Centralized versus Decentralized Information Systems

A Historical Flashback

Mats-Åke Hugoson

Jönköping International Business School, Sweden; mats-ake.hugoson@ihh.hj.se

Abstract: This paper brings into question whether information systems should be centralized or decentralized in order to provide greater support for different business processes. During the last century companies and organizations have used different approaches for centralization and decentralization; a simple answer to the question does not exist. This paper provides a survey of the evolution of centralized and decentralized approaches, mainly in a Nordic perspective. Based on critical reflections on the situation in the end of the century we can discuss what we can learn from history to achieve alignment between centralized and decentralized systems and the business structure. The conclusion is that theories, management and practice for decisions on centralization or decentralization of information systems must be improved. A conscious management and control of centralization /decentralization of IT support is a vital question in the company or the organization, and this is not a task that can be handled only by IT-specialists. There is a need for business oriented IT management of centralization/decentralization.

Keywords: Structures of information systems, Systems interaction, Independence between systems, Centralization, Distributed systems, Decentralization

1. Introduction

Should we centralize or decentralize information systems to create greater support to different business processes? Different approaches for centralization and decentralization have occurred during the last century; and there is not a simple answer to the question.

This paper gives a survey of the evolution of centralized and decentralized approaches, mainly in a Nordic perspective. The paper uses observations and episodes starting in 1965, which means that some of the conclusions reflect personal opinions. Other persons working in the field during this period may very well have reached other conclusions. As the topic centralization versus decentralization often causes debate, perhaps the paper can give additional fuel for discussions.

After a general analysis of what centralization and decentralization really means, starting from a business perspective, we follow and scrutinize the centralized track and the decentralized track for information systems.

A major difference between the two alternatives is the degree of independence between systems; therefore, we analyze the problem and principle of independence in detail. We relate the situation in the end of last century to the different views from some researchers. Finally, we address the aspects that we can learn from history, which leads to some conclusions.

2. The Concept of Centralization and Decentralization

In a business perspective, decentralization means that business can make decisions locally. A business unit can choose the way to use local resources to fulfill objectives for that unit. The unit must cooperate with other units in the company (perhaps also externally), and must report to management in a specified way. However, there is a freedom of action to perform *within* each business unit.

In addition, in a decentralized organization there must be a central coordination. Without management and control, the organization ends up in anarchy. Even if this central coordination such as standardization can set restrictions for each unit, the main criteria for decentralization is the right (and responsibility) to form an efficient *inner structure* in each business unit, using local resources to fulfill objectives and tasks that are set for the unit. Changes in this inner structure should not affect other units.

When applying the concept of centralization and decentralization to information systems, we must be more specific and analyze some different alternatives. To develop one common system for an organization is of course a centralized approach. To develop and implement a number of systems in the company is on the other hand not always a decentralized approach. If these systems are developed "wildly" without any coordination and have no computerized interaction, we must characterize it as anarchy. However, even if these systems have a computerized interaction the question remains: what is really a decentralized structure of systems? If the different systems have a common database, we must still consider it as a centralized approach, as each system is directly dependent of the common database, and a change in this database affects many systems. Also distributed systems with a centrally controlled data storage have a limited freedom of action in each local system.

A strict definition of decentralized systems could be that each system in the structure must fulfill specified demands on interaction with other systems, but it should be possible to develop (and change) the inner structure in each system, including data storage, without dependences to other systems, as long as the specified interaction stands. It must for instance be possible to insert systems of different origin into the structure. The main condition is that each system must interact with other systems as specified.

From this short analysis, it is evident that we must consider both the delineation of systems (central – local) and the principles for systems interaction

when we discuss decentralized information systems. Independence between systems is a major criterion on decentralization, which we will discuss after a historical review.

3. The Centralized Track

According to a statement, which is said to come from IBM research and development in the early ages of computing, the estimation was that three or possibly four big mainframes would give enough computing capacity for the industrialized world, as long as could be foreseen. This really represents a centralized view, both on computers and on computing. Even if we cannot state the origin of this statement, it is easily verified that the start and evolution of computing in most companies followed a centralized track.

3.1 The MIS Concept in the 1960s

Between 1965 and 1970 industry initiated many large projects to support the total organization with *one* information system, generally referred to as the "management information system". For example, with Volvo, the project was named VIS; in SAAB it was the TIPS-project that had the same total scope.

There was a general opinion among people working with ADB (Automatisk DataBehandling) – EDP (Electronic Data Processing) that this was the right approach, and the Nordic Data Union arranged conferences on the MIS concepts in Denmark, Norway, and Sweden.

However, there were people who did not fully agree to this 'one system approach'. At that time, I was a young systems manager and became secretary of the planning committee for these Nordic MIS-conferences. Professor Börje Langefors was an invited speaker to one MIS-conference in Kungälv, Sweden. I remember from the planning discussions that he hesitated, with the argument that total management systems are unperceivable and will fail. He had already in 1967 published a paper about "Directive Information for Systems Control" [7]. He pointed out the difference between directive and local information and he argued for a split between what we today refer to as executive systems and systems to support local operations. Furthermore, in the book System för företagsstyrning [8] he had presented Teorin för oöverblickbara system (The Theory for Imperceivable systems). He clarifies the necessity to handle complex systems through structures of perceivable subsystems instead of a total system approach. Anyhow, he accepted the invitation and gave at this conference a very humble presentation of his objections to MIS.

The outcome from the MIS-projects in different companies was generally not the expected. They gained experience, some results were achieved, and projects were perhaps not considered as failures. However, none of the projects mentioned ended up in a total management information system. Among the reasons given as explanation to the shortcomings, we often find technical limitations, lack of methodology, limited resources, or perhaps sometimes insufficient experience to

run big projects, but the basic centralized approach was very seldom called into question. Also in the academia, the centralized approach has been dominating from the beginning. Some examples will be given.

3.2 VLDB in the 1970s

Development of database technology strengthened the centralized concept. The (implicit) basic assumption was that each company must have one total database, and therefore methods were developed to handle the large data base that was necessary to support the total organization. In the end of the decade conferences on very large databases (VLDB) were arranged all over the world.

3.3 The Search for a Total Integrated Conceptual Model in the 1980s

After the period with focus on technical databases, the data modeling and later the conceptual modeling became the dominant paradigms. Despite these changes, the aim was still to find a tool to make a total central model for the whole company. Thus, based on these theories they started new projects for total data models in many companies, even at Volvo.

People gradually understood that it was not possible or suitable to have all data processing in one system, but then this total conceptual model should serve as a blueprint to implement different information systems. The principle was that different systems should refer to a central data model implemented either as a central database or as a set of centrally controlled databases.

In the discussion about centralization versus decentralization, this view must still be considered as a centralized approach, even if the outcome is a number of systems. In this approach, there are strong dependencies between the different systems. There is not much freedom of action to implement systems of different origin in such a structure, which we took as a major criterion on decentralization. Attempts to split the central database into a number of distributed systems do not really change the picture as the inner structure in each system is still centrally controlled.

Another problem with total conceptual /data models was that a single project could generally not get an overview of the total conceptual model, in any case not in more complex organizations. The alternative was instead in practice to establish a number of projects within the company, each one using conceptual data modeling for the scope of their own system. The theoretical (and practical) problem was then to fit these models together.

In 1989, an IFIP conference on information system concepts took place in Namur, Belgium. One of the main themes was ways to integrate conceptual models, with presentations such as Integration of Information Submodels [6], and Levels of Abstraction as a Conceptual Framework for an Information System [5]. Attempts during the conference to discuss decentralization of these submodels, through other principles for interaction between systems, were directly rejected.

Again, we must view this as a centralized approach, limiting the freedom of action in each project. Before we sum up and state the results from the centralized approach during the last century, let us have a look at the decentralized track.

4. The Emergence of Decentralized Information Systems

As long as computing was performed as a 'closed shop' in the data hall, ideas of local systems were just ideas. At the end of the 1970s, when minicomputers and microcomputers began to be available at reasonable prices, the scene changed. At about the same time, new methods and tools for development of small systems were developed (prototyping, experimental design). In addition, system packages for local applications, for instance for production planning, appeared on the market. Suddenly, companies bought or developed and implemented local systems in different parts of their organization. One could view this movement as a protest against central systems (or at least as a way to fulfill local demands that could not be supported by centralized systems).

Professor Staffan Persson at Stockholm School of Economics used the programming language APL to develop local systems extremely fast. For example, within a day he could help a company find a solution to a planning problem, program the system, set it in operation on the computer he brought with him, and just leave a system in use on the table. This way to cut lead-time in systems development from several years to hours must be some kind of a record. These local systems evidently represent a high degree of independence, but according to earlier discussions, they do not fully meet the demands for decentralization, as they did not consider interaction with other systems. One could characterize this approach for systems development as anarchistic, and it very often caused conflicts with the IT-department.

A little more coordinated approach emanated from organizations with many units of the same kind. Healthcare is a good example. Extensive handling of local information, both for administration and for medical records, called for IT support. A local system was developed or bought, often based on local initiatives, sometimes as a part of an IT plan. When the system was tested and workable, it was implemented also in other units in the organization, but still as stand-alone systems. This represents a more long-term perspective that makes the structure of systems more unified and facilitates systems' maintenance; however, a decentralized structure of interacting systems was not achieved. The result is a set of information islands, as the systems have no computerized interaction. One might have to enter basic data for a certain patient several times, and there is no possibility for a doctor to get a total view of a patient's medical record.

Also, Professor Börje Langefors addressed the need for fast development of local systems. The tool he used, together with Ola Langefors, was Dataflex (a simple high level programming language) and he published a report [11] on Prototyping with Dataflex. In the work by Langefors we however find a more structured approach, than just to develop local systems. Already in 1982, he presented a report on Four Cases with Decentralized Management and Local

Computers [9] and in another report [10] he discussed Integrated versus Decentralized Information Resource Management. The problem he addressed was to find a solution both to support users with local information from local systems, which exchange what Langefors calls translocal information, and at the same time maintain independence between these local systems. This gives us a reason to analyze independence between systems more in depth.

5. The Problem and Principle of Independence between Systems

Starting in the general principles for decentralization used in this paper, we define the basic demand for independence: Changes in one system (handling local information that is relevant for a certain business unit), should not cause changes in other systems that support other business units.

5.1 Systems Delineation

One condition for independence is that each local system is delineated to support only a specified business unit (an area of responsibility, for instance a business process). A simple example is given in Figure 1.
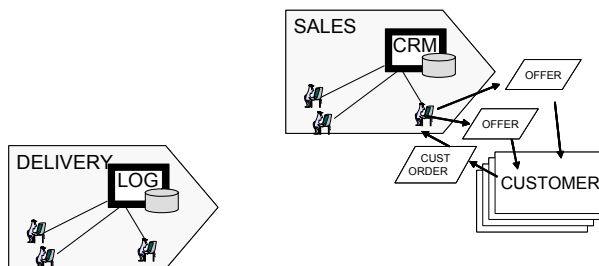

Fig. 1. Local Information Systems
for the Sales process and for the Delivery process

The system CRM (Customer Relationships Management) supports the Sales process from the very beginning of a customer relation and holds all necessary information for the sales process until the company invoices the deliveries. The LOGistics system contains modules for warehousing and delivery planning, and supports logistics until a delivery order is completed.

5.2 Systems Interaction

These two systems CRM and LOG are independent of each other, but so far they do not interact, they are still two information islands. The challenge is to find the manner in which the systems can interact as two decentralized systems with

maintained independence.   To address this problem, we must answer two questions:

- o What translocal information should be exchanged?
- o How must translocal information be described and exchanged in order to maintain independence?

The general answer to the first question is that translocal information is the ordinary information exchanged between different business units to perform business. In the example, it is necessary for the Sales process to know what is available in the warehouse (including expected deliveries into the warehouse), called DISP QUANT in Figure 2, otherwise the company cannot give an offer with a delivery date to the customer. If the customer accepts the offer, the sales process must send a delivery order (DEL ORDER in Figure 2) to the Delivery process.   When delivery is completed this must be reported back to the Sales process (DEL REP), as this information is the basis for the invoice to be sent from Sales to the customer. This translocal information constitutes three computerized messages as shown in Figure 2.
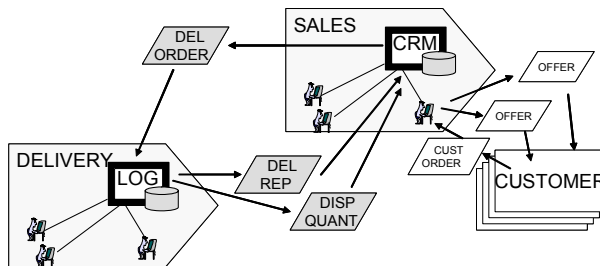

Fig. 2. Translocal Information to be exchanged
as computerized messages between systems

We can describe these messages in detail through business analysis; the point is that we can do this without any knowledge about data storage or processing in each one of the two systems. This enterprise based systems interaction (discussed in [3] and [4]) takes us to the principle of independence that states:

> *Interaction between decentralized systems should be described independent of the inner structure in each system, in order to maintain independence between the systems.*

We must then give the messages for interaction a technical format and transfer the data between systems with some message handling system in the company's IT-infrastructure. Standardized middleware is available for this process.

This approach represents a different way for interaction between systems, a change from access thinking to messaging. If we want to maintain independence, it is not enough to 'open the books' and show the data structures in each system. We must instead bridge the Interaction through messages based on business relations.   It is not a question of understanding data in other systems; it is a question of understanding what translocal information to be transferred between

the systems. The different data structures are local and they connect to the messages through mapping mechanisms, which must be developed and maintained for each system. If the inner structure in one system is changed, then we might have to change the mapping mechanism in that very system to maintain proper interaction. But, there will be no change in any other system as long as the business relation stands.

Message interaction ensures consistency in the structure. Customer information in the LOG system is, for instance, based on (derived from) computerized messages DEL ORDER, delivery orders. There is no manual updating of information on customers in the LOG system. Because of this, a certain customer in the CRM system may not yet be registered in the LOG system, as there has not been any delivery order so far. This is not inconsistency; it is just normal business interaction.

In the 1980s, the concept for decentralized systems using the principle of independence became part of business analysis [3] and later on presented as Verkssamhetsbaserad Systemstrukturering (Business Based Structures of Information Systems). In 1986, it was part of a presentation at the NordData Conference [4] and actually awarded the Nordic Prize that year.

The paradox is that this principle for systems interaction is quite natural in inter-organizational systems interaction (EDI-solutions were already used in the early 1980s) but within the company, this way of thinking for many years was (and sometimes still is) blocked by centralized principles like: *Store data once, General access to all stored data,* and behind that: *Total Data Models.* The concept of messaging for transfer of information between different systems within a company was not widely discussed or used until the end of the century.

6. The Situation at the End of the Last Century

The total result from IT development in the last century is not impressive from a structural point of view. Even if companies applied a centralized approach, in most cases it was not possible to use just one system in the company, or even to develop and maintain a total data model. When new systems came on board in the structure, they used principles for accessing and updating different databases, which in many companies and organizations ended up in complex structures of systems, often referred to as a "spaghetti structure". If one system changed, it affected many other systems. The structure gradually grew unstable.

The centralized approach often resulted in big systems and projects running for several years. Another problem had been the limited possibilities for users to have any real impact on the project. Researchers as Enid Mumford already addressed this problem in the 1970s and presented models for participative design [13], pointing out the value in the organization of user oriented development. However, this approach did not work so well in centralized projects.

On the other hand, in many cases the decentralized approach ended up in a number of separate systems, without any computerized interaction, with multiple

updating, inconsistency, and unclear responsibilities for these different Information Islands.

Börje Langefors gave 1995 in the book Essays on Infology [12, p 159] a really good summary of the situation, which is here quoted:

> *Infology and Decentralization*
> *Information systems theory has, since its beginning in the early 1960s, been facing a contradiction. One of its main visions was that data in the system had to be available to "everybody" (Langefors 1961,#29, 1963, # 37). But it was soon detected that a set of data does not inform everybody (the "infological equation." Langefors 1966, #1). It had to be concluded that efficiently designed information systems had to be structured as a network of communicating more or less separate subsystems based on local data systems. This insight took a surprisingly long time to gain recognition in the data profession, as well as, for instance, in accounting.*
>
> *Even when, in the 1980s, small local systems came to be fairly common, this was in many cases due to the emergence of inexpensive micro-computers, rather than to an understanding of the often local character of data.*
>
> *And, with the maturing of the technology of connecting small computers to form networks, one has begun again to talk about making all data accessible to everybody. We conclude that there is still lacking the understanding that some data are only intelligible to restricted groups of people. This suggests that there is need for case studies, in order to reach and disseminate a more concrete understanding of this aspect.*
>
> *It is often stated, e.g. by data managers, that the popping up of isolated local systems will lead to chaos. Leaving aside the fact that some amount of chaos may be useful, we point out that keeping isolated such data as are in in any case unintelligible outside a limited context can't by itself generate chaos. Of course, such data as have to be used in several locations, but those only, must be subject to integrated management – but this should not be done indiscriminately.*

From an international perspective during the last decade of the century, we can see that some researchers maintained the centralized approach while others questioned it. One example is Allen and Boynton who advocate a centralized structure, mainly from an IT point of view, and they argue that central structures are flexible since one can make changes centrally [1]. On the other hand, Bacon analyzes how systems decentralization can be derived from organizational principles [2].

7. What Can We Learn From History?

A rephrased question could be: When will we ever learn from history? Failures to implement real huge centralized systems in the last century such as the RAINBOW project, aiming at a total world wide system for the company TetraPak were followed by new failures like the SIRIUS project that tried to replace forty-two systems in the Swedish Defense organization with one big system. Worst of all, soon after the termination of this big project (which was closed down by the minister of defense) a proposal for a new total system, this time an ERP system, was given from the IT-specialists. It seems as if the fight

between decentralization and centralization continues as with totally integrated ERP systems versus "best-of-breed" systems.

Another example on different competitive views on centralization /decentralization can be observed within the healthcare area. Many hundred separate systems containing medical records were implemented gradually over the last three to four decades in different medical clinics, healthcare centers, and medical support units. As a patient in Sweden is free to visit different healthcare units, there is a need to give the treating doctor a more complete view of a patient's different medical records. One alternative (the centralized alternative as proposed by some IT-architects) would be to gather all information from local treatments in a total centralized national database for medical records easily accessible by anyone with authorization. The other alternative is to create only an overview of patients´ visits in a coordinating system and then base systems interaction on transfer of requested medical records between existing local systems (the decentralized alternative). There is a huge difference in investments and in impact on the healthcare organization between the two alternatives.

The main problem is that the alternatives so seldom are evaluated. The focus for the IT specialist is generally on technical solutions for total access, not on the alignment between the organization and the structure of information systems, or on what centralization /decentralization ends up with in a long-term perspective.

In the connected society, we definitely have the technology to create suitable structures of systems, but there is no real interest to solve or even to discuss explicitly the balancing problem between centralization and decentralization. Centralists (often IT specialists) keep on proposing centralized solutions based on the basic believe that centralization is necessary to create order; while decentralized proposals, more aligned with objectives out in the organization, such as freedom of action to meet changes through step by step development of IT support, often come from local management. Sometimes a centralized alternative is chosen, sometimes a decentralized alternative is decided. A general observation is however that there is very seldom any real analysis of the alternatives behind the decision. What really must be centralized, what can more efficiently be handled locally, and how local systems can interact with maintained independence are questions seldom under discussion.

8. Conclusion

The conclusion is that theories, management, and practice for decisions on centralization or decentralization of information systems must be further developed. Langefors pointed out the "*lack of understanding that some data must be local, as these data are only intelligible to restricted groups of people*". This is for sure true, but to my opinion *understanding* is not enough. A conscious management and control of centralization or decentralization in IT support is a vital question in the company or the organization, which is not a task that can be handled only by IT-specialists. The structure of information systems should be aligned with the business structure.

There is a need for business oriented IT management.

If the last century brought us extended knowledge about the outcomes from centralization and decentralization, including consequences and failures, then perhaps a main question for this century is how to use this knowledge to learn how to manage, balance, and control centralization and decentralization in a long-term perspective.

References

[1] Allen, B.R. & Boynton, A.C. (1991). Information Architecture: In Search of Efficient Flexibility. *MIS Quarterly/December*.

[2] Bacon, C.J. (1990). Organizational Principles of Systems Decentralization. *Journal of Information Technology, vol.5*.

[3] Hugoson, M-Å. ,Hesselmark, O. & Grubbström, A. (1983). *MBI-metoden, en metod för verksamhetsanalys*. Studentlitteratur, Lund.

[4] Hugoson, M-Å. (1986). Verksamhetsbaserad systemstrukturering. *NordData, Stockholm 1986*

[5] Iivari, J. (1989). Levels of Abstraction as a Conceptual Framework for an Information system. *Proceedings on the IFIP TC 8 /WG 8.1 Working Conference on Information Systems Concept.* Namur, Belgium 1989.

[6] Jardine, D.A., Yazid, S. (1989). Integration of Information Submodels. *Proceedings on the IFIP TC 8 /WG 8.1 Working Conference on Information Systems Concept.* Namur, Belgium 1989.

[7] Langefors, B. (1967). Directive Information for Systems Control. *The analysis of Business Systems,* Glerup, Lund.

[8] Langefors, B. (1968). *System för företagsstyrning.* Studentlitteratur, Lund.

[9] Langefors, B. (1982). Four Cases with Decentralized Management and Local Computers. *Research report, Department of Computer Science,* 82:09, Chalmers University of Technology & University of Gothenburg 1984.

[10] Langefors, B. (1983). Integrated versus Decentralized Information Resource Management. *Research report, Department of Computer Science,* 83:09, Chalmers University of Technology & University of Gothenburg 1984.

[11] Langefors, B. (1984). System development using Prototyping with Dataflex. *Research report, Department of Computer Science,* 84:03, Chalmers University of Technology & University of Gothenburg 1984.

[12] Langefors, B. (1995). *Essays on Infology.* Studentlitteratur, Lund.

[13] Mumford, E. (1971). *Systems design for people, Economic evaluation of computer based systems.* Book 3, the National Computer Center (NCC), Manchester.