Distributed and Network Computing in the Information Society
Tempus project
(S-JEP 12495-97)

DISTRIBUTED BUSINESS AND ADMINISTRATION INFORMATION SYSTEM
subproject

Distributed business information system development methods a management perspective

PROF. DR. PETER LÔRINCZ

KANDÓ KÁLMÁN TECHNICAL COLLEGE
INSTITUTE OF COMPUTER TECHNOLOGY
HUNGARY - SZÉKESFEHÉRVÁR

AUGUST, 1998

WESTMINSTER UNIVERSITY
LONDON

ACKNOWLEDGEMENTS

We would like to express our gratitude to the staff of the Centre for Parallel Processing Department of Westminster University (London UK) who have contributed to preparation of that teaching material, and personally to ***Dr Steve C. Winter*** *Head of Cavendish School of Computer Science and to* ***Nicholas Lambrou*** *Principal Lecture (School of Computer Science and System Engineering).*

CONTENT

Acknowledgement

1. Organisation challenges and opportunities
- *1.1. Driving forces in 1990's*
- *1.2. Organisations in the 1990's*
- *1.3. Information Intensive Organisation model*

2. Distributed information system
- *2.1. Basic conceptual elements*
- *2.2. Distributed information system and technology*
  - *2.2.1. Degree of co-ordination*
  - *2.2.2. Component of distributed information system*
  - *2.2.3. Distributed transparency*
  - *2.2.4. Advantage and disadvantage of distributed information system*

3. Distributed information system development
- *3.1. Our perspective*
  - *3.1.1. System thinking*
  - *3.1.2. System developer types*
  - *3.1.3. Conclusion = our perspective*
- *3.2. Development methods*
  - *3.2.1. Open development processing: the engineering viewpoint*
    - *3.2.1.1. Open distributed model*
    - *3.2.1.2. The alignment the data flow and object oriented approaches to ODP reference model*
    - *3.2.1.3. The link model*
    - *3.2.1.4. Application architectures*
  - *3.2.2. 3-schema Specification Architecture*
    - *3.2.2.1. Approach*
    - *3.2.2.2. Design principles*
    - *3.2.2.3. Decision specific to DIS*
- *3.3. Management decision criteria*

References
Further reading
Further development opportunities
Appendixes
- *Appendix A Beers Viable System model*
- *Appendix B ANSI SPARC database model*

1. *Organisation challenges and opportunities*

1.1. Driving forces in 1990's

In order that we understand the evolution of the business in the information age we should see the effects so called driving forces influenced the organisations. A part of that forces are originated in the social, economical, and technical environment surrounding the firms and any other inside of the organisation. We focus our attentions primarily on the business world which are affected most dramatically the business organisations by the turbulent and changing environment. Doing so we stress the validity of the contingency theory.

External forces

The market:

- *uncertain business environment*
- *complex environment*
- *globalisation: global market (products and financial assets)*
- *turbulent market (new entrants, rapid product changes)*
- *the "rule of the game" are ever-changing*
- *new entrants operating under different cost structures and cultures*
- *Aveni, R. : terms this competitive environment hypercompetition*[1]
- *rapid product obsolescence*
- *quality product*
- *product differentiation's*
- *premium price for the "early to market" or "time-to-market" actions ("time based competition")*
- *low price*
- *increasing intelligence level of the employees*
- *strong civil values in the social environment*
- *computer and information literate employees*

- ***developing IT***
  - *information as a corporate resources*
  - *shared information resources*

Otherwise we should see changes inside of the organisations. Most of them are not really can be considerate as forces, they are some reactions to the effects coming from external environment. The management realising the

[1] The frequency, boldness and aggressiveness of movement of the industry players creates a constant state of disequilibrium and change. Market stability is threatened by short product life, short product design cycles, new technologies, frquent entry by unexpected outsiders. In other words **environment escalate toward higher levels of uncertainty, dynamism, heterogeneity. (Aveni, R.)**

chaining response in some way, which details can be seen in the following sections.

.

Internal forces - management reactions

- *organisation are downsizing (flat or lean organisation syndrome)*
    - *eliminating of middle management position*
    - *worker empowerment*
        - *knowledge and responsibility are available to the employee*
        - *decision and responsibility are shifting to the operation level*
- *process oriented management paradigm (re-engineering the business processes)*
- *mission critical applications*
- *IT technology*
- *strategic thinking and strategic management*
- *geographically separated organisation*
- *multi divisional business organisation*
- *business culture as a co-ordination force*

1.2. Organisations in the 1990's

Organisations should be or are being in 1990s:

- *adaptive*
- *flat*
- *fast*
- *flexible*
- *focused on areas of core competency*
- *empowered*
- *interfunctional teams*
- *information intensive*
- *forming*
    - *partnerships*
    - *strategic alliances*
- *expanding organisational*
    - *capabilities*
    - *scale*
    - *scope*

1.3. Information Intensive Organisation model

The managers dilemmas and challenges:

- *to manage both **speed** and **complexity** simultaneously in the organisation*

  Percy Barnevik, CEO of Asea Brown Boveri (ABB):
  *"ABB is an organisation with three internal **contradictions.** We want to be global and local, big and small, and radically decentralised with centralised reporting and control. If we resolve those contradictions, we create organisational advantage."*
- *how can take firms advantages of the information management and communication tools to handle the management problems ?*
- *define*
  - *the information that needs to be in the systems*
  - *the meaning of the information*
  - *how to use to co-ordinate and manage the business*
- *respond more quickly and effectively to changes in the business environment*

By the field research of M. L. Applegate in the US. the managers produced a comprehensive approach[2]

they altered in the firms

- *the strategy*
- *the structure*
- *the operating and management processes*
- *the distribution of power and authority*

they need to provide

- *more flexible, networked information infrastructure to manage the corporate information*

***The solution** of the management dilemmas is to define and to use new organisational model:*

Information Intensive Organisation

Creating the Information Intensive Organisation

- *demands **comprehensive approach***
  - ***to strategy development***
  - ***to organisational design***
  - ***to performance monitoring***
- *that will enable the firm to achieve third commitment*
  - ***to continuous learning***
  - ***flexible adaptation***

[2] by a field research Of M. Lynda Applegate

- ***flawless execution***

Lessons *for the* ***design*** *of the Information Intensive Organisation:*

- ***Speed counts****, but not at the expense of control*
  - *the right decision must be made quickly*

- ***Empowerment[3] is not anarchy !***
  - *to avoid the disaster by a more comprehensive redefinition of authority and control throughout the organisation*
  - *the organisational must be*
    - *more clearly communicated*
    - *closely monitored*
- ***Transforming*** *an organisation requires* ***more then*** *just change the structure*

Requirements:

- *the managers adopt a* ***comprehensive approach*** *to organisational change*
- *rethink the nature the control an* ***authority***
- *information processing demands increase dramatically as organisation attempt to cope with greater complexity, uncertainty and speed change*

A basic rule:

> ***"****Transforming the firms* ***information infrastructure is a critical component*** *that supports and enables the organisational transformation." [5]*

Flexibility vs. complexity

- *cycletime of business is determined by:*
  - ***operating processes*** *series of activities that define how a firm designs, produces, distributes, markets, sells and supports its product and services*
  - ***management process*** *are those activities through which a firm defines strategic direction and co-ordinates and controls the operating processes that execute strategy.*

which are ***tightly integrated***

- *effort to increase the speed and complexity of core operating processes without increasing the speed and level of detailed* ***information available*** *to manage those processes lead to a disaster*
- *to design a firm that optimises flexibility requires:*
  - *a shift in the nature of control*
    - ***from the standardisation and supervisions***

[3] push decision making down the line

- ***to learning model*** [4] *(see learning organisation and organisation learning paradigm)*
- ***system thinking,*** *the ability to see a situation as a whole is at a* ***heart of learning***[5]
- ***to integrate and continuously improve*** *the processes*[6]
  - *to that* ***detailed and timely information*** *is essential*
    - *to* ***co-ordinate the flow*** *of activities*
    - *to provide to understanding the business* ***dynamics***
- *management processes must be*
  - ***streamlined***
  - ***integrated***
  - ***time-synchronised***

with operating processes.

- *the information must be*
  - ***widely shared***
  - ***fact-based***

"Managers and decision makers at all levels of the firm must be able to access information directly and must possess the skills and experience to extract meaning that guides effective action an decision." ***(M. L. Applegate)***

An other contradiction: independence and interdependence

- *the formal and informal structures, co-ordinating mechanisms, responsibilities of an organisation define the* ***distribution of power***[7] ***and accountability*** *within the organisation (firm)*
  - ***traditionally*** *the formal distribution of authority is a trade-off between centralisation and decentralisation:*
  - ***at the increased organisational complexity*** *the decision making should be decentralised to the point at which individuals possessed the information*

---

[4] discussion of learning organisation paradigm is out of scope our teaching material

[5] it demands the understanding the **causal relationship** between the elements of the system and whole

- it implies that learning model of control requires the development the **shared purpose (direction)** and **shared understanding the individual actions**

the learning process : the managers continuously **monitor the environment** and the organisational **capabilities and performance,** seek to **recognise** when a particular strategy **does not meet** the demands of the environment or the **nature the organisation,** and is ready and able **to define and implement** a more appropriate **strategy, structure or processes.**

[6] see BPR method

[7] more information in Do. Dahlbom and lrs Mathissen Computers in Context, The Philosophy ant Practice of System Design, Blackwell

- ***information intensive organisation*** *In that case we must rethink the nature of the authority. It is not simple linear trade-off between autonomy and control.* ***Teams[8] of employees representing the range of expertise and accountability*** *required to make decision*
    - ***shared incentives*** *reinforce* ***shared authority***

The Impact of IT on work and people[9]

Information technology:

- *improve* ***process efficiency*** *and* ***productivity*** *and substitute labour*
- *to enable people to* ***better understand the dynamics*** *of the business, they generate new insights and* ***new knowledge****,*
- *at all levels work become* ***more analytical*** *and require more judgement and experts*
- *push authority for decision making from corporate headquarters to the fields*
- *The role of the factory employees is changing: less time manually operating, more authority for co-ordinating production*
- *IT has power to* ***integrate work,*** *(job enrichment)*
- *IT can be used to integrate work both* ***vertically*** *and* ***horizontally (laterally)***
- *IT connect people and work across* ***time and space,*** *distributed information and communication system allow the formation of* ***"virtual team", "virtual organisation", "virtual factory[10]"***

Some generic organisation system characteristic *should we regard at system development:*

- *every organisation is originally*
    - *horizontally (functionally)*
    - *vertically differentiated*
    - *geographically dispersed*

[8]Common examples of team - based organisational units:

- **operating team** composed of middle managers from manufacturing, logistics, marketing etc. This team usually is located on middle management level
- **work team** interfunctional teams of line employees for defining, executing and managing work
- **process management team** is created to co-ordinate, control, and continuously improve operating processes

[9] M. L: Applegate Designing and Managing the Infromation Age Organisatin
Harvard Business School 196-003

[10] it is out of scope this material

2. Distributed information system

2.1. Basic conceptual elements

- ***Information technology (IT)*** *defined as the range of electronic technologies for collecting, processing, distributing data and consist of computers, communication technologies*

- ***IT infrastructure*** *is a set of IT components designed and implemented to satisfy the* ***information needs of specific communities of users.*** *It comprises the elements that make up* ***the physical system*** *supporting data collection, processing and presentation.*
- ***IT*** *can be described using* ***logical and physical view.***
  The logical view *describes it as* ***co-operating processes*** *that manipulate data and present information to users of IS.*
  The physical view *consist of network equipment , computers, operating system, DBMS, application SW,*

- ***Distributed IT infrastructure is an IT infrastructure incorporating multiple autonomous processors that do not share local memory, but co-operate by sending data messages over an information network. The IT infrastructure exhibits distribution transparency and its attributes:***
  - ***Multiple autonomous processing elements***
  - ***Information exchange over an information network*** *connects the autonomous processing elements*
  - ***Processes interacts via non-shared local memory***
  - ***Distributed IT infrastructure*** *exhibits*

2.2. Distributed information system and technology

There is not universally accepted definition of distributed information system, it can be characterised by four ***principal attributes:***

- *it is implemented over a distributed IT infrastructure*
  giving a system designer and user a freedom to optimise the placement of distributed system components:
  - data
  - processing
- *it can access and intercorporated a variety of media types*
- *it processes data from a variety of media sources in an integrated manner. A physical view of processing consist of a web of co-operating processing objects and related data objects which may be distributed across the IT infrastructure*
- *provides a high level of user interaction and control*

- *to allow to shearing of information and resources over a wide geographic area*

2.2.1. Degree of co-ordination

*What do we mean by a "**distributed system"?** The problem is not so much what distributed means - **but what system means in this context.***

- *When activities in different location are co-operating with each other , in what circumstances should they be regarded as a single system?*
- *On in what circumstances should we regard them as distinct systems with a capability of **communicating with each other ?***

> ***An important characteristic** in determining whether a system is **distributed** is **the degree of co-ordination** between activities in different location.*

Degree of co-ordination

- ***Tightly-co-ordinated activities*** *we expect that what happens in one activity should be consistent with what happens in other activity, when viewed by an external observer independently of where those activities are located. we take a **set of tightly - co-ordinated activities to be a single system.***
  - ***Benefits** of developing a system of tightly - co-ordinated activities are:*
    - *we produce a system that implements the users business requirements in the **most coherent way***
    - *we eliminate all problems of inconsistency between locations, provided that the hardware and communication system operate fault-free*
  - *A tightly co-ordinated system **costs more** and **takes longer to develop***

- ***Uncoordinated activities** we mean that locations can inform each other about what are doing, but each location are free to take notice of others communication. There are the **separated systems.**[11]*
  - ***Benefits** of this approach are that activities can be designed and implemented separately* [12]

[11] In this volume we don't regard this as a distributed system

[12] As a part of the development of distributed system analysis of existing uncoordinated systems my be necessary. Integration of separately - developed systems could be regarded as a distributed system problems but it is outside the scope of this volume. It is addressed for example in the ISE Library Volume: Application Partitioning and Integration with SSADM.

- ***Loosely - co-ordinated activities*** *allows parts of the system to become* ***temporarily inconsistent.*** *We take a set of loosely - co-ordinated activities to be a single distributed system.*[13]
- *With respect to the relaxed business rules the system is still tightly-co-ordinated. But we need two levels of description:*
    - *the description of the semantics of the desired tightly - co-ordinated system*
    - *the description of the relaxed semantics of the loosely-co-ordinated one to be implemented*

We also need a ***statement of the relationship*** *between two. This should include* ***external processing constraints*** *to be applied to make the relaxed semantics approximate the tightly-co-ordinated semantics.*[14]

- ***Benefits*** *of developing a system which loosely-co-ordinated activities are:*
- *we produce a system that implements an acceptable compromise between the most desirable semantics of the users business and the achievement of an acceptable level of fault tolerant operation.*
    - *we eliminate problem of inconsistency between locations*
- ***The risk*** *is that analysis and design is more complex because we have to define:*
    - *the tightly-co-ordinated semantics*
    - *the relaxed semantics* [15]
    - *the relationship between two*

2.2.2. Component of Distributed Information System

Components:

- ***data*** *may consist of many data object, representing many media types*

- ***processing*** *is concerned with processing data object. It may contain processing objects such: viewers, browses, search engine, application-specific logic, business logic,*

[13] The design of such system is within the scope of this volume.
[14] For example although it is possible under the relaxed semantics to accept a reservation without checking customer status there should be a constraint that if it is possible to check the customer status then it should be done.
[15] usually more complex than the tightly-coordinated semantics

- ***presentation*** *is processing directly concerned with making data visible to users and handling users interactions.*
  - ***user interface*** *The aspects the user interface that he user physically perceives.*
  - ***presentation management*** *that provides the basic operation to build and control the user interface under application. The key functions:*
    - ***display service*** *manage all user interface devices*
    - ***dialogue control*** *provides mechanism for creating an manipulating the user interface and event handling functions.*
    - *application programming interface (**API**)*

In addition four

- ***services*** *are required which are normally assumed to be incorporated into the distributed IT infrastructure:*
  - ***separation mechanism***
  - *mechanism for **identification** of the target object*
  - ***administration and management*** *of the overall system*
  - ***security***

Classes of distributed support services

*Because many application require the **same high-level services** in the IT infrastructure much of functionality required is offered as **services as an integral part of the distributed infrastructure.***

Classes:

- ***distributed presentation services***
- ***distributed processing services*** *in that case the DIS consists of co-operating processing objects distributed across client and server elements of IT infrastructure*
- ***remote data access services*** *in this configuration the presentation and processing components reside on the client . The data components reside on one or more servers using database management system software*
- ***remote file access services***
- ***distributed data management services***
- ***distributed object management services***[16]

[16] An object is the data and associated operations are closely coupled, An object represents a a self contained unit which encapsulated data and operations (usually known as methodes. Objects inteacts with each others using some form of message passing. An example of an architecture based on a distributed object model: the CORBA.

2.2.3. Distribution Transparency

Moving to distributed IT infrastructure enhances flexibility but increases complexity
users are demanding a simple user interface to this complex world for concealing complexity
users desire ***a single system view***

- ***Distribution transparency:*** *describes the visibility of distributed components within a DIS*
- *Transparency choices:*
  - ***full distributed transparency***[17] *if all of transparency functions are built into the IT infrastructure*
  - ***partial or selective distribution transparency***

- ***Transparency functions (ANSA 1993, ISO 10746-3)***
  - ***access transparency*** *hiding the use of communication to access of remote resources*
  - ***location transparency****users do not have to know the location of resources. Resources are accessed by logical address or name*
  - ***concurrence transparency*** *the user is unaware of the existence of parallel access to remote resources*
  - ***migration transparency replication transparency*** *the potential to relocate resources dynamically without users being aware of reconfiguration*[18]
  - ***partition transparency*** *hiding any difference between the replicated and non-replicated resources*
  - ***persistence transparency*** *hiding the difference the partitioned and non - partitioned resources or service*
  - ***failure transparency*** *hiding the effect of partially completed interactions that fail (partial failures)*
  - ***performance transparency*** *minimising the performance overheads in using remote resource, so that response times and throughput are comparable with the case of local resources*
  - ***scaling transparency*** *is hiding variations in system behaviour due to arbitrary scaling.*

[17] generic distribution functions are implemented at the operating system level
[18] IT continually change due to growth, enhancement, and reconfiguration for fault avoidance and load balancing.

2.2.4. Advantage and disadvantage of DIS

Advantage

- ***Improved flexibility***
  IT infrastructure components can be flexibly located at points within organisation where they can be utilised most effectively. Components can be
  - added
  - upgraded
  - moved and removed
  without impacting upon other components
  scalability
- ***Local autonomy***
  - *by allowing **domains of control** to defined where decision are made*
  - *Each domains decides where resources are located*
- ***Increased reliability and availability***
  - *in a distributed system multiple components of the same time are configured and provide a level of fault tolerance*
- ***Improved performance***
  - *a service which is partitioned over many server computers a smaller set of application and users with access to a local data and resources result a faster access (response time)*
- ***Security breaches are localised***

Disadvantage

- ***Difficulties in management and security***
  - *DS require more complex procedure for security , administration, maintenance and user support*
- ***Reduced reliability and availability***
  distributed system consist of many more components which potentially fail, causing loss or availability
- ***Shortage** of support and development staff*
- *Increasing **complexity***

3. Distributed information system (DIS) development

3.1. Our perspective

3.1.1. System thinking types [10]

*System developers and analysts need a **framework** in order to successfully organise their own work A framework is consist of - in generally speaking- **theories, methods and concepts.** To understand and to make an assessment of the different development and analyse methods we accepted an **system approach** elaborated by Dahlborn and Mathiassen. There are three different system approaches:*

- ***hard system thinking***
- ***soft system thinking***
- ***dialectical system thinking.***

*These approaches differ in the way they support the **understanding and designing** the system. **The characteristics** of the approaches:*

- ***Hard system thinking***
  - *from the hard system thinking point of view the system is **hierarchically organised***
  - *the emphasis is on the **internal structure of the system***
  - *the system is equal of **functional system** (hierarchically organised structure of interrelated functions)*
  - *the system behaves as **determined** object, like a machine*
  - *the central point of the system design an analyse is the **functional analysis and design** (the classical approach of the programming and system development)*
  - *the hard system thinker proceed that real world (the reality) is itself **an ordered, stable system** (the reality has abovo system characteristics) and the task of the system designer is no more than to represent the **entities, relationships among them and the dynamic behaviour of them***
  - *in that approach the system thinker (user, or designer) is an **outsider who observe the reality (**or part of it)*

- ***Soft system thinking***
  - *by the soft system thinkers the system is the result of our **experience** of the reality*
  - *the system thinking a way to deal with a real world being diverse, and constantly changing*
  - *the different observer (user and designer) has a different **perspective** (the sum of the perceptions or the observers) depending on**: interests, culture, etc.***

- *the methodology of the soft system approach is based on **the helping the designer to understand** these perspectives (For example: P.Checklands Soft System Methodology)*
- *The different perspectives force the participant to debate, to communicate with each other to reach some sort of **agreement** formulated in the **system requirements.***
- *for the soft system thinker the system development seems a **learning process** resulting system requirements as a system perspective This perspective is communicated, negotiated with the purpose of reaching common understand and **common interest in change***
- *the designer is **part of the system** (he or she is an **insider)** whose task is to involve the users (potential users) into the analyse and design process, to help them to be participants of the learning process*

- ***Dialectic system thinking***
  - *based on the idea that the world is ever changing*
  - *the dialectic system thinker are thinking in terms of **contradictions** in order to explain the changes*
  - *is an dialectic **syntheses** of the others two*
  - *it takes both reality and perspectives, matter and mind etc., but **it places all this in the context of contradictions , interests and struggle for power***
  - *the dialectic system thinker is outsider and insider in the same time, and combine the two roles*
  - *the goal of system analyses is to get information of contradictions*
  - *he/she is the **change agent***

*In our work we **prefer** and propose to prefer the **dialectic system thinking** as a philosophical background of system analyse and design procedure because*

- *the approach deal with the technical and social aspects of the organisational systems*
- *it emphasise the organisations unstable and unordered, ever-changing being*
- *it tuns the designers, users and the designers attention to the well known fact that the consensus in the organisation is only temporary and the contradictions may and should be resolved by actions*
- *the information system as a system of the special organisation asset, the information is a reach area of the contradictions (for example the struggle for power) and the computer system is a powerful tool to resolve or maintain that conflicts.*

3.1.2. The system developer types[19]

[19]B. Dahlbom, Lars Mathiassen, Computers in Context

- ***Computer expert***
  - *offering a technical expertise*
  - *they know about the computers and how to build a computer system*
  - *the attitude is humble and comfortable*
  - *their expertise is limited*

- ***Sociotechnical experts***
  - *believe that taking social issues into account will they be able to develop good information system and get satisfied users*
  - *the social issues are as important as technical ones*
  - *they use sociotechnical methods of social and organisational analysis*
  - *they have to analyse and appreciate the attitudes and expectation of the future users*
  - *a sociotechnical approach requires the active participation of the users*
  - *the attitude is concerned and co-operative*

- ***Political agents***
  - *they argue that as a professionals we have to commit ourselves and answer a fundamental question: Who really is the user ?*
  - *depending on the answer the world will look very different*
  - *they cannot say good without asking, good for whom?*
  - *the evaluation of computers in use is always relative to one or another interest*

3.1.3. Conclusion = our perspective

On the base of the theory of information system development methods, with a background of many year long industrial application development practice and the teaching experience gained in the higher education, in this teaching material we are presenting some aspects of the development methods of distributed systems with the approach consisting the following key points:

- *the system developer is a **system thinker** (following the system thinking rules) and analyse and develop a system in a dialectical way*
- *the system developer is **sociotechnical expert** or **political agent***
- *the system design process is maintained as **intervention** (in B. Dahlboms manner)*

3.2 Development methods

Introduction

There is a wide range of methodologies for IS development. The common characteristics is the ***need to model three orthogonal views*** *of system,*

- ***data modelling,*** *model the static structure of the system,*
- ***functional modelling.*** *concerning with the data processing in the system,*
- ***entity life history dynamic*** *modelling, which is concerned with modelling the* ***time-dependent behaviour of the system***

Most methodologies also adopt a ***life-cycle approach*** *encompasses separation of*

- ***logical***
- ***physical view***

Structured methodologies

A criticism of many current development methodologies is the ***implicit assumption that the underlying IT infrastructure is centralised.*** *The modelling techniques are de-emphasising* ***distribution during logical design,*** *but ignoring distribution totally* ***is not helpful*** *and modelling techniques are being extended to consider issues such as:*

- ***component placement***
- ***distribution transparency***
- ***multimedia aspects***

In the physical view the logical design will need to be transformed into candidate physical design.

Some international aspects:

- *there is a rapidly - increasing desire for information systems which* ***range over very wide areas.*** *The EU and single market spur demand for system allowing Europe wide exchange and administration of information systems e.g. EURES [20] and EDIS[21]. Each of these systems requires a considerable number of* ***co-operating data processing systems.***
  - *the development* ***of Euromethod[22]***
- *Currently* ***island solutions are connected by ad-hoc approaches.***

[20] European Employment IS

[21] EDI for Social domain

[22] Euromethod: The progaram developing an fremework for the EU countries has evolved aproposal made in . 1989. The method being as an information development standard of EU is concerned with defining the deliverables to be exchanged between customer and supplier during all aspects of system developments, Seven method were selected as the basis for development of Euromethod: SSADM(UK), MERISE(France), SDM (Netherlands), DAFNE(Italy), Vorgehensmodell(Germany), MEIN(Spain) and Information Egineering. The first version was delivered in 1994.

Specialities of the development of DIS

- ***Issues in distributed systems development*** *there are three key areas, where distributed systems development is different from development of uncoordinated systems*
  - ***Consistency and functionality vs. robustness*** *Distribution provides a way of making remote locations less dependent for their operation on the availability of a system at a central location. But transactions that require data at several location are still compromised if one of the involved locations is out of communications. Indeed the risks of failure for some transactions are likely to be higher if more than two locations are involved. The probabilities of failure are* ***multiplicative,*** *not additive. But what if the* ***dominant or important transaction are multi location? There are several solution:***[23]
  - *It may be possible to reorganise the location of data so that the transactions become* ***single location transaction.***
    - *or so that data is available in several location*
    - *It may still be possible to* ***split a transaction*** *into several pieces, one of which can be done locally*[24]
  - ***Visibility of geographical distribution facts in requirements and code****It should be possible with* ***modern systems development methods*** *to buffer the geographical distribution decision from most of the rest of the system description.*[25]
    - ***Technical architecture and platforms*** *The logical design approach used in this volume generally assumes a suitable implementation platform that provides capabilities for:*
      - *reliable communication between locations*
      - *distributed commit management (including deadlock resolution)*

General design principles for distribution

We can identify three principal techniques used to implement ***scaleable, dependable services:***

- ***replication***
- ***partitioning***
- ***caching***

Replication

[23] There a trade-off to be made. The only person who can decide is the customer of the system. To this extent at last **distribution issues must be raised early system requirements analysis process.**

[24] The remote pieces either need not be done at all (providing a degraded service) or may be done later, making the overall system temporarily inconsistent.

[25] Our assumption in this volume is that such a separation of concerns is desirable and achievable.

*A service is replicated when multiple identical instances (known as **replica**) of the service appear on different nodes(server) in a distributed IT infrastructure. Each replica holds an instance of the shared system state. Replicas that holds the most recently written state data are said to hold **authoritative data.** Ideally a client request can be directed to any replica and all replicas have consistency of state*

*The placement of replicas is another important issue and depends on whether replication is primarily to **improve performance** or to **improve availability.***

***Replica control mechanism** are required to ensure that replicas continue to hold authoritative data. **Strict or strong consistency techniques** ensure that all replicas hold the same authoritative state data at any given time. This is achieved:*

- *all replicas have received every shared state data update request*
- *all replicas process received requests in the same (correct) sequence*

3. 2. 1. Open distributed processing: the engineering viewpoint [1]

Successful development of DIS needs to be a radical re-think of

- ***methods,***
- ***design process***
- ***tools***

*A relatively recent development in distributed system design approaches is **ISO reference model for open distributed processing (RM-ODP).** Work firs began in 1987 and submission started in 1991 ISO 10746). ODP standard identifies several **interfaces** which standardisation may be required[26]:*

- *application program interface (API) allowing access a defined function*
- *human-computer interface*
- *interconnected and interworking standards*
- *multimedia storage and access standards to allow information exchange between systems*

ODP** adopts **the distributed object model,** because the object is a natural unit of distribution. ODP deals with the full complexity of DIS by defining **viewpoints of a system** each of which focuses on a particular **abstraction .** Each viewpoint describe a ODP system in terms of different **aspects (logical grouping) or viewpoints:

[26]Linington 1992, Introduction to the Open Distributed Processing Basic Reference Model

- ***enterprise viewpoint*** *focuses on the purpose of the system. This is the viewpoint of users, or sponsor.*
  - *how is the system used in the enterprise*
  - *relevant business and management politics*
  - *processes user roles*
  - *the enterprise models identify*
    - *organisational structure*
    - *the business activities for which an ODP system should be available*
    - *user location and user roles, tasks to be automated using DP system*
    - *security and management policies*

- ***information viewpoint*** *focus on the information aspects of the system, this is the viewpoint of*
  - *information managers*
  - *system analyst*
  - *information engineers*

  It is concern with
  - *information* ***requirement***
  - *information* ***flow***
  - ***logical partitioning*** *of the information*

  The model includes:
  - *structure of* ***data elements*** *that defines the information needs*
  - ***rules*** *stating the relationships between data elements*
  - ***data transformations*** *as a result of data processing*
  - *data* ***flows***
  - *the forms in which data and data processing are visible to users*

- ***computational viewpoint*** *focuses on the computational elements of the system. This is the viewpoint of the application developers. This is concerned:*
  - *processing*
  - *associated data structures*

  The model expresses:
  - *synchronisation between programs*
  - *inter process communication requirements*
  - *data types, the processing operations associated with each data types, These requirements are expressed in terms of* ***abstract data types or in objects.***

- ***engineering viewpoint*** *focuses on the concrete mechanism . This is concerned with* ***mapping*** *the computational model on to the distributed system. This includes*
  - *the IT infrastructure*
  - *the mechanism used to provide the various* ***transparency*** *functions*
  - *the mechanism to support efficiency in terms of*
    - ***performance optimisation***
    - ***robustness***
    - ***reliability***
    - ***and other service level requirements***
- *The model expresses*
  - *distributed IT infrastructure components requirements* [27]
  - *placement of presentation, processing and data objects*

- ***technology viewpoint*** *focuses on conformance issues in component selection and procurement The technology components include:*
  - *client and server hardware platform, operating system, I/O devices, hard disk capacity, network type and equipment ...*
  - *generic service to be employed*
  - *network configuration and components*
  - *transparency functions to be employed*

[27] the emphases is on component functionality not the supplier product.

3.2.1.2. The alignment the data flow and object oriented approaches to the ODP reference model

(Source: E. Simon, Distributed Information System, Table 10.1. pp. 295)

ODP VIEW	FOCUS	MODELLING REQU.	DATA FLOW		OBJECT ORIENTED	
			MODEL	TECHNIQUES	MODEL	TECHNIQUES
Enterprise	purpose, scope and policies for the system	Identification of the domain Federation requirements Roles played by the system Policy statement about the system			object modelling ?	
Information	Semantics of information and information processing	Static schema Dynamic schema Integrity constraints	EAR model Data model	EAR modelling	Object model	Object modelling
Computational	Functional decomposition into objects which interact at interfaces	Interactions between presentation, processing and objects of DIS	Process model	DFD Entire life history modelling	Functional model Dynamic model	Data modelling State transition modelling
Engineering	Mechanism and functions required to support distributed transactions	Object placement Transparency functions Linkage mechanism between presentation, processing and data objects of IDS	Few established models or techniques		Few established models or techniques	
Technology	Choice of technology	Generic functions for which standards and products must be sought Standards and selection and/or built Procurement, implementation and management	Few established models or techniques		Few established models or techniques	

*As we move from the enterprise viewpoint to technology viewpoint we are moving from the highest level to lowest level of **abstraction.** At highest level the real world is modelled At the lowest level we see the implementation of these objects and linkages between them.*

Aspects:

- ***storage***

- ***process***
- ***user access***
- ***separation***
- ***identification (naming and addressing)management***
- ***security***

*A modelling areas which are less developed are **engineering an technology model.** E. Simon in the book [1] in 10,4. section outlines an approach to developing a **link model** which defines an engineering viewpoint of DIS class of ODP system: **the link model..***

3.2.1.3. The link model: an engineering viewpoint

*A potentially distributed IT infrastructure, an engineering model of a DIS emphasises a view of the system as a **network of associations of six fundamental types entities:**[28]*

- ***actor object (AO)***
- ***user interface object (UIO)***
- ***processing object (PO)***
- ***data object (DO)***
- ***user agent object (UAO)***
- ***link***

Object placement: an infrastructure diagram
*In order to consider different scenarios for placement of various object types in a concrete distributed IT infrastructure, the various components of the infrastructure must be defined **An infrastructure diagram** uses entities:*

- ***a location type** classification of location instances (branch office, headquarters, ...)*
- ***location instance,** a physical location at which an object may be placed*
- ***node type** which define a particular type of node which may be placed at a location instance (client PC workstation, UNIX server ...)*
- ***node instance** instances a node type at a particular location instance*
- ***node link type** defines the connection between two nodes, describes the characteristics of the node link*
- ***node link instance** is instance a node link type (ISDN, ...)*

An distribute IT can be modelled using modelling entities. Each location, node and node link instances can be represented diagrammatically. A link model is produces from the computational model and infrastructure diagram by carrying the following steps:

[28] adapted from Domville (1992), and hackathorn (1993)

- *annotate the **computational model** indicating the types of each object (AO, UIO, PO, PO or DO)*
- *define a **placement policy** which defines the rules for the placement of various UIOs, POs, and DOs taking account:*
  - *centralisation*
  - *replication*
  - *partitioning*
  - *caching*
  - *service level policies and decision*
- *place the various object on the **infrastructure diagram** according to placement policy*
- ***simulate** the resulting **link model** using **volumetric**[29] in order to evaluate whether the link model meets functional and service level requirements*

3.2.1.4. Application architectures

One-tier application architecture

*A one - tier application architecture is typified by the traditional mainframe application. The applications are characterised by tight **integration of***

- ***presentation***
- ***processing***
- ***data components.***

Two-tier application architecture

application component s are decoupled and placed on two IT infrastructure elements: client and servers computers
This gives rise two configurations:

- ***The fat client***
- ***The fat server***

Three - tier application architectures

The presentation, processing and data components are completely decoupled and typically reside on different computer platforms:
***Tier 1** compromises client workstations each running the presentation component of an application.*
Tier 2** comprises **shared application services
***Tier 3** comprises **shared data services** which may be widely dispersed geographically.*

*Choosing the right application architecture is a **rightsizing** issue.*
The table summarises the strengths and weaknesses of each application architecture.

FEATURE	***ONE-TIER***	***TWO-TIER***	***THREE-TIER***

[29] erre vonatkozó information Breu -. !

Primary system type	*Operational /transaction processing*	*Group/departmental operational/decision support*	*Enterprise level operational/decision support*
Scalability	*low*	*low-medium*	*high*
Position in technology life cycle	*decline or re-positioning as high-end server*	*growth*	*growth*
Design and implementation flexibility	*low*	*low/medium*	*high*
Implementation complexity	*more difficult to implement*	*easy to implement*	*more difficult to implement*
Rapid application and deployment	*no*	*yes*	*possibly(depends on the application complexity)*
Security, performance and data protection	*high (with OLTP and database recovery support)*	*low-medium (dependent on use of stored procedures and transactional (SQL)*	*high (with DTP in tier2 or alternative midleware with transactional semantics)*
Ease of management	*good (due to centralisation)*	*good (small scale) poor (large scale)*	*more difficult (due to increased complexity as a result of decentralisation)*

Typical features of each category of distributed application development[30]

FEATURE	*ONE-TIER*	*TWO-TIER*	*THREE-TIER*
Scale	*mainframe application*	*Workgroup application*	*Enterprise application*
Toolkit	• *CASE tools with* • *centralised IT infrastructure assumed* • *Team development and centralised repository* • *Configuration management/ version control*	• *GUI front end and a single SQL database server development tools* • *an existing RDBMS is presumed*	• *integrated development of all three tier components* • *distributed IT infrastructure assumed* • *team development and centralised repository* • *configuration management and version control* • *decoupled presentation, processing and data components* • *flexible component placement options*
Application	*large scale operational*	*small scale decision support*	*large scale operational and decision support*
Support for building server based business logic	*mainframe based*	*very limited (normally via stored procedures)*	*yes*
Support for GUI design	*limited (dumb terminals or PC terminal emulation)*	*yes*	*yes*
Batch processing application development	*yes*	*yes*	*yes*
Linkages/middleware	*terminal emulation protocol*	*local: DDE, OLE remote: RDA and RFA*	*Local: DDE, OLE Remote: RPC, transactional. message queuing, ORB*
Design for reliability/ performance	*limited scope*	*replicated database server*	• *replicated shared application and data services* • *load balancing dynamic*

[30]Table 10.3. (pp311)

reconfiguration

3.2.2. 3-schema Specification Architecture

A good example of how to current methodologies , SSADM and the BOS Engineering Method, can be extended is found in Breu at al. (1994)

3.2.2.1. Approach

This method can be considered as some extension methods for instance the SSADM and this includes:

- *the 3-schema Specification Architecture*
- *explains how the 3-schema Specification Architecture maps on to network components*
- *describes the approach to design distributed systems*

3-schema Specification Architecture for Information Systems

*The design of a distributed systems consist of **description at different level of abstraction.** The 3-schema Specification Architecture is a general approach to **buffering** the main part of description from information about the systems **geographical distribution**. The method base on the **ANSI Sparc 3 - schema database architecture** (more information see at Appendix B.)*

What do users want from the information systems? *For most systems users require **simulation** of what is happening in their business based on **databases**. They can then use information obtained from the databases to **support decisions** that are applied to the **real world** of business.[31] Users do not want the **entirety** of their business to be simulated. Many facets of the real world are **not relevant** to the business activities.*

The scope of an IT system is determined by:

- *describing the **Business Process View (BPV),** [32]*
- *defining the **support** need **from the IT system***

*We define functional requirements by developing detail for two facets of this **support:***

- *defining, what information is needed to support each BPV process*
- *deciding whether any BPV processes can be wholly or partially **automated.***

A conceptual schema for information support

[31] Users bases **real-world** decisions on view of **simulated reality**.
[32] BPV develop is outside the scope of this volume

In IT system development we define provisions of information support by:

- *developing a **data model** within the scope of the BPV processes We **turn** the data model **into one or several databases**[33]*
- ***testing the data model** for capability of providing information support*

*The data model must be **kept up-to-date,** we need to identify **the changes in the business** that have to be reflected in the data model, we must determine **how** it can be kept up-to-date.*
We must ensure that inputs are identified for:

- *creation and removal of each entity type*
- *swapping of changeable relationships*
- *making and breaking of optional relationships*
- *updates of every changeable attribute*
- *imposing and removing constraints by setting states of entities*

*We must also ensure that we know which **actor roles** provide which inputs.*

A 3 - schema processing architecture** In addition to the conceptual schema we also have to be concerned with an **external** an **internal schema.** All three schemata lead to **code at the technology level.

External design

External schema is defined in:

- ***procedures** (from DFD and Requirements Catalogue Entries in SSADM, or Business Transactions in BOS Engineering Model, and/or ad-hoc enquiry specifications*
- ***dialogues***
- ***the batch input-output subsystem***

The external schema has three major elements:

1. ***packaging***
2. ***processing:***
   - *to convert external DP input messages to update data and enquiry triggers*
   - *to convert update and enquiry responses to update for actors*
   - *to detect and diagnose syntactic errors*
   - *to report semantic errors*
   - *to navigate between the elements of a procedure/function and between procedure / function*
3. ***a mapping** of external schema logical specification to an implementation technology*

[33] that provide a basis for the **simulation of the behaviour** of those entities and relationships

The external schema differs from the conceptual schema:

- *it is developed by design and engineering, not by discovery*[34]
- *there could be* ***several different*** *external design for single conceptual schema*[35]

Internal design *is an constructed system, and there could be* ***several internal design*** *for the same conceptual schema.*[36]

Overview of general notion of services[37]

Mapping of 3-schema Specification Architecture on IT facilities

The 3-schema Specification Architecture maps on ***three kinds of IT facilities***[38]

- ***User- Interface Servers*** *on the workstations at the business locations*
- ***Process Servers*** *at DP locations*
- ***Data Servers*** *at DP locations*

Design by the 3-schema Specification Architecture

Summary of design principles

- ***concepts*** *of distributed systems are design are defined in terms of 3-schema Specification Architecture*
- *a distributed system is a* ***co-ordinated*** *system*
- *systems that serves different location types can be co-ordinated on* ***entity*** *aspects*
- ***logical design*** *should be independent of architectures*

---

[34] there are many factors influencing the external schema: allocation of user responsibilities within the business, level of ability and training of users, constraints on technology to be used, interface style guides, arbitrary rules and performances, legislation, The designer has to make trade-off between requirements and constraints

[35] This is obviously true at the **technology level:** Motif, MS Windows, SQL, In A distributed system these could all exist **concurrently.**
But the difference could be more fundamental. The information support provided by a conceptual schema could be packaged in different way into functions, to serve different organisational structures and users roles.

[36] There are many DBMS technologies, and many ways to use their facilities to meet the **physical design objectives.**
In the **distributed systems** the same data model (or submodel) could be implemented at a **different location , using different DBMSs.**

- [37] explains how the 3-schema Specification Architecture maps on the **network components or IT facilities,**
- provides an simplified overview of **services** in **distributed systems.**

[38] IT facilities are no separate physical network, but three kinds of component in the same network

- ***data location*** *should be transparent up to limits imposed by robustness requirements*
- ***process location*** *should be transparent to the user*
- ***technical architectures*** *are packages of required services*
- ***logical requirements*** *are mapped to architecture via required services*
- ***distribution-related*** *concepts are identified in logical requirements*

Detailed description of design principles:

- *A distributed system is a* ***co-ordinated system***[39]

The related activities are co-ordinated when what happened in one activity should be ***consistent with*** *what happened in other related activities .The co-ordination may be:*

- ***tight***
- ***loose***
- ***both in the same system***

- *Systems that serves different location types can be co-ordinated on entity aspects,*[40]

  Entity aspects is a view of some more-widely-defined entity type. We define aspects so that we can develop separate specification for an entity type ***that appears in more than one context .*** *Although* ***aspects modelling*** *is a useful* ***extension*** *of logical data modelling, it is applied only to entity types that* ***shared between different context.***[41]

- ***Logical design*** *should be independent of architectures*

  When a decision has been taken to develop a ***co-ordinated system, the common approach is taken to***

  - ***specification of requirement***
  - ***logical design***

  The design process and method is part of the "traditional" SSADM method.

- ***Data location*** *should be transparent up to limits imposed by robustness*[42] *requirements*

  Ideally, location of data should be ***completely transparent.*** *There should be a software service layer that maps data requests to the appropriate location.*

[39] some system that operate in several location are **uncoordinated.** They are regarded as **separate system.**
[40] For example EU Rent is one of three different businesses - car rental, hotels and airline. It makes sense for all three business areas to have **consistent customer information.**.
[41] see. Further development areas
[42] **Robustness of a distributed system** is defined in terms of services each type of location must provide when communication with other type of location is not possible.

There is no rule on how to reconcile theses conflicting requirements, it is a trade-off[43] ! *One opportunity: maximise transparency in the location until the robustness requirement can be satisfied.*
Note: *Our point of view on the most important and useful decision criteria can be the trade -off between the robustness and the consistency of the system. But in that case we should take account the fault tolerant capability of the information technology.*

- ***Technical architectures*** *are packages of required services*
  The technical architectures assumed to be available are:
  - *distributed database management system*
    - *remote database access without distributed commit[44]*
    - *remote database access with distributed commit*
  - *distributed transaction processing*
  - *object oriented distributed transaction processing*
  - *electronic mail*

TECHNICAL ARCHITECTURE ARE NOT THE STARTING POINTS FOR SYSTEM ARCHITECTURES !

- ***Logical requirements*** *are mapped to architecture via required services*
  Types of services needed to support distribution - related concepts:
  - *information exchange services*
    - *remote database service*
    - *transparency of data location*
    - *transparency of data replication*
    - *message exchange*
  - *process co-operation services*
    - *remote processing*
    - *distributed commit management*
    - *security services*

The logical chain of the development:

logical requirements **>** *services* **>** *technical requirement*

The technical architecture are constrained by (among others) by the hardware and software procurement opportunities.

- ***Distribution-related*** *concepts are identified in logical requirements*
  - *The major distribution related concepts[45]:*
    - ***business related***

[43] see Further development areas
[44]
[45] definition of the concept see below

- ***distributed system***
- ***business location type and instance***
- ***actor type and instance***
- ***role played by actors***

- ***data processing related***
  - ***distributed data processing system***
  - ***data processing location type and instance***
  - ***workstation***
- ***mapping of data and process to location***
  - *support for user activity by user-interface (UIO= user interface object)*
  - *availability of automated processes at data processing location(s)*
  - *storage the data at data processing location(s)*

The distribution related concepts should be included in logical requirements !

Design steps
in the 3-schema Structural Architecture Model
(Summary)

- ***decide*** *on a co-ordinated distributed system.*
  Note that
  - *the decision may have been taken at a* ***strategic level***
  - *a distributed system may be an option to be compared with a centralised system*

- ***use*** *SSADM (see Reference) develop requirements and logical design. Define data and processing specifications as if for a* ***single, integrated system*** *with the addition of*
  - *business locations and actor roles*
  - *data processing locations*
  - *estimation of traffic between locations*
  - *robustness requirements*

- ***map*** *external processes to user - interface server and further, to workstation types of business location types, duplicating processes as necessary*

- ***partition the data in system*** *across data processing location and allocate the corresponding conceptual process these locations*

- *carry out a* ***robustness analysis.*** *Introduce compromises into the design to satisfy* ***robustness requirements:***
  - *replicate data*
  - *specify degradation in some services when part of required data is not accessible*
  - *identify services that will not be available when communication lines are down*

- *identify the services needed to support logical requirements and select the technical architectures that best provide them*
  - *the selection may be constrained by installed technology or procurement policy*
  - *the project may have to develop software for some of required services*

- *specify the* ***physical design***
  - *map the logical design on to selected technology*
  - *analyse performance and modify the design to reduce/remove performance problems*
  - *produce component specification for implementation*

Distribution concepts

We may describe an distributed system on several ***levels of abstraction.***
Most of the concepts are ***specific*** *to distributed system.*
Concepts can be classified into[46]

- ***business concepts***
- ***data processing concepts***

Both categories can be subdivided:

- ***location concepts***
- ***object/data concepts***
- ***processing concepts***

- *Actor/business location concepts*

Location instance *is a location that can be identified uniquely.*

- *Business Location Instance*
- *Data Processing Location Instance*

Distributed Information System(DI) *is a set of* ***co-ordinated business location***
instances

Distributed Data Processing System *is a of* ***co-ordinated DP location instances.***

DI system ***supported*** *by a DDP system, in some cases may be supported* ***more than one.***[47] *or a DDP system may support more than one DI system.*

Actor *is an agent that carries out a task in an enterprise at a Business Location Instances. An actor may be:*

- *human, user*
- *DDP system*
- *(in extreme cases) DDP system only*

The processing in a business is carried out in units called ***Procedure/Function*** *which is a set of* ***business activities*** *that is seen a unit from business point of view. From data processing point of view a process/Function consist of*

- *a set of Manual Activities*
- *a set of Massages in Dialogue*
- *a set of Validation/Formatting Processes*

Actual business takes place at a Business Location Instance. At a Business Location Instance the Actors do their job by carrying out

[46] Figure 4.1.
[47] a JIT logistic system may be supported by several companies DDP systems

Procedure/Function. For this Actors generally need the service of DDP system i.e. ***automated*** *Procedure/Function and stored Data.*

The service of a DDP system are provided at a Business Location Instance by a set of ***Workstations,*** *Which enable Users to invoke Procedures/Functions and to use Data stored by DDP System.* ***A DI system usually involves a large number of Actors an Location Instances!***

The Procedure/Function carried out by an Actor are classified by ***Actor Role,*** *i.e. Actor Role is a set of Procedures/Functions. Actors are classified by* ***Actor Type.*** *An Actor Type is a classification of Actors according to similar tasks or jobs An Actor Type is defined by a set Actor Roles*

Business Location Type: *denotes a set of Business Location Instances where the same Actors Roles are played. Each Business Location Type has a set of* ***Workstation Requirements that result from the Actor Roles*** *that make up the business Location Type.*

3.2.2.3. Decisions specific to DIS

***Decisions** which address distribution requirements*

The single (centralised) subsystem development methods (the SSADM for instance) are a well elaborated, structured procedures. In order to that these methods can be applicable in the distributed system development circumstances we should consider other decision criteria:

- *Decision taken before the project*
  - *whether to use already installed technology*
  - *recognising the effect of external requirements*
- *Whether to build one co-ordinated system or several uncoordinated. Decision factors*
  - *functional requirements*
  - *cost*
  - *stability of design*
- *robustness vs. consistency*
- *distribution of data*
  - *partitioning of data*
  - *replication of data*
- *distribution of processing*
  - *assignment of internal processes to data processing locations*
  - *assignment of external processes to user interface server (at business location)*
  - *replication of processes*

***Pre-project decision** strategic decision requiring a distributed solution may have been taken before the project*

For example;

- *machine and network may already be installed*
- *external requirements*
  - *legal constraints:*
    - *certain data may have to be stored separately for reason of data protection*
    - *fiscal information relating to the national part of an international organisation*

Single co-ordinated system vs. multiple uncoordinated system

Factors to be taken into account:

- ***Reasons for developing a several projects***
  - *after we have defined a syntactical interface between projects we can separate the development into more independent project*
  - *we get individually imlementable packages*

- *they can be delivered earlier*
- *the reduced complexity can lead to decrease of unit cost for the development*

- ***Reasons for developing a single project***
  - *there is a risk of consistency problems between systems whether are developed through several projects*

Distribution on grounds of functional requirements
What is to be decided? developing a centralised or distributed system because of ***functional requirements***[48]

Factors to be taken into account:

- *when the network is down in a centralised system the it is impossible for users to progress at any location. In distributed systems there may still be processing what can be carried out locally.*

  The process would be:

  intra - location process: *processes requiring local data only*

  inter - location process: *processes requiring data from other locations*
- *when one of the components in a distributed system fails, the most of the work in progress need not to be interrupted if there is* ***sufficient redundancy***

Distribution on the ground of the cost *It means that we must make a business case for the option to be chosen*

Factors to be taken into account:

- *Usually when comparing* ***computer system costs, "lifetime " costs***[49] *are used*
- *accurate lifetime costing of any system is difficult*
- *Cost factors which contribute to the total cost of a distributed system are:*
  - *communication costs*
  - *hardware costs*
  - *operational costs*
  - *software costs*

Distribution on the grounds of stability design, *apart from robustness and cost considerations there are other benefits that might lead to the choice of a distributed system*

Factors to be taken into account:

[48] this decision is closely related to robustness issues (see that point)
[49] that is the cost of initial system together with estimated costs over the system lifetime

A distributed system tends to be more stable with respect too each of the following:

- ***load function*** *- in a distributed system where a lot of a processing can be done locally, the response is faster and more predictable than in a centralised system where the response depends on the work of other users*
- ***system changes -*** *it seems easier and more economical to add or update locations in distributed system than to expand a large central computer*
- ***data theft -*** *the potential for loss or damage is greater with a single centralised location then where data is spread across many locations*

Robustness[50] vs. consistency *Once the decision in favour of a distributed solution has been made, we have to decide on the* ***detailed trade - off*** *between consistency and robustness*

What is to be decided? The decision is about the ***degree of the autonomy of the location of the system***

One extreme - *where the location have no autonomy, thus none is able to work independently of the others*

Other extreme - *where locations have total autonomy, the don't need any communication with other locations*

In distributed system - *a solution between two extremes, in order to satisfy the users* ***"robustness and consistency requirements"***

Relationship between autonomy, consistency and robustness:

	AUTONOMY	*LOCATION CONSISTENCY*	*ROBUSTNESS*
MULTIPLE LOCATION ONE SYSTEM	*NONE/MINIMAL*	*FULL/OPTIMAL*	*NONE/MINIMAL*
MULTIPLE LOCATION MULTIPLE SYSTEM	*FULL/OPTIMAL*	*NONE/MINIMAL*	*FULL/OPTIMAL*

Factors to be taken into account:

- *The decision on the importance of robustness and consistency can be guided by:*
  - *he robustness which is inherent in the business*
  - *the risks to the business in the case of consistency violations*
- *In every distributed system there is some* ***inherent robustness*** *because of possibility of doing some work locally or involving*

[50]The extent to which an IT system can keep operating when communication links to one or more of its dataprocessing locations are out of action.

*only of subset of all locations. This level of robustness is then even in the case of **maximal constraints on consistency.***

- *It has to be decided what the system is to do in the case that it cannot guarantee data consistency because of network or location failures. We need to decide:*
  - *what the **minimum level of consistency** to continue working*
  - *what **reduced functionality** the system should provide in the case of failures*
  - *how the system should **recognise inconsistent states***
  - *how the system can reach a **consistent state** again*

Distribution of data** Distribution of data is about defining **the internal schema

What is to be decide? On where to store objects physically has two dimensions:

- ***partitioning** means the separation of data representing an object class into proper subset which are assigned separate location*
  - ***horizontal partitioning** where we separate the rows of a table*
    - *is particularly useful in a situation where a **function is replicated** at several location and each location is responsible for subset of the object*
  - ***vertical partitioning -** where we decompose a table by columns*
    - *supports functions at different locations with responsibly separate data requirements*
- ***replication** means storing copies of object or parts of objects at more than one location*
  - *is particularly recommended for data which is updated relatively infrequently (catalogues, train schedules, telephone directories...)*

Guidance:
The default principle in distributing data is to store data at the location where it will be accessed most frequently.
*Other consideration like **security, data integrity or cost.***

***Partitioning the data** facilitates:*

- ***efficiency** - data stored where it is used and separated from other data used by other users or applications*

- ***local optimisation*** *- data can be stored to optimise performance for local access*
- ***security*** *- each location controls the access to local data, only giving access to other location via secure interface*

Disadvantages of partitioning:

- ***inconsistent access speed:*** *When data is required from several locations, access times can be significantly lower*
- ***back-up vulnerability:*** *If data is not replicated and the location at which the data is stored becomes inaccessible then no process will be able to use that data.*

Replication of the data *facilitates:*

- *reliability*
- *fast response*

Disadvantages of replication:

- ***storage requirements***
- ***multiple updates:*** *requires additional processing capacity and increases network load.*
- ***synchronisation*** *If updates are not carefully synchronised copies of same data will be inconsistent*
- ***propagation of corrupt data***

Distribution of processes *When developing a distributed system we have to decide how to distribute processes:*

- *procedure implementation*
- *method implementation*
- *validation and formatting implementation*

In the 3-schema Specification Architecture the ***conceptual model*** *is both*

- ***static***
- ***dynamic***

model of business world *capturing all of the* ***semantics***[51] *about the system.*

The conceptual model processes are implemented as ***methods at data processing location.***[52]

The ***external design*** *is a model is how the user interact with the data processing system, i.e. a model of the procedure implementation.*

The ***internal design*** *includes the program - data interface between method implementations and data servers.*

What is to be decide? Procedure implementations, and validation and formatting process implementation are allocated to ***user interface server,*** *Method implementation is allocated to* ***process server.***

[51] the necessary business knowledge

[52] which may be distinct from data servers and user interface servers.

What is then needed is ***mapping of***

- ***user interface server***
- ***process server***
- ***data server***

to data process location. *The decision of where to locate the processes has the dimensions:*

- ***partitioning -*** *we mean the assignment of server to location*
- ***replication -*** *means the functionality is offered at more then one location.*

Guidance
The default *is to place* ***user-interface servers at the same locations as the users.*** *We recommend* ***replication*** *of user-interface servers at each business locations.*

The process server: *the problem is of a different nature. For the decision partitioning we can take* ***the location of data*** *which is needed by the processes*

- *when we partition data* ***horizontally****, we locate a* ***copy*** *of the process server with each partition of the data*
- *in the case of* ***vertically*** *partitioned data, the will be* ***different*** *types of process server for each partitions,*

3.2.2.4. Conclusion

- *the 3-schema Specification Architecture is an extension of the SSADM which is originally developed for single, centralised system design and so the backbone o this method is the SSADM procedures*
- *the system design*
  - *is a very complex intellectual activity which consist of well defined (engineering activities) and heuristic (discover) element.*
  - *has not only technical but sociotechnical aspects*
- *the distributed characteristic of the distributed system comes into the scene when we consider the decision criteria specific to DIS*
- *in the system being under development or modification business and actor location instances are* ***distributed,*** *it is an natural specific of the organisation (location, or lay out)*
- *the data processing location may be single and in that case it seems centralised, or dispersed, and so* ***distributed***

- *the workstations are linking chain between business location instances and data processing location*
- *the data and process mapping (location) are depending on the location of data processing and/or workstation locations*
- *one of the most useful designer behaviour or thinking concerning mapping decision to shift the decision as near as possible to the physical design level*
    - *the technical design is not a start-point*
- *there are non answered questions in development of DIS:*
    - ***partitioning** of database (local or global approach)*
    - *optimised solution of the trade-off between robustness and consistency taking account the system stability*
- *the object as a natural unit of distribution force the evolution of DIS developing methods into the object oriented development direction, but it has not final answer of the questions mentioned earlier*

3.2.3. The SSADM Extension[11]

This section describe how the SSADM can be used in distributed system development environment. We suppose the traditional SSADM system knowledge or refer to books and reference manuals.

***Representation of distribution concept in work products** describes the mapping of the general distribution concepts to SSADM terminology*
Source: [11] Figure 9.5.

DISTRIBUTION CONCEPTS	***SSADM TERM***	***DESCRIBED IN SSADM PRODUCTS***
actor	*user(interface)*	*individual users not described*
actor role	*user role*	*user catalogue or user roles*
actor role on workstations	*no specific term*	*menu for user role or user role/function matrix*
actor type	*user(type)*	*user catalogue*
association	*relationship*	*LDM, relationship description*
business location instance	*no specific term*	*individual locations not described*
business location type	*no specific term*	*modelled as entity type in LDM*
commit unit	*event* *enquiry*	*effect correspondence diagram* *Enquiry Access Path*
commit unit trigger	*event data* *system-recognised event* *enquiry trigger*	*Update Process Model* *Update Process Model* *Enquiry Process Model*
Data Processing Location Instance	*no specific term*	*individual locations not described*
Data Processing Location Type	*no specific term*	*May be modelled as entity type on LDM*

DP input message	*Input structure*	*Function Definition*
DP output message	*Output structure*	*Function Definition*
Input to Method	*Effect Data*	*Derived from event data*
Location instance	*no specific term*	*individual locations not described*
Manual activity	*User procedure*	*User Manual*
Message channel	*no specific term*	
Message in Dialogue	*External Data Flow*	*I/O Structure* *I/O Structure Description*
Method	*Effect*	*Entity Life History* *Effect Correspondence Diagram*
Method	*EAP entity access*	*Enquiry Access Path*
Method Dependency	*ECD correspondence* *EAP correspondence*	*Effect Correspondence Diagram* *Enquiry Access Path*
Method Implementation	*no specific term*	
Method Invocation	*no specific term*	*Effect Correspondence Diagram* *Enquiry Access Path*
Object Class	*Entity aspects*	*LDM, Entity Description*
Object Class Partition	*no specific term*	
Object Population Representation	*no specific term*	
Output by Method	*no specific term*	
Procedure Implementation	*no specific term*	
Procedure used in Role	*Entity in User Role/Function Matrix*	
Procedure/Function	*Function*	
Record Population Partition	*no specific term*	
Record Type	*Database Record*	
Record Type Storage Capability	*no specific term*	
Role at Location Type	*no specific term*	
Role Played	*no specific term*	
Validation/formatting Process	*Input Process* *Output Process*	*Physical Process Specification*
Validation/Formatting Implementation	*Input Process* *Output Process*	*Physical Process Specification*
Workstation	*no specific term*	
Workstation Requirement	*no specific term*	

3.2.3.1. Changes to modelling steps and techniques

Summary of the changes

- *only one structural change is needed: selection of Business Options has two parts:*
  - *selection of business option for functional scope*
  - *selection of distribution options for chosen business option*
- *there are some changes of detail to products see Table below*

SSADM PRODUCTS	***SCHEMA***	***SERVER TYPE***	***LOCATION TYPE***
FUNCTIONS,	*EXTERNAL*	*INTERFACE*	*BUSINESS*

DIALOGUE		*SERVER*	
LDM, ECD, EAPs	*CONCEPTUAL*	*PROCESS SERVER*	*DATA PROCESSING*
PDL, database	*INTERNAL*	*DATA SERVER*	*DATA PROCESSING*

- *An important new concept introduced: an entity that is shared between locations is divided into **aspects.** This partitioning is done from Logical Data Model (LDM)*
- ***Techniques** are newly introduced:*
  - ***robustness analysis:** to identify what business activity can be supported if locations are cut-off from rest of the system*
  - ***first-cut distribution design:** on the base of data design technique*
  - ***communication mapping:** summarise the types and volumes of communication between location*

3.2.3.2. Changes to the SSADM Structural Model and Products

Feasibility study

- *no structural changes are needed*
- *the Project Initiation Document should provide an overview of the **geography of the organisation***
- *Requirement Catalogue is extended of requirements for*
  - ***robustness***
  - ***consistency***
- *in the case of modelling current DIS the DFD must be modified:*
  - *communication between different instances*
  - *use different instances of the same type of data store*
  - *communication with different instances of the same type of user*

Investigation of Current Environment (Stage 1)

- *no structural changes are needed*
- ***the LDM is developed as a single model for the entire system (not a separate LDMs for each location)***
- *some changes in details of steps*
  - *should describe the geography of the organisation*
  - *reason of distribution*
  - *requirements addressee the*
    - *robustness*
    - *consistency*

        - ***constraint for distribution***
    - ***types of business location*** *must be defined*
- *Derive Logical View of Current Services (Step 150) includes* ***additional***
    - ***techniques***
    - ***event identification***
    - ***some guidelines*** *for development of* ***Data Flow Model (DFM)*** *and* ***LDM***
    - ***DFM***
        - ***DFM*** *is directed at the external design*
        - *it is not concerned with data location*
        - *it is built from the bottom up around the user roles*
        - *user roles are defined at business location, if the same user role occurs at more than one type of business location a* ***separate low-level DFD*** *is drawn for it*
        - *every process needed by business location is assumed to be invocable from workstation at that location, the are not location information on DFD*
        - *communication via DFD processes between different instances of the same user role at different locations have to be shown on DFD*
    - ***LDM***
        - *list of attributes is built up for each entity type*
        - *enquiry rows in the Entity Access Matrix*

Business System Options (Stage 2)

Two part of procedure proposed:

- ***selection of business options*** *concerned with functional requirements not supported the current system*
- ***selection distribution option*** *concerned with how the agreed business option can be distributed across the business and data processing location.* ***The major concepts in distribution:***
    - *functions are available at business locations that need to use them*
    - *update and enquiry processes are available at data processing location and invoked by functions*
    - *the database is stored at data processing locations*
    - *distribution is based on partitioning the LAM and deciding:*
        - *location of data storage*
        - *location of event and enquiry processes*
            - *at business location*

- *at data processing location*
- *distributed across several types of data processing location*
- *split between business location and data processing location*

- *whether replicate data*
- *the trade off between robustness and temporary inconsistency*

- *for each design identify*
  - *communication between types of location*
  - *state whether is to be losely-coordinated or tightly co-ordinated*
- *carry out robustness analysis for each type of business location:*
  - *what can a location do if it is cut ?*
  - *what can a location not to do if it is cut off?*
  - *what can other locations not to do if this location is cut off?*
- *identify the inconsistencies and its* ***timescales***

The steps are not in sequence, they need some ***iteration.***

Requirement Specification (Stage 3)

- *no change in procedure*
- *separately-drown LDM is needed for each type of data processing location, where an entity type is used in more than one location, it is split into aspects (some attributes may be replicated in more than one aspects)*

Technical System Options (Stage 4)

The procedure is generally unchanged but has to be take account of distributed processing architecture that will best support requirements

Logical Design (Stage 5)

No structural changes are needed

The changes in ***processing:***

- *design user dialogues (Step 510) is unchanged*
- *define update processing (Step 520) the technique is extended:*
  - *developing consolidated UPM[53], PDI[54] will mange remote database access (where needed)*
  - *partitioning the ECD[55] by location and develop local UPMs*

[53] update process model
[54] process data interface

- *object-oriented implementation, where each ECD node is defined as a separate method*

Physical Design (Stage 6)

- *Optimise Physical Data Design (Step 640) is as in the SSADM, but there are more design issues to be considered, In particular **minimising the volume of messages** passed between sites may be important*

Changes to current SSADM techniques

***Requirements Definition** is extended*

- *what is required from DIS*
- *robustness requirement*
- *consistency analysis*
- *constraints on distribution*

***Logical Data Modelling** has two significant modifications:*

- ***Validation** extending the techniques for validating the LDM against processing requirements*
- ***Entity aspects***

***Data Flow Modelling** needed an extension of notation:*

- *to show the external entities, process locations and data stores*

***Relational Data Analysis,** the is two **minor** changes to Relational Data Analysis*

- *system generated keys: for some entities mechanism are needed within the system to generate keys for new instances.*
- *aspects: when an entity is split, one aspects is defined as a ”base” aspects, modelling the basic existence of the entity*

***Entity Type History** (ELH) analysis: there are two significant changes*

- *building an **Entity Access Matrix** before developing ELHs. The essential characteristic of the conceptual schema is that what information support is provided: the content of LDM and the means of keeping it up to date*
- *we can identify the **events** need to update the LDM and develop a matrix of required access to the LDM. We need to know about events: from which type of location originate in order to develop distribution options.*
- ***Creating the Entity Access Matrix:** it is an extension of the idea of the entity/event matrix Suggested form:*

55

			ENTITIES		
EVENT/ENQUIRY	*VOLUMES*	*INVOKED FROM*	*ENTITY 1*	*ENTITY 2*	*.....*

The type of business location from which the event/enquiry can be invoked

Indicates the LDM entry point

- ***modelling of entity aspect***

Update Process Models *three are three approaches*

- *create UPMs as in SSADM and have the PDI issue local and remote database calls,*
- *to treat each entity type aspects as a separate object class and specifies an object-oriented implementation*
- *to create local process models for each location.*

Additional SSADM techniques

Robustness Analysis *is carried out for each distribution option. It is done initially from Entity Access Matrix produced in Stage 1 and 2. It is confirmed for the selected option after EAPs and ECDs in Stage 3.*

First-cut Distribution Design *is produced in Step 230. It aims the most distributed option and provides a self-contained data model for each type of business location.*

Produce Communication Map *provides a summary of communication between data processing locations and is produced to support distributed options*

- *it is developed from Entity Access Matrix built in Step 230, Develop Distribution Option an confirmed in Stage 5.*
- *it is a From - To table (between data processing location)*
- *is supported by a table of volumetric data*
    - *is one of summaries used to estimate traffic across the network*
    - *the other summaries used to estimate traffic:*
        - *messages between business location and data processing locations*

- *messages between data processing locations and data storage locations*

3.3. Management decision criteria[56] that can be used attempt to balance the distribution IT resources

PRESSURES	*TOWARD CENTRALISATION*	*TOWARD DECENTRALISATION*	*TOWARD DISTRIBUTION*
Management control	*Hierarchical* *Standardisation* *Efficiency* *Organisational security*	*Entrepreneurial* *Local responsiveness* *Effectiveness* *Local security, reliability*	*Information age* *Learning* *Efficiency and effectiveness* *Global security, reliability*
Technology	*Efficient use of technology* *Specialised, costly equipment that is required by all* *Requires specialised expertise to operate and manage*	*Effective use of resources* *Low cost of "off-the-shelf" equipment's* *Expertise needed to operate and manage is widely available*	*Effective and efficient use of resources* *Mix of specialise and "off-the-shelf"* *Mix of specialised and general expertise*
Information	*Organisational information* *Maintain information standard* *Level of information sharing can be accommodated by network capacity & budget*	*Local information* *Maintain data relevance* *Desire to optimise network capacity and minimise cost*	*Increased need for vertical and lateral sharing (including interorganisational)* *Desire to optimise information relevance and standards* *High capacity, low cost networks are available and manageable*
IT professionals	*Scarce resources with specialised knowledge and expertise* *Minimise turnover risk*	*Widely available resources and generalised expertise*	*Mix of SCEs and generalised IT professional resources*
End users	*Low level of technical skill* *Satisfied with routine information reporting* *Lack of motivation to manage IT*	*Technical sophistication* *Desire flexible access to timely information* *Motivated to manage IT*	*High level of technical and information literacy* *High level of commitment and motivation to become an active player in defining, managing information*
Culture/organisational fit	*Organisation is structured as a functional hierarchy* *"Command and control" culture* *IT function has always been centralised*	*Organisation is structured as decentralised profit centres* *"Results "culture* *Significant decentralisation of IT resources and control*	*Organisation is structured as matrix. of autonomous, interfunctional teams* *"Commitment, collaboration and result" culture* *Both centralised and decentralised IT resources and control*

[56] [56] M. L: Applegate Designing and Managing the Infromation Age IT Architecture Harvard Business School 196-005 pp. 11.

REFERENCES

Distributed Information System

[1] E.Simon
Distributed Information Systems
From Client / Server to Distributed Multimedia
McGraw Hill Companies, 1996

[2] Kathy Borer, Verlyn Johnson, Anders Nilson, Bradley Rubin
Business Process Components for Distributed Object Applications
Communications of the ACM June 1998. pp. 43 - 48.

[3] Geoffry Bock, Linda M. Applegate
Doing Business in a Distributed World: Clients, Servers, and the Staff in Between
Harvard Business School Publication, 9-195-211
October 12, 1995

[4] Linda M. Applegate
Managing in an Information Age: IT Challenges and Opportunities
Harvard Business School Publication, 9-196-004
August 8, 1995

[5] Lynda M. Applegate
Managing in an Information Age: Organisational Challenges and Opportunities
Harvard Business School Publication, 9-196-002
September 25, 1995

[6] Lynda M. Applegate
Designing and Managing the Information Age
Harvard Business School Publication, 9-196-003
October 12, 1995

[7] John Simon, Thomas Davenport
Managing Information Technology: Organisation and Leadership
Harvard Business School Publication, 9-189 - 133
January 19, 1989

[8] Lynda M. Applegate
Designing and Managing the Information Age IT Architecture
Harvard Business School Publication, 9-196-005
September 26 1995

System Development

[9] D. E. Avison, G Fitzgerald
Information System Development: Methodologies, Techniques and Tools
Second Edition
McGraw Hill Book Company, 1993

[10] Bo Dahlborn, Lrs Mathiasen
Computers in Context,
The Philosophy and Practice of System Design
NCC Blackwell Oxford1993

[11] Dr. M. Breu, A.A.J. Hall, K. Robinson
Distributed Systems: Application Development
September 1994,
The Government Centre for Information Systems

[12] D. J. Flynn, O. F. Diaz
Information moiling an international perspective
Pretice Hall, 1995.

FURTHER DEVELOPMENT AREAS

- *object oriented distributed system in the business*
- *CASE methods in the distributed system development*
- *learning organisation paradigm and the distributed systems*
- *in the DIS development method*
    - *data modelling method extending the partitioning procedure concerning the local and global requirements*
    - *the handling the trade-off between the robustness and consistency requirements*

FURTHER READING

- *SSADM V4 Reference Manual*
  ISE Library
- *Customising SSADM*
  ISE Library
- *Application Partitioning and Integration with SSADM*
  ISE Library
- *Reuse in SSADM using OO*
  ISE Library