delivery processes are customer-facing. They are concerned with obtaining and fulfilling orders, enlarging the customer base, obtaining feedback as a useful form of input to the scoping processes and evaluating the resourcing processes.

Again, it is easy to see a number of possible failure modes immediately. Inappropriate market positioning, inadequate resourcing and delivery processes, failures in communication both within and between these processes. From the point of view of DSoS, though, it is again the case that we are interested in the additional failures that can occur when organisations are composed.

One very common way in which these processes are composed is in a supply chain or network, in which the unit of composition is by linking the resourcing and delivery processes:

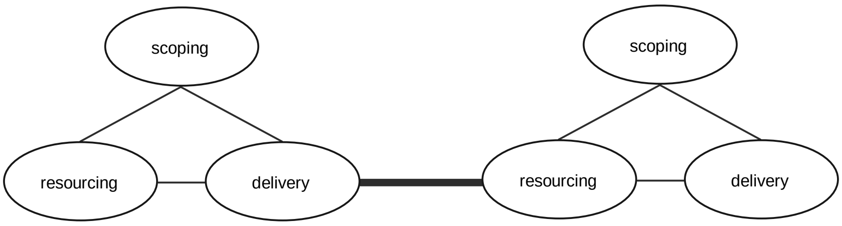


Figure 6

Or with multiple suppliers and customers

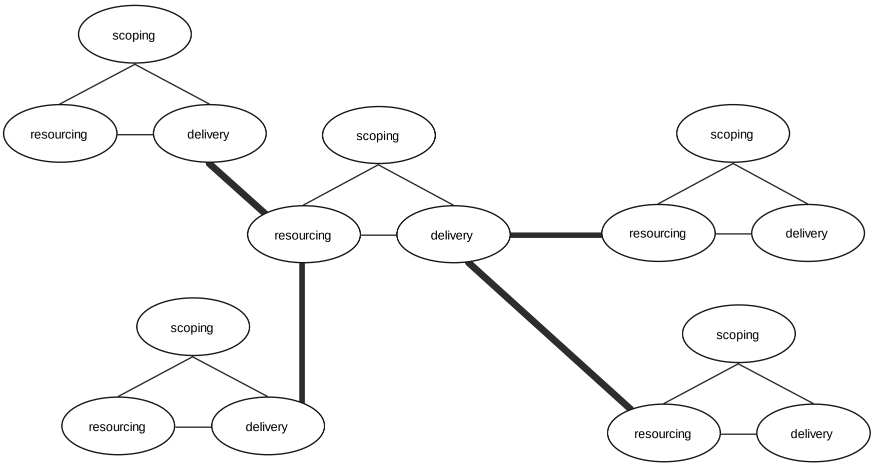


Figure 7

We can now see additional failure modes concerned with mismatch of various kinds between organisations represented by the double links. In the simple case of a chain which is concerned with extracting value from a market,

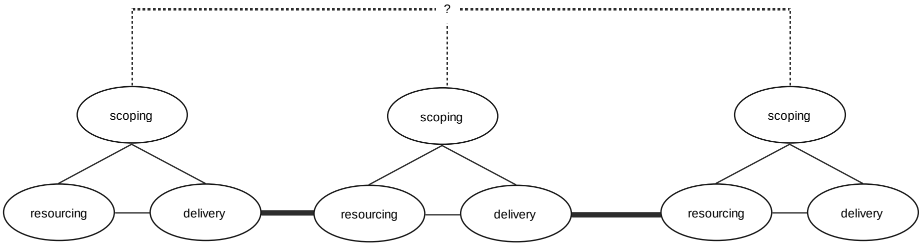


Figure 8

there needs to be some agreement on the apportionment of value, and this can only occur within the scoping processes. It is no good if the prices charged at the bottom end of the value chain mean that the prices at the top end are so high that the market is no longer viable.

There is also the possibility of mismatch between the scoping decisions. For example, a travel agent will normally try to match the market position of the hotel with the market position of the airline, preferring to fly travellers to cheap hotels using no-frills carriers, or to expensive hotels using full-service carriers, and indeed may choose to specialise in one end of the market or the other. Another example is an aero-engine manufacturer who changes their scope from being a vendor of engine units to being a provider of thrust-hours. This might, or might not, prove to be a source of organisational failure. However, such a decision would have an impact on the management and execution structures. It would also have an impact on the relationship with the company's customers, which would change from being a product and maintenance relationship to a service relationship. One form of organisational failure would be a failure to renegotiate this relationship.

4. So what can go wrong?

Although the scoping/resourcing/delivery model describes an organisation in terms of processes, these processes do not always (or indeed hardly ever) employ distinct mechanisms. Thus the delivery mechanism will embody aspects of the (results of the) scoping process; and so on all the way round. And, as explained previously, in an effective value-adding network, the scoping policies of individual enterprises may not be independent. This means that we cannot assume that we can provide a failure[1]-proof travel agent simply by connecting together delivery mechanisms from component suppliers. Examples of scoping mismatch include marketing policies (e.g. different booking systems can assume it is the customer, or an agent, who is interacting with the system), systems with differing models of trust (e.g. credit card authentication required before the transaction can begin, as opposed to authentication when the transaction is committed), and so on. These policy mismatches should not be seen as mere technical

[1] where failure, here as always in this deliverable, is a failure to deliver a desired service to the customer, not just failures of a technical mechanism

glitches to be overcome by ingenious Java programming in the travel agent system. Rather, they are policy mismatches which constitute a fault which may result in a failure of the travel agent to deliver an adequate service to the customer, and recovery management needs to be addressed at that level. In the EXaMINE case study where component systems are behind national boundaries these policy mismatches can take the form of political debates to be resolved via diplomatic channels and appropriate regulatory bodies before the SoS is formally specified. We discuss these issues in section 8.

Two particular sources of problem arise from (a) post-transaction failures and (b) post-transaction changes.

If one component organisation or service in a brokered package of services fails, then under some but not all circumstances it is the responsibility of the service provider to make alternative arrangements, or compensate, for the loss. Where there is a lack of transparency, which is particularly true of the travel industry, it is understandable that a travel agent may not wish to make it clear to the customer in advance what the possible failures are and how they might be recovered from. In the event of airline failure (whether lack of aircraft or cessation of trading), for example, some –but not all– airlines will themselves try to rebook passengers on other flights. Some –but not all– hotels will seek to re-accommodate travellers if booked accommodation is not available. Some –but not all– travel agents have an emergency number which clients can phone for assistance in the preceding cases. And so on.

There are three major strategies which can be used to deal with these post-transaction organisational failures. They are

Forward recovery

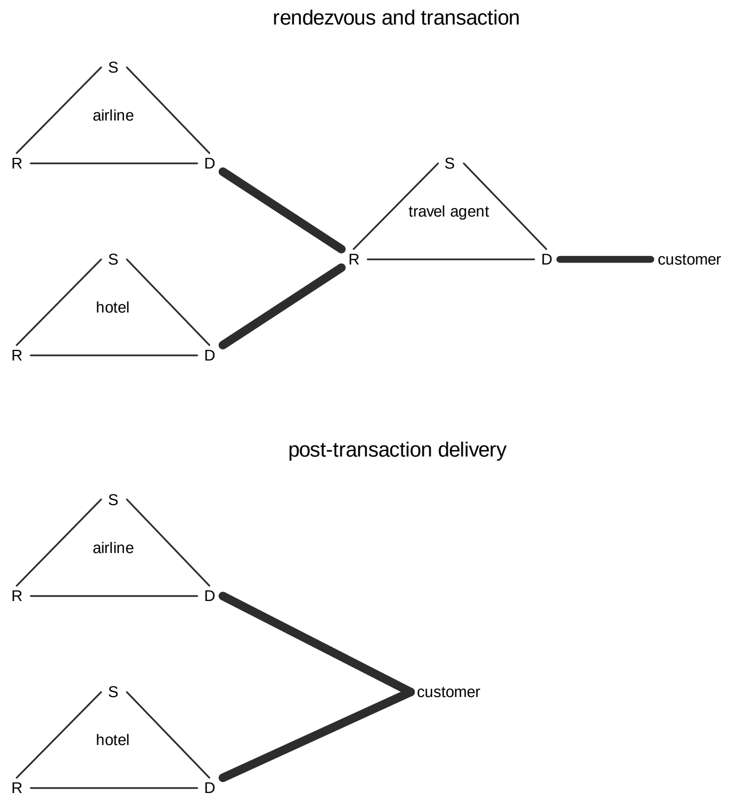
Alleviation

Compensation

Fault-tolerance (“We have booked you onto BA *and* KLM *and* United Airlines so that even if two of them fail, you can still get to New York”) does not seem to be an option.

An example of forward recovery is rebooking, either by the failed airline or the travel agent, onto an alternative carrier. An example of compensation is leaving the responsibility for alternative arrangements up to the traveller who can then claim on some insurance policy — either traveller’s or the travel agent’s. An example of alleviation is the facility offered by some charge cards that under circumstances of failure, a certain amount may be charged to the card which will not be recharged to the cardholder (usually provided the original charge was made on the card).

We can draw a simple model to show how these strategies relate to the simple S/R/D process model:



rendezvous and transaction

post-transaction delivery

Figure 9

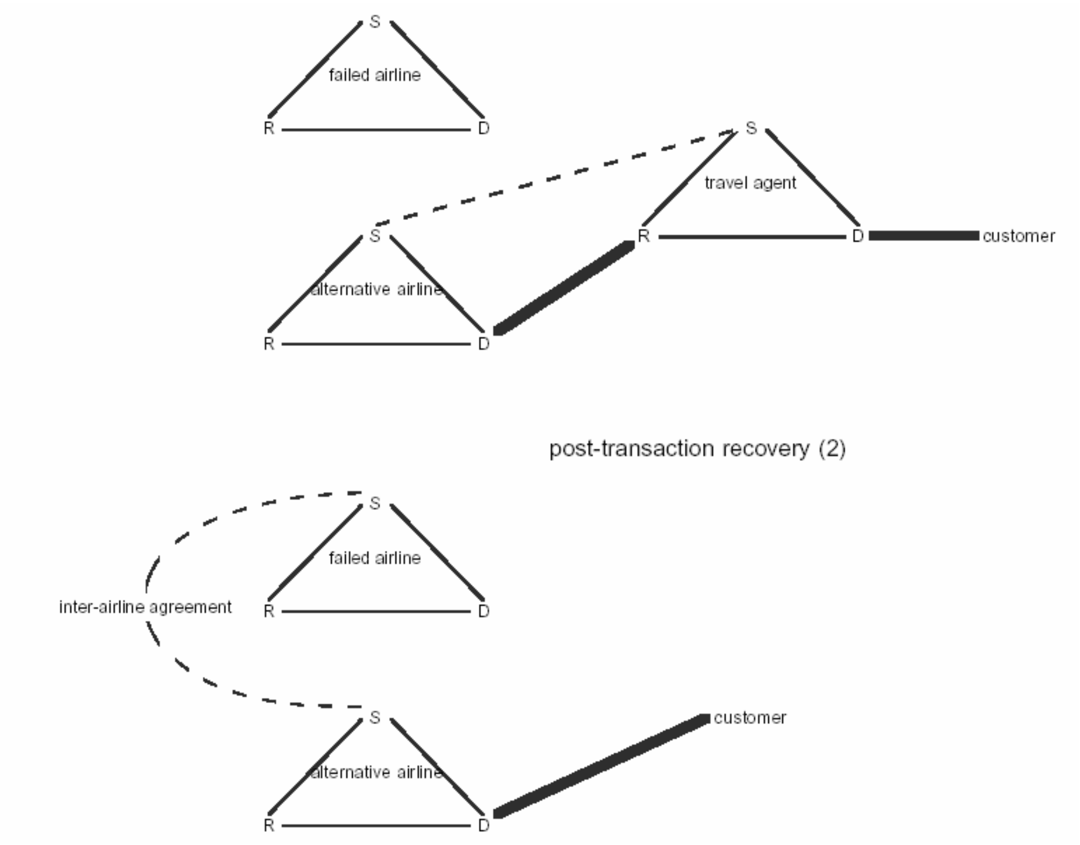


Figure 10

As indicated in the diagrams above, post-transaction recovery needs to be co-ordinated through the existence of some communication channel between the scoping processes of the individual organisations. This can be relatively informal, or it may involve an additional organisation such as a trade association of some kind (such as IATA or ABTA). Such organisations will have their own system and access mechanisms (which may or may not be accessible to the traveller and may or may not be online to the travel agent). There is also a degree of freedom concerning the level in the accessing organisation at which the access is permitted, which may be at the execution level or the management level.

We have here an example of a commonly observed phenomenon to which perhaps not enough attention is paid, which is that in the presence of failure it is often necessary to expand the boundaries of what is considered to be the system. In the case of a DSoS, this boundary expansion can occur with respect to any of the component systems, and it can also occur by having to consider some new system as part of the DSoS whose function will only be invoked in failure recovery. Identifying the need for, and appropriate use of, such systems requires a systematic approach to the analysis of a DSoS with respect to organisational failure through the construction of appropriate stories. So a design proposal for a design as configuration process is to think through the possible organisational failure modes and decide what level of recovery (including none, of course) is feasible.

5. Post-transaction Communication Management

To do business with an organisation requires knowledge of three things: how to communicate with the organisation, how to transact with the organisation and how to recover if the organisation fails. For example, if an organisation receives a message but due to internal communication problem is unable to deliver it to the correct place, it is usually not effective simply to resend the message; an alternative route must be found, knowledge of the management structures sometimes being of assistance here, or perhaps an alternative sender must be found such as a lawyer of prominent consumer journalist. Transaction is more complex than communication, and it is important to discuss post-transaction management since it has many implications both at the policy level and the delivery system. Some of the questions we can raise to illustrate this point are Who is responsible for informing the client about changes of the trip's details? and Who handles customer complaints?

We start discussing problems of communication by asking the following question: Can the component systems (autonomous systems) contact the TA to inform it of changes in policies, service etc?

We have taken the view that autonomous systems expose a call interface via which we send requests regarding their services. The implication of maintaining the ability to initiate and terminate a conversation rules out the possibility of a component system informing the TA about changes in its structure, policies, operation etc. This has a major implication on the organisation of the TA and poses a major challenge. Alterations in the operation, policies and scope of the component system need to be detected by the TA itself. The immediate question is whether we can we successfully detect these via the call interface only. As we have so far made clear changes in policies, customer base and operations cannot be viewed by the call interface. The protocol may remain as it is even if the customer base changes.

More generally, the travel agent case study presented here is concerned with multiparty transactions which are distributed over many locations and which may require a considerable time to complete. Each party in such a transaction has a set of preconditions and a set of post-conditions which must be met before the transaction is judged to have been successful from that party's point of view. Thus, for a transaction to be judged to be well formed, the evidence, embodied in a set of instruments, must reliably reflect the intended acts of remote parties. For this to be the case, there are three characteristics of the instruments and the operations on them which must be assumed:

Atomicity: specific actions occur exactly once or not at all and the parties are able to confirm completion of an action.

Persistence: once information is generated it does not disappear; it may be changed, but the instrument(s) must record the original and the changed information.

Security: which, in this case, refers to the authenticity and integrity of information represented in instruments.

There are two configurations of the relationships of a multi-party transaction at the structural level:

- A centralised transaction monitor in which each of the participants has a direct relationship with one particular participant in the role of transaction manager. The logical point of co-ordination is also a physical point of control.
- Distributed transaction management, in which each participant undertakes transaction management responsibilities and the logical point of co-ordination is, in fact, replicated and distributed.

In the first approach, which is implemented by transaction monitoring functionality, all transacting parties must have a pre-defined relationship with one particular party responsible for the co-ordination. "Pre-defined" here means that these relationships were established outside the context and infrastructure in which the transactions will be executed. In the second, which is implemented in distributed transaction management, each party depends on all the others and must be able to monitor and interpret their acts.

These two approaches to the allocation of responsibilities in a distributed transaction result in a different relationship between structural and infrastructural responsibilities. In the case of the centralised transaction monitor, the participants depend on the nfrastructure only for potentially unreliable message transport services. Atomicity of operations, persistence of information and security, authenticity and integrity of messages are dependabilities or qualities of service which are delivered at the structural level either as end-to-end or centralised mechanisms.

In the case of a distributed transaction, the economies of provision are quite different. Since each participant takes responsibility for components of the transaction and needs to be able to monitor remote activities and states, each needs to be able to rely on the quality of a set of service and applications components within their own domain and in each of those of the other participants. In this case, the pre-established relationship must be with the infrastructural suppliers and it is possible that the transacting parties are establishing a new context as well as a new instance of commerce. In this case, new instruments, which arise from the characteristics of the new context, may well be required.

In the distributed approach to transaction management, and here we are concerned not merely with distribution over time and space but, more significantly, distribution over the boundaries of different enterprises, each enterprise must have the option and capability of replicating all those aspects of transaction co-ordination which are relevant to their particular interests. They must also be able to rely on the provider of the infrastructural environment to ensure that their view of the current state of any transaction is coherent with the views of all the other participants of that transaction. Thus, atomicity, persistence and security become responsibilities of the environment provider and infrastructural in nature, and it is these qualities of service and application which dictate the characteristics of the instruments of the structural conversations. We hope to show, in deliverable DSC3, how these considerations can form the basis of an approach to the management of at least some forms of organisational failure.

6. Application to the Travel Agent case study

The purpose of this section is to indicate a number of examples of how the simple models introduced can be used to think about organisational issues arising in the design of a DSoS to support an online travel agent (TA). Previous deliverable CS1 has introduced the travel agent case study and explained our approach to this form of brokerage.

The structure model of the TA reveals its scope, resources and delivery system. The TA provides full trips consisting of separately chosen accommodation, flight and vehicle to holiday makers. This automatically sets the market which the TA targets. After the market has been identified there are certain assumptions and decisions that need to be made. Bearing in mind that the scope determines the type of resource, we have selected a number of booking systems which are considered appropriate for the TA in the sense that their scope is compatible – we are, for example, excluding package holiday providers.

Within the scope we have defined a number of policies regarding the operation of the TA, assumptions about the clientele, the interaction process and the overall responsibility held by the TA. The selection of the type of resource is based on the assumption that the booking systems comply with the scope we have determined for our system.

The delivering system provides the service determined by the scope using the resources. Again, the type of delivery system is determined more by the scooping decisions we have made than by the resources brokered. Design decisions, and implementation schemes are based on this.

The following diagram illustrates what we have discussed so far.

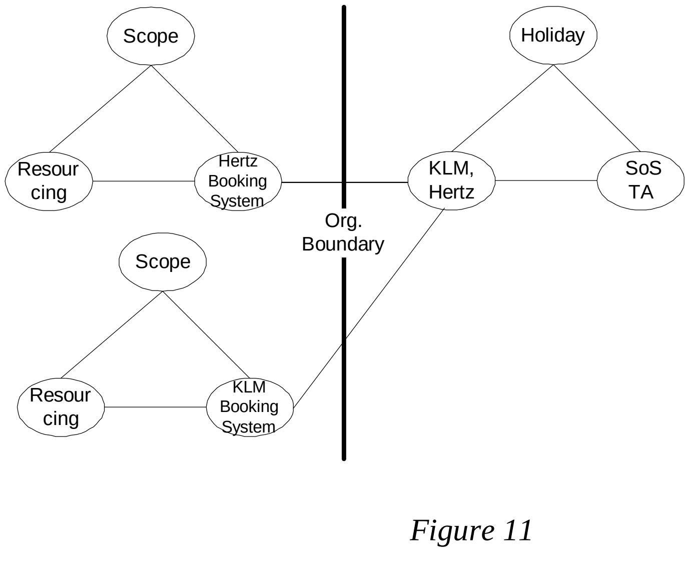


Figure 11

Notice that the connections between the resources and the TA are taking place via the linking interfaces of the delivering systems. The booking systems are assumed to be autonomous organisational structures that provide a particular service. We obtain their services via a linking interface. The thick black line indicates an organisational boundary which prevents us viewing the scope of the participating systems. We only have access to the service i.e. the delivery system.

Observing a system from a call interface only provides a limited view about the system's services and in particular the policies and scope of those services. Given that the scope of the TA defines its market target and the interaction with its clients it is wise to obtain resources (i.e. booking systems) whose scope complies with the scope of the TA. The scope however cannot be seen via the call interface. We could indeed observe and test a LIF to improve fault tolerance of the overall system but we cannot obtain information about the scope of the service (e.g. who it is intended for). We can obtain partial information on how the service is delivered by looking at the protocol and the relevant transactions that take place but we could not for example obtain information about the policies regarding a protocol, authentication, levels of trust etc. This limitation can be accommodated within a static system because decisions can be made prior to setting up a TA delivery mechanism. What cannot so easily be done is to detect, and respond to, changes in the scoping decisions of suppliers.

Some typical failures would arise in the following scenarios. We have assumed that the TA policy has defined a way which the TA interacts with customers and processes their requests. Such policy assumes, for example, that the TA does not require registration and shows a high level of trust in its clients. We also assume the TA chooses to consider a journey as an end-to-end arrangement.

a) Booking System A requires customer registration.

This is in direct conflict with the protocol of the TA since the latter does not require registration. Although protocol differences like this can be observed at the LIF, additional policies regarding the levels of trust (which the registration process is related to) cannot be observed at the LIF. This can be regarded as a mismatch between policies of organisational trust.

b) Company A is a 'no-frills' airline (customer base).

Such airlines typically take no responsibility for knock-on consequences of delayed or cancelled flights, even when the next leg is one of their own operations. This is an example of market target clash between the TA and the supplying system. Obviously this cannot be observed at the LIF and additional monitoring mechanisms at the level of the TA would be needed to resolve these issues. This is in direct contrast with the scope of the TA which targets independent holiday makers traveling on full-service airlines.

There could be many examples like the ones above that illustrate the point that LIFs do not offer adequate information for including an autonomous system as a resource of a SoS. We count these as examples of organisational failure since recovery, if it is to be achieved at all, has to be done at the level of the organisation.

Let us view another category of failures that is raised during the actual service delivery. Remember that we have assumed that systems are autonomous in the sense that they operate outside the scope of the SoS.

c) Inaccuracy of information

The quality of information produced by the booking system could hinder the overall service offered by the TA. Questions such as ‘is the information up-to-date’ and ‘is the source reliable’ cannot be answered by only looking at the LIF. The main question that we need to consider when we compose a service from various sources is whether the quality of data adheres to the quality expected by the TA and its customer base. It is a sad fact that some operators publish incorrect information about their services on their own websites. Here again it is a policy decision to be made by the TA how much effort they are prepared to spend in dealing with recovery from failures experienced by their clients due to misinformation outside the control of the TA.

d) Service offered differs from the service promised or advertised

This is closely related to the first point and again unless performance records are maintained this type of information cannot be found at the level of the LIF. However, this raises the possible need for an additional interface to the TA which allows clients to submit reports to the TA on the services brokered by the TA. This is a facility already offered by conventional travel agents, particularly those serving the business travel community.

e) Time semantics

It would be wrong to assume that component system operate on the same time semantics regarding the handling of requests. In fact it is likely that component system will operate according to their own semantics which are embedded into the delivery system and hidden from view. The TA needs to be aware of these prior to making any requests. Consider the following example. The TA has a 30 second timeout rule. This implies that a reply for any request has to be received within 30 seconds. If this does not happen the TA throws a timeout. The booking system however operates a queue which due to its nature and the number of requests received operates a 45 second timeout rule. This as one can imagine can lead to the booking system actually making a booking i.e. handling the request successfully while the TA thinks otherwise. Can we observe time semantics over the call interface?. Unfortunately these are set as part of the operation policies of each system and are hidden from view.

We have shown so far that call interfaces (LIFs) can only offer some indication regarding the services of the booking system. In fact we have shown that although they can show how a service is delivered (protocol) they do not indicate the operation policies of the service. Additional interfaces are necessary to do this.

Composing an emerging service out of services obtained from autonomous systems can lead to failure due to a number of errors. We summarise these below.

a) Market

The service may not be intended for the same market base. These targets are set as part of the scope of the organization and are therefore invisible at the interface level.

b) Protocol

Although the protocol is visible and to an extent it can be manipulated, we cannot always assume that we can compose the trip using any component system. Although wrapping would allow us to hide some of the incompatibilities regarding for example requirements expressed as additional requests, they cannot hide certain aspects of the interface that are part of a wider policy e.g. authentication, user registration etc.

c) Reliability of Service

As we have mentioned earlier this cannot be assumed and additional mechanisms would have to be in place to ensure that the same quality as described by the scope of the TA would be maintained throughout.

d) Responsibilities

There are a number of responsibilities that need to be assigned to certain roles in order to avoid failure. Consider the following questions: Who is responsible for informing the user of changes? Who is responsible for compensating the user? Can the user cancel the trip and within what time scale? All these responsibilities need to be assigned roles in order to avoid failure.

7. Recovery Strategies

Some of the organizational failures can be prevented by adding additional layers of exception handling, maintaining additional information about each component system, and keeping track of performance records. Some of the following could solve some of the errors that give rise to failures.

a) Accessing additional interfaces.

Although this may not always be possible it is desirable to obtain some information (such as public information) about the scope and policies of the systems. An interface between the scopes of the two systems would eliminate failure raised from clashing policies. It would also provide a better idea as to whether it is feasible to include a particular system as a resource. However, to be effective, this would require support at the management level of the separate organisations.

b) Metadata

Metadata information could be used to maintain certain policies in data structures. While not all policies can be represented in data structures, keeping metadata about customer base, protocols and authentication would allow the TA to reason about the composition of the emerging service (e.g. why provision of the emerging service by components *a*, *d* and *f* is better than *a*, *b* and *c*). Since operation policies can change, metadata would also need to be changed. Additionally we would need to ensure a reliable interface for obtaining such information.

c) Model of Responsibilities

While maintaining the brokerage model of operation, providing the user of a model of responsibilities (e.g. who is responsible for what) would help the TA to assign roles for every responsibility (cancellation policy or changes in trip details). As we mentioned earlier post transaction management needs to be dealt with at the level of the SoS and all responsibilities derived by it need to be assigned roles.

d) Composing according to user requirements

Being able to compose an emerging service according to user requirements would allow the TA to avoid failures regarding clashes between certain policies (the user is a backpacker while the trip is for business class travelers). This of course implies certain technical implementation in order to obtain data about the type of client and address that particular client using the appropriate service.

e) Maintaining performance records

Maintaining records on different compositions and users would help the TA to assess the compatibilities between the services offered by its component systems and the type of users it services. It would also allow the TA to evolve its services according to the evolution of the component system. Additionally they could provide an indication about changing policies, scope, operations etc.

f) Involving the user

Finally involving the user in certain decisions would allow the TA to drop certain responsibilities. The user can be involved in selecting a particular configuration of a particular trip that addresses his type. The TA could additionally make suggestions about certain configurations and maintain track of users' preferences. This would help resolve a number of issues regarding targeting the right customer with the right service. Involving the user in this process would help the TA to resolve issues raised by quality of service (user can select a service based on past experience), reliability, accuracy etc.

Having discussed the domain of organizational failure using examples from the Travel Agent case study let us discuss these issues from a different angle. In the next section we are taking a look at a case study where the types of failures we have discussed so far have to realized and resolved prior to requirements engineering. We will see how social, economic and political factors play an important role in the delivery of a service whose components reside behind national boundaries. As we mentioned earlier although we received the case study at the later stages of our project it serves as a good way to ground our concepts.

8. Application to the EXaMINE case study

During the later stages of the project we identified a case study which we think can help us expose a number of our concerns in a real life scenario. We have on many occasions in this document used the travel agent case study as an example. The EXaMINE case study is a study of the vulnerabilities of the European Electric System (EPS) mainly in physical technical terms (both internal and external). We have however identified a number of possible and real scenarios that need to taken into consideration in order to assess the feasibility of such a Pan-European network of power distribution. In the document so far we have talked about organisational structure and process and how faults in these can eventually lead to what we call organisational failures. Given the current case study which deals with a number of interconnected grids operating under national law, we would like to abstract and show how political, social and economic factors embedded inevitably into such a system can lead to “failures”. Notice that we are not abstracting from the previous chapters but rather expanding the notion of organisational boundaries to national boundaries where laws, regulations and governing bodies affect the way a system is architected and consequently managed.

8.1 Introduction to the case study

The European electric power system (EPS) is composed of a set of national grids each of which is in turn itself a complex network, including a large number of components and devices. The behavior of the EPS is controlled by a layered structure designed to cope with national grids usually encompassing a few regional control centers and one national [EX2002a]. Contractual arrangements determine the degree of power interchange that can take place. The organization here (equivalent to the component system of a SoS) is the national electric power system which operates behind national boundaries and under government regulation.

The EXaMINE case study is studying the feasibility of the development of an EU wide network to support power distribution. It is particularly interested in the dependability aspects of the network mainly from a technical perspective. We are using the case study mainly to show that there are a number of additional issues namely political, economic and social factors that can determine the architecture, design and implementation of such a network.

The aim of the EXaMINE case study is to assess and identify an optimum security model for the EU wide electric power system. We are merely concerned with the organizational failures that could occur in terms of political, economic and social impact. The basic idea behind the case study is that the various national networks could work together to form a network of networks. This is not much different from a system of systems. Likewise a SoS where part of the emphasis is on crossing organizational boundaries (implying dealing with diverse political issues, cultures, goals etc), a network of networks for power supply within the EU is an abstraction of a SoS, in the sense that it crosses national borders, laws, governing bodies etc. For consistency purposes we are referring to the EPS as a SoS.

8.2 EXaMINE and Dependability

In the beginning of this deliverable we stated that although in technical terms failure is the deviation of the behavior of the system from what was intended, an organizational failure is a judgment for the organization. In this section along with our original goal we will use some of the dependability terminology to assess how it applies. We know [JCLDep] that dependability is the property of a computer system such that reliance can justifiably be placed on the service it delivers. Applying this to an Electric Power system we could infer that dependability of such a system refers to its ability to provide a continuous service its users. Security in this sense is also related and affected by the reliability of the system which is the property of the system that ensures continuous service. Of course a system such as an electric power system is permanently exposed to a number of internal and external problems, ranging from component faults to terrorist attacks. For an electric power system that extends throughout the EU, properties such as *readiness* which implies availability and *continuity* of service which implies reliability are essential. In the EPS these properties are not only properties of a computer systems that delivers this service but also properties of the political and economic network that plays a substantial role in its operation. As we shall see the whole issue of dependability is also political issue.

The purpose of the case study is not to address these issues in technical terms but to show how organizational failures (sometimes influenced strongly by economics, social and political factors) play an important role in the delivery of such a service (through a computer system).

8.3 Organisational Failures of the EPS

In the travel agent case study we envisaged organisational failures stemming from incompatible customer bases, diverse goals within the organisations, lack of communication and so on. Within an EPS we can classify these in terms of economic factors (e.g. market environment), political factors (e.g. authority, power balance) and national factors such as interests and security all of which are intertwined. An additional dimension in EXaMINE is that although in the case of the TA we had overall control maintained by the SoS itself here control is distributed. In fact as was mentioned earlier control is the major cause of "failure" in EXaMINE. Notice that we are moving from organisational to national boundaries where competitors in the TA case study can take the form of political enemies in EXaMINE.

From the EXaMINE case study [EX2002b] we learn that economic changes within the EU call for a more dynamic market based environment and therefore re-structuring of the entire electric sector, following the European Community directive 92 of 1996. The result of this directive is open competition and third party access to the transmission systems, which in turn increases the volatility on the system as a whole. Competition in an open market can create strong demand for interconnection capacity to cope with the amount of interchanges across the full interconnected system. Since we have the technology to visualize such systems we ask the same question; What can possibly go wrong?

Crossing organizational boundaries implies in many cases crossing cultural differences, laws, goals. A European wide electrical power system is effectively a network of interconnected grids. Each grid is operating behind strong national borders each of which operates according to its own laws and regulations. At one end of the spectrum this may have to do with constitutional obligations regarding the operation of the market (In some EU countries power supply is operated under government owned monopolies) where changes are decided by the government itself. It is not however only a matter of laws and regulations.

Electric power has strategic importance and as such failures in a national electric power system are a matter of national security.

The regulations regarding security, as well as security prevention are also set and dependant upon government regulators. The degree of security applied and the security prevention regulations that are in place are based and estimated upon the degree of national threat. Since there is such instability in terms of regulators, market policies and national laws it is clear that the network of interconnected grids is a network of incompatible and independently operated national electric power systems. More specifically the incompatibilities relate to national laws which govern the market conditions which in turn determine the environment (monopolistic as opposed to open market). Regulations are nationally determined without an overall SoS wide policy determining the operation of the SoS. So the question that needs to be answered is? Who has control of the SoS in the case of EXaMINE? Could an EU wide regulatory body prevail and set the standards for operation as well as security and security prevention?

Assume that EU forced participating countries to accelerate the replacement of laws and regulations to create an EU wide grid for power distribution that complies with a particular structural context. On the other hand EU does not have legal power to enforce a common way of organizing the whole business. For reference in EU there are as many regulators as there are countries and about as many different regulations as regulators [EX20002b] So another failure as such arises from the fact that separate laws govern each national power system. It is also a question of authority which we have raised on DSoS in the past. Can we provide a SoS dependably where control is distributed?

An EU regulatory body which would effectively manage the SoS would have to be developed by all participating systems. Unlike the TA case study where the SoS (owner of the emerging service) was the regulator in such a system as EPS the authority and legitimacy of the body would be questioned.

Let us draw some parallels.

In the TA case study the SoS we can set requirements regarding the types of systems to participate. In EPS an EU regulatory body could not possibly set requirements regarding political scene (given the diversity between western and eastern Europe) of each participating country. So in the first case authority can remain central. In the case of EPS this is a reason for dispute and consequent "failure". So the property of availability becomes primarily a political issue.

Consider the point of diversity. On the travel agent case study we used a simple algorithm to divert to a similar service when a component system was not functioning. In EPS diversity (i.e. the continuity of service property) becomes a political rather than a technical issue. Interchanges of power would have to be carefully thoughtout and operated within a legal domain. In fact the "algorithm" that would carry out such a process would have to comply with international law.

The SoS TA can make use of new components as long as it adheres to its regulations. In EPS new participants would have to have SoS wide agreement. What happens however when a participant becomes a political enemy?

Consider the case of national security. Some of the components of such a vast network are of strategic importance. Geopolitical position of these components will also cause disputes as they raise the issue of "power" distribution of the network.

Information exchange is another important issue is such a vast system. The model of networks is essentially formed by a series of national mathematical models. These include all electrical components such as generators, lines and transformers. Some of the practical problems to be solved are that the model (as a whole) is too large to compute [EX2002b]. If neighbouring networks must be modeled in each country, this will require a detailed and updated information of each network and this will inevitably have an impact to that country's own network. A large part of the algorithms that carry out the analyses have been patented and are therefore commercially sensitive. The input data is obtained via the results of the electric market and according to bilateral contracts among agents (generating and consumers). Input is also provided by voltage magnitude in different substations, active and reactive power of each station and network topology. The information needed as input to the process is located in each national system and market operators. The quality of the information can vary, because not all countries have the same rgulation in the market resolution or the information is (or became) confidential.

Given an EU wide system for power distribution one can conclude the need for a vast amount of information exchange between the various grids. Some of the information may relate to usage on certain times, days etc. In a number of EU countries (specially where EPS is operated under a monopoly) information such as statistics and statistical analyses are confidential information. This may not only be a way to hinder possible market expansion (or evolution to a more competitive environment) but because of national security (terrorism). This also raises the issue of trust within the system.

On the other hand moving to an EU wide competitive market information exchange may also be hindered on the grounds of competitive advantage.

At this point one would have to think about diplomatic disputes, national disagreements on certain policies, politics and even cases of war. Political pressure will tend to reduce freedom to exchange information and generally hinder decision making in special cases where issues such as serious disputes or war arise. Again the question of power plays an important role.

In terms of security such a vast network would require a number of monitoring tools. The tools needed to monitor the security of such as network must be reliable with an increasing need of information. Information however in competitive markets is normally economically and commercially sensitive. So open access to information cannot be guaranteed.

8.4 What have we learned?

Looking into the EXaMINE case study it is evident that a number of factors can play an important role in the way a service is delivered. The conclusions we want to draw by looking at these two diverse case studies is that since political thinking which implies cultural aspects, goals, long term strategies and short term objectives are embedded into a computer system then computer systems can be in conflict when delivering a common service. New architectures (e.g. dot.NET) seem to be pushing forward the idea of cross language, cross platform connectivity of modular pieces of software that reside behind organisational boundaries. Indeed the security issue may not be as strong as in the case of EXaMINE, nevertheless as long as software reflects the objectives of its owner or regulator there will be disputes which may lead to failures. Can this be prevented? We need to take a new look at the conceptual level, and explore the problem space that is created in more detail. We are moving from centralised to distributed control, and as such we need to acknowledge the need to recognise systems or components not only as entities of a process a function or a service but as responsibility holders with a role to play within a domain of distributed process and distributed control. We will look into these issues in the final deliverable.

9. Scoping Organisational Failures

A major scoping issue for DSoS is the decision between the following choices:

a) Are we adapting the SoS to the way its participants work

b) Are we adapting a Participant to the SoS's scope?

c) Are we adapting the SoS to its customer base?

Each question has different implications regarding the architecture of the SoS and the type of service is provides. Adapting to a particular participant implies that the TA does not need to define a scope as it would act as an extension of an existing service. The scope and operation policies would have already been defined and therefore the effort should indeed be concentrated on transferring data from the source to the SoS and finally to the customers.

If we assume that the airline systems are adapting to the scope of the TA then we need wrapping mechanisms that will resolve incompatibilities regarding protocol differences, and operation policies.

Adapting to the customer base would require the maintenance of records that capture user requirements. Furthermore regular assessment of these would also be necessary in order to identify changes in the customer needs. This would also help resolve compositional issues when selecting particular services. There are many important questions which need to be resolved before we develop a delivery mechanism and we start defining the scope of the SoS. Although we want to maintain the brokerage model of operation we do not want to completely hide the services out of which we compose the emerging service. The implication of this is that we need to develop an intelligent brokerage mechanism that will not only pass information obtained by several systems to the customer base, but will also advise customers with regards to many issues mentioned earlier.

The TA does not necessarily need to hide completely the component system from the user's view. In fact providing the user with the model of responsibilities (who sells what, who handles what) would allow some of the issues raised by the inability of the component systems to talk to the TA to be resolved. The TA also needs to maintain an exception model from each component system in order to advise the user regarding exceptions being thrown or mask these exceptions by taking action (diverse; use other component systems) depending on whether a component system is busy or offline.

Although option (c) above would seem in many ways to be the most desirable, we have identified a number of additional information structures that are needed to support the travel agent one of whose uses is to assist in the recovery from organizational failure. Examples of these are to be found in

the *catalogue of offers*, which needs to include information concerning aspects of the service providers' policies and market scope, and strategies for dealing with exceptions thrown by the service providers' systems

the *register of bookings,* which needs to include an access mechanism to be invoked when a supplier changes the terms and conditions or quality of service offered the *customer reports*, which needs to be consulted when enquiries are made as to the suitability of service suppliers

the *(emergency) advisory system*, which deals with important notifications affecting the holiday and offers assistance in rescheduling and other forward recovery procedures.

The fact that additional data structures are needed to cope with failure is an instance of a more general characteristic of organisational failure, which is that recovery from it cannot be brought into a framework offered by current approaches to recovery of a concurrent and composed technical system, because the relationships between organisational scope and policy on the one hand and organisational state, process and behaviour (particularly in the presence of failure) on the other, simply cannot be expressed in the purely behavioural concepts adopted by such approaches. Although such approaches may be systemically desirable, they may not be organisationally feasible (because of organisational policy for example). An example of this is a low-cost airline which takes no responsibility for assisting passengers who miss a connection due to late arrival of the inbound flight, even if they operate both the incoming and outgoing flights. Under these circumstances, company policy is that travellers are simply told to make their own alternative arrangements, which may involve recourse back to the travel agent and/or a travel insurance company who may not have been a party to the original transaction.

The relationships just mentioned need to be maintained by the owner of the DSoS, since they are an emergent feature of the composition of systems. We have some ideas of what a systematic approach to strategies for managing (i.e. recovery from or compensating for) organisational failure in component systems over which the DSoS owner has no direct control could involve, and propose to report on this in Deliverable DSC3 "Dependability in Multiple Domains of Management" (due at the end of the project)

10. Further Reading

The following texts have provided ideas for this deliverable, though sometimes the connection is pretty indirect:

Beer, S. *Brain of the Firm*, Wiley, 1981.

de la Billiere, P. *Looking for Trouble*, HarperCollins, 1995.

Star, S.L.. *The structure of ill-structured solutions: boundary objects and heterogeneous distributed problem solving*. In L. Gasser & M. N. Huhns (Eds.), Distributed Artificial intelligence. (pp. 37-54), Morgan Kaufman, 1989.

Weick, K.E, Making Sense of the Organization, Blackwell, 2000.

11. References

[EX2002a] Ferrante A, Diu A Needs Expression :Revised Version. Technical Deliverable. EXaMINE Project (IST-2000-26116). May 2002

[EX2002b] Moyano D, Durand M, Lopez C, Invernizzi M, Belletinni C 2002, Functional Specifications, Technical Deliverable. EXaMINE Project (IST-2000-26116). July 2002

[RA1964] Ross W Ashby, Introduction to Cybernetics, Routledge Kegan & Paul, 1964

[JCLDep] Dependability: Basic Concepts and Terminology: A Glossary in English, French, German, Italian and Japanese (Dependable Computing and Fault-tolerant Systems), Springer-Verlag Vienna; ISBN: 3211822968

[DMS3] Periorellis & Dobson 2001 P. Periorellis, J.E. Dobson. Case Study Problem Analysis. The Travel Agency Problem. Technical Deliverable. Dependable Systems of Systems Project (IST-1999-11585). University of Newcastle upon Tyne. UK. 37 p. 2001. www.newcastle.research.ec.org/dsos/