



HAL
open science

Analysis of Organisational Dependency for Urbanism of Information Systems

Julie Chapron, Xavier Boucher, Patrick Burlat, Pierre Lebrun

► **To cite this version:**

Julie Chapron, Xavier Boucher, Patrick Burlat, Pierre Lebrun. Analysis of Organisational Dependency for Urbanism of Information Systems. *International Journal of Computer Integrated Manufacturing*, 2008, 21 (03), pp.337-350. 10.1080/09511920701370761 . hal-00513392

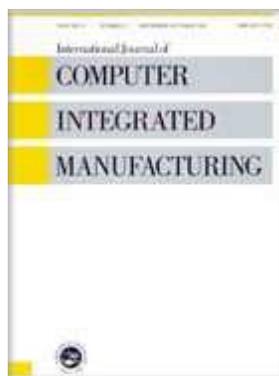
HAL Id: hal-00513392

<https://hal.science/hal-00513392>

Submitted on 1 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Analysis of Organisational Dependency for Urbanism of Information Systems

Journal:	<i>International Journal of Computer Integrated Manufacturing</i>
Manuscript ID:	TCIM-2006-IJCIM-0138.R2
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	23-Mar-2007
Complete List of Authors:	chapron, julie; STMicroelectronics Boucher, Xavier; Ecole Nationale Supérieure des Mines de Saint-Etienne, G2I Burlat, Patrick; Ecole Nationale Supérieure des Mines de Saint-Etienne, G2I Lebrun, Pierre; STMicroelectronics
Keywords:	CLUSTERING, ENTERPRISE MODELLING, INFORMATION SYSTEMS
Keywords (user):	dependencies, coordination



Analysis of Organisational Dependency for Urbanism of Information Systems

J. CHAPRON‡, X. BOUCHER*†, P. BURLAT†, P. LEBRUN‡

† Ecole Nationale Supérieure des Mines de St Etienne, France

‡ ST Microelectronics, France

Correspondence Ecole Nationale Supérieure des Mines de St Etienne, Centre G2I,
158, cours Fauriel, 42023 Saint-Etienne cedex 2, France
boucher@emse.fr

Abstract. Information Systems Urbanism is a method providing new solutions to manage the complexity of Information System (I.S.) transformations. This paper expands the urbanism approach by integrating an organisational analysis. Moreover a scientific approach is provided to support the complex decisions involving Information Systems change. For this purpose, dependencies among information processes are formalised. Dependency is based on three complementary dimensions: actors, material resources, information. These dependencies are used to generate process clusters. The mathematical approach used to create several cluster maps is explained, as well as the use of these maps for organisational diagnoses. The goal is to provide support to decision-makers for better change management. Finally, a case study in a manufacturing plant in micro-electronics shows the feasibility of the approach.

Keywords: dependencies, processes, clustering, evolution management, coordination

1. Introduction

In the field of manufacturing science, the ability to change has been identified as a crucial issue in the recent years. With a holistic vision for the future of manufacturing science, the report (Manufuture, 2003) emphasises the need for agility and management

Organisational Urbanism of Information Systems

of organisational transformations, with manufacturing businesses having “to make continuous reassessments of their core strengths and competencies”. Reconfigurable manufacturing systems (Harrison *et al.*, 2006) enhance the responsiveness to customer requirements. Agile manufacturing aims at providing rapid change over between manufacturing processes (Fujii *et al.*, 2006). However, such adaptability of technological systems will be fully efficient if they are complemented by the agility of information systems. Furthermore, agility requires the ability of managers to ensure the coherence of their strategy with industrial constraints at the tactical and operational levels of organisation.

From this perspective, the paper focuses on the management of information system transformations. Information system (I.S.) changes are directly induced and constrained by changes in the manufacturing firms’ strategies, technologies, or organisation of processes. This issue is especially relevant in the microelectronic domain, which has to adapt competitively to highly reactive markets. Indeed, these companies are subjected to a fluctuating worldwide demand, and must respect constraints generated by high levels of both process and information technologies. This research is the result of the collaboration between the Ecole des Mines de Saint Etienne and the firm STMicroelectronics, resulting in the PhD thesis (Chapron, 2006).

Herein, a process-oriented approach is used to present new tools for change management applied to information systems, which is called “Organisational Urbanism

of Information Systems”[†]. The objective is to take into account organisational factors when one has to decide on transforming the information system, thus not restraining the decision scope by considering only technical factors. To implement this approach it is necessary to analyse rigorously and then formalise the dependency links between processes, which could have an impact on organisational changes.

In section 2, the scientific positioning of the paper is explained and the objectives of the method are defined. The notion of “Organisational Urbanism” is also presented. Then, at the heart of the article, section 3 presents three complementary dimensions that have been chosen for process dependency and the measures of dependency are detailed. The selection of dependency dimensions is based both on the expertise of the industrial partners and on a state of the art of scientific literature. Section 4 focuses on the use of dependency links through a clustering method. Finally, section 5 applies this approach to a STMicroelectronics case study, and outlines the possibilities of organisational diagnosis using the clustering results.

2. Scientific positioning and objectives of the « Organisational Urbanism”

2.1. From Information System Urbanism towards Organisational Urbanism

Few formal approaches are available to answer, with a rigorous and scientific method, the joint transformation of the organisations and their information systems. The first contributions in this perspective are to be found in the field of enterprise modelling: in the domain of automatic control, (Malhéné, 2000) represents the evolution trajectory of

[†] Please note that “Urbanism” is used in the paper with the meaning of “City planning”.

Organisational Urbanism of Information Systems

a firm ; in the field of enterprise engineering the concept of “organisation life-cycle” aims at managing enterprise transformations (Williams *et al.*, 1998), (GERAM, 2000) ; this idea was also recently expanded by innovative research on virtual firms (Camarinha-Matos and Afsarmanesh, 1998), (Strader *et al.*, 1998), (Hoffman and Schlosser, 2001). However, these approaches only provide a partial answer concerning organisational flexibility (Dursun and Perakath, 2003).

In the field of Information System Science, the industrial approach of “Information Systems Urbanism” has been recently developed. Urbanism has the objective to make an existing information system more adaptable to the various types of transformations which can occur within an organisation. The current urbanism methodology was essentially structured via the implementation of industrial experiences (Le Roux, 2004), (Club-Urba, 2003), (Papadacci and Salinesi, 2005), (Chelli, 2003). This methodology presents the basic concepts necessary to implement information system adaptability (Longépé, 2001), (Carvalho, 2002). The urbanism procedure has been structured into four well-differentiated levels, to ensure the consistence between the strategy, internal organisation, and information technologies of a firm.

This paper aims at expanding the current urbanism approach by better integrating organisational factors. Indeed, current methods of Information System Urbanism lack scientific tools (i) to manage the organisational view, and (ii) to analyse objectively the consistence between the organisational and technological levels (Chelli, 2003).

From this perspective, the overall information system needs to be differentiated from its

computerised sub-system. Referring to the definition provided by (Vernadat, 1999), the information system involves not only a technological but also an organisational dimension. To refer to the information system independently of the technology, the term “*information processes*” will designate information operations without considering whether or not they are technologically implemented. Besides, the “*computerised system*” represents the technological sub-system. The term “*computerised information processes*” will represent automated data treatments.

The authors’ contribution to the field of urbanism consists in implementing new urbanism mechanisms and tools at the organisational level. This approach has been called “Organisational Urbanism”, and focuses notably on the following objectives:

- Defining a modular and generic framework to provide managers with a decision-support to manage information system transformations. The organisational urbanism of business processes will be based on criteria clearly distinct from technological constraints.
- Analysing scientifically, the consistence between organisational and technological urbanism. This issue has been identified as a key point in current urbanism methodologies (Longép , 2001), (Chelli, 2003).

Based on a process-oriented characterisation of the information system, Organisational Urbanism intends to offer a new view on the organisation and ways to transform it. As we will explain hereafter, the approach is based on the identification of information process clusters, with three main goals: (i) generate relevant clusters of processes,

Organisational Urbanism of Information Systems

making it easier to manage changes ; (ii) analyse these clusters to define general transformation targets and trajectories ; and (iii) manage future projects in coherence with this overall cluster analysis.

For a complete vision of organisational urbanism, the reader can refer to (Chapron, 2006). This paper only focuses on the clustering approach, which is the heart of the method, and the direct use of clusters for organisational diagnosis. Organisational Urbanism requires generating clusters which will later constitute relevant organisational units to manage changes. The primary objective is to manage together and within the borders of a given cluster, most of the changes which concern each process. Each cluster gathers highly interdependent information processes, and thus constitutes a basic and consistent enterprise entity for the deployment of changes. The clustering criteria will be defined thanks to the study of information process dependencies. Therefore, carefully defining the dependency links used to generate process clusters is important, as emphasised in the following sections.

2.2. Scientific literature on process dependencies

Following (Malone and Crownston, 1994) and (Crownston, 1997), who state that enterprise coordination can be managed through process and entity inter-dependencies, our research is based on the hypothesis that the coordination and then the transformation of the information system can be managed through the direct and indirect interactions between processes. Indeed, every coordination mechanism or every interaction between processes either results from or induces interdependency links. Complementary research

Organisational Urbanism of Information Systems

works like (Hee-Wong, 2000) or (Pollalis, 2003) also confirm that, to manage the evolution of processes, it is necessary to identify the trans-organisational processes, rationalise the flows and supervise their dependencies.

In their coordination theory Crowston and Malone define the coordination as the management of interdependencies among activities or processes (Malone and Crowston, 1994), (Crowston, 1997). Coordination mechanisms are considered as key components of organisational structures and they can be managed through several types of dependencies among tasks and sub-tasks. Generally, the dependencies are considered at the level of the tasks, and always linked to shared-resources.

Coordination has also been deeply studied in the context of product design and concurrent engineering. In this perspective, (Kusiak and Wang, 1993) and (Chen and Lin, 2003) propose a re-organisation of product design processes based on task dependencies. Other process-based approaches like Business Process Reengineering (BPR) include an analysis of dependencies between activities (Hwang and Yang, 2002).

All these approaches advocate that to manage the transformation of a system and to ensure coordination in the long run, studying the dependencies among the basic components is necessary. However, the link between dependencies and the different types of “resources” used by the activities or processes has not yet been detailed.

Resource-based dependencies have been studied in other fields. Notably, the management of information systems has induced research on the automation of information processes. The development of workflows has encouraged studies of tasks

dependencies to implement automated data exchanges among applications. (Attie *et al.*, 1993), (Carter *et al.*, 2003) have analysed task dependencies by taking into account information flows among the resources used for each task. (Lim and Park, 2005) uses event dependencies to improve coordination among the sub-components of an information system. (Albino *et al.*, 2002) go a step further, with the identification of dependencies among pieces of information manipulated by the tasks, in order to detect critical points of coordination among the processes. Based on similar studies of information flow dependencies, the integration of applications turns out to be a compelling objective (Castano *et al.*, 1999). For instance, (Lee, 1998) applies informational dependencies to determine clusters of processes defined by the pieces of information in use.

Complementary approaches consider the dependencies created by interactions at the level of the actors involved in the processes. Actually, human actors can also be considered as shared resources, generating dependencies among processes. Obviously, specific links between the actors also have to be considered. Moreover, human-centred coordination mechanisms are necessary to ensure work efficiency. In their socio-technical approach, (Yu and Mylopoulos, 1997) show that actors of a firm depend on each other: such dependencies must be managed in order to be more reactive in change situations. Dahlgren and Söderlund (2003) study the impact of the structure of the actors on the output of a firm's processes. In the field of information systems Castro *et al.* (2002) also use actor dependencies for a better definition of information system

requirements.

All these contributions concerning task, activity or process dependencies underline several types of dependency links related to (i) technological resources, (ii) shared information, or (iii) human actors, when considering activities within their socio-technical environment. Based on these conclusions, section 3 specifies the dependency dimensions considered in this approach.

3. Formalisation of process dependencies

3.1. Selection of dependency dimensions

The dependency criteria needed for this research must comply with the change management objective. The goal is to generate clusters of processes whose level of interdependence justifies managing internal changes altogether. Furthermore, since change is dealt with at the organisational level, organisational dimensions have to be considered. Both the scientific state of the art provided by the previous section and a study in the field done by STMicroelectronics has led to the selection of the three following dimensions of dependency: shared technological resources, shared information, or actor relationships. As a matter of fact, the “Organisational Urbanism” method was first designed to fulfil the specific requirements and needs of the company STMicroelectronics. Based on the expertise of information system specialists in the company, each of the three dimensions above has been validated as essential to answer

Organisational Urbanism of Information Systems

the in-house needs of change management.

As a consequence the three aforementioned dimensions have been implemented to evaluate process dependencies, relative to:

- human actor relationships, called Actor Dependency D_A ;
- use of technological resources, called Resource Dependency D_{RE} ;
- information sharing, called Information Dependency D_I .

Therefore, there is a need to develop an evaluation of these main dependencies, using basic measures of more elementary dependency links.

Moreover, process dependencies will be defined on two levels. Following Malone's coordination theory, most of the current dependency approaches only consider direct dependency among tasks or activities. Organisational Urbanism does consider the direct dependency, at the first stage. This dependency is induced by the sharing of the three "dependency-objects"[†] corresponding to the three selected dimensions: shared actors, technological resources, or information. But in addition, at the second stage, indirect dependencies are considered: dependencies among the dependency-objects themselves, even when these objects are not directly shared by the processes[‡].

Thus, in the following sections, dependency measures will be formalised covering both

[†] Only necessary to simplify these explanations, the 3 following generic notions, "actors", "technological resources" and "pieces of information", will be named "dependency-objects".

[‡] For instance, there is an indirect dependency between two processes when one actor of the first process has a hierarchical link with an actor of the second. Since there is no shared actor between the two processes, there is no direct actor dependency, but an indirect dependency.

the three dimensions selected and the direct and indirect dependencies. The three dependency dimensions also define three complementary axes of action to improve coordination. Indeed, the identification of dependencies can also detect constraints and obstacles that confront any system change.

3.2. Formalisation of the dependency concept

In the context of change management, two objects are dependent when the modification of one has an impact on the functioning or the transformation of the other (Chapron, 2006). Given two processes or sets of processes C_1 and C_2 , the goal is to evaluate their dependency $dep(C_1, C_2)$, using a real number in the interval $[0,1]^\dagger$. However the method only requires modelling mutual dependency and not causal dependency. The following restrictive hypothesis can be made: “*The dependency between processes is modelled by elementary dependency-links which are neither quantified, nor oriented*”. This hypothesis implies that no causal information on the dependency link is used, and that the quantification of the dependency will depend only on the number of elementary dependency links, and not on their intensity. The first justification is that, for the purpose of change management, generating clusters of processes presenting strong mutual dependencies is required, and this mutual dependency is independent of the orientation of the causal link. Secondly, a reliable quantification of each elementary dependency links would require a very precise level of detail for the process models. Considering the large number of dependency links, such a detailed modelling would

[†] In the interval $[0,1]$, 0 means no dependency at all and 1 means full dependency.

make the method non applicable to the industrial context. As a consequence of the hypothesis, the mutual dependency is considered here as a restrictive version of dependency[†], which fits with the property of symmetry $dep(C_1, C_2) = dep(C_2, C_1)$.

In the three following sections, the formalisation and evaluation of the main dependencies actor D_A , resource D_{RE} , and information D_I will be presented. For each of them, all the elementary dependency links will be identified. These elementary dependency links are not quantified: they are only valued 0 or 1 depending on their existence (as a consequence of the preceding hypothesis). Then section 3.4 will explain the aggregation of the elementary dependencies to provide the final evaluation of D_A , D_{RE} , and D_I .

3.2.1. Actor dependency - D_A

The actor dependency only concerns human actors, while the dependencies among technological resources are represented with D_{RE} . To formalise D_A , the three elementary dependency links are defined: (a) actor sharing, (b) structural dependency and (c) role dependency.

Actor sharing: two processes P_n and P_m are dependent in terms of actor sharing when they have in common the same person. For each pair of actors A_i, A_j mobilised, respectively by the processes P_n and P_m , the function expressing if the actors A_i and A_j are the same is noted $d_a(A_i, A_j)$ (Binary value 1, if the actors are identical, otherwise 0).

[†] Further research could also develop a formalisation of causal dependency using non-symmetric measures.

Organisational Urbanism of Information Systems

The value of d_a is obtained for each combination of pairs of actors mobilised by P_n and P_m . The aggregation function used to define the final dependency between the processes will be specified in the section 3.4.

Structural dependency: two processes P_n and P_m are dependent when they involve distinct actors who are hierarchically linked in the enterprise organisational chart. The structural dependency between two actors is noted as $d_s(A_i, A_j)$. d_s is measured in reference to the enterprise organisational chart. Conventionally, $\text{depth}(A_i)$ is the depth of the node A_i in the hierarchy (i.e. the number of edges on the path from the root of the tree to the node A_i). Given any two actors, the *Lowest Common Ancestor* - $\text{LCA}(A_i, A_j)$ - is the node of greatest depth that is an ancestor of both actors A_i and A_j .

If $\text{Max}(\text{Depth}(A_1), \text{Depth}(A_2)) - \text{Depth}(\text{LCA}(A_1, A_2)) \leq \alpha$ then $d_s(A_1, A_2) = 1$ else $d_s(A_1, A_2) = 0$. The variable α represents the number of edges in the organisation chart separating two actors from their *Lowest Common Ancestor*, from wherein the two actors are no longer considered dependent. This formula can be adapted to the individual structure of an enterprise (depending on the number of hierarchy levels).

Role dependency: two processes P_n and P_m are dependent when they involve exactly the same role[†]. For each pair of actors A_i, A_j mobilised by the processes, the function expressing if the actors A_i and A_j perform the same role is noted $d_r(A_i, A_j)$ (Binary value, 1 if the roles are the same, otherwise 0). The value of d_r is obtained for each combination of pairs of actors mobilised by P_n and P_m .

[†] Here, we do not identify similar roles but only identical ones. This level of precision is sufficient with regards to the constraints of the case study, as underlined in the results (section 5).

Organisational Urbanism of Information Systems

Thus, the notion of role applied to human actors is used. Referring to a state of the art on this subject -e.g. (Ould, 1995), (Archimate Project, 2003), (ISO 19440, 2005)-, the concept of “role” has been defined in (Chapron, 2006) as *the mission and scope of responsibility of an actor, in a specific context*. The role is a generic concept used to describe business activity domains by consistent sets of missions and competencies required. Considering change management, the role concept highlights the domains under common responsibility and, therefore, reveals processes which depend greatly on actor competency.

3.2.2. Technological Resource dependency - D_{RE}

Since this study only focuses on information processes, the technological resources involved in the processes are computer applications. The link between processes and hardware is no longer considered as a constraint at the process level. The high performance of the hardware allows most types of process implementation. Thus, two types of resource dependency will be considered:

Software sharing: two processes P_n and P_m are dependent when they have in common the same software. For each pair of applications S_i, S_j used respectively by the processes P_n and P_m , the function expressing if the applications are the same or not, is noted $d_{so}(S_i, S_j)$, taking a binary value, 1 if identical, 0 otherwise. The aggregation defining the final dependency between the processes will be specified in the section 3.4.

Data flow dependency: two processes using different applications are dependent if there are significant data flows exchanges between these applications. For each pair of

Organisational Urbanism of Information Systems

applications, S_i , S_j used respectively by the processes P_n and P_m , the data flow dependency is noted $d_f(S_i, S_j)$.

$d_f(S_i, S_j)$ can be measured using the outputs of the standard Information System Urbanism method (not the “Organisational” Urbanism). Regardless of enterprise processes, the Information System Urbanism generates software clusters based on the data flow intensity between the applications. These software clusters will be used here to define the data flow dependency. Two applications S_i , S_j belonging to the same cluster will have a dependency of 1 (full dependency), 0 otherwise[†].

3.2.3. Information dependency – D_I

The information dependency will be based on the concept of *business object*, defined in (Chapron, 2006) as a set of information manipulated by information processes and, whose meaning is unique and shared by all the business actors irrespective of the process. The identification of the business objects in use, and the specification of their boundaries require a contextual expertise of the firm. Such expertise is also needed to model the processes and their boundaries, as underlined in section 5.

The use of the business object concept takes into account the semantics of the information used by the processes regardless of IT implementation, operational data structure and data life cycle. Thus, information dependency is complementary to data flow dependency which only focuses on IT implementation constraints and data structures. Information dependency is evaluated with two elementary dependency links:

Business object sharing: two processes P_n and P_m are dependent if they share a common

[†] Here again a more precise measure would not have provided the method with any real added-value.

Organisational Urbanism of Information Systems

business object. For each pair of business objects O_i and O_j used by the processes P_n and P_m , the function expressing if the business objects are the same is noted $d_o(O_i, O_j)$. The aggregation function used to calculate the final dependency between the processes will be specified in the section 3.4.

The event dependency: two distinct processes are dependent if a business object they share induces a logical coordination between these two processes or, in other words, when their logical sequencing is constrained. This logical coordination covers synchronisation constraints, in the case when synchronism is required at the logical level, and not only induced by an implementation constraint. This dependency is noted $d_{ev}(O_i, O_j)$ (Binary value, 2 if the business object is a logical coordination between the processes, 1 otherwise).

From a quantitative point of view, the event dependency will be introduced as one of the factors necessary to calculate business object sharing dependency. When a business object induces a logical coordination between two processes, this information is taken into account to modify the evaluation of the business sharing dependency. This is obtained with the aggregation function (weighting the business objects) – see §3.4.

The calculation of these elementary dependency links related to D_A , D_{RE} and D_I is summarised in Table 1. Note that the elementary dependency measures are relative to the dependency-objects and thus they have to be aggregated to provide an evaluation of dependency links at the process granularity level. The following section will explain the

Organisational Urbanism of Information Systems

aggregation mechanism to calculate D_A , D_{RE} and D_I .

[Insert Table 1 about here]

3.3 Aggregation of dependency measures

The objective here is to aggregate elementary dependency measures applied to the dependency objects (i.e d_s , d_r , d_a , d_{so} and d_f) in order to obtain the corresponding process dependency d_S , d_R , d_A , d_{SO} and d_F .

First, an aggregation function is applied to d_s and d_f to calculate d_S and d_F for two processes P_1 and P_2 .

$$\text{For } k = s \text{ or } f, \quad d_k(p_1, p_2) = \frac{\sum_{i=1}^n \sum_{j=1}^m d_k(O_i, O_j)}{n * m}$$

n and m are the numbers of dependency-objects O_i / O_j (e.g. “actors”) involved in the two processes P_1 and P_2 . The elementary dependency measures d_s or d_f take binary values 0 / 1, while the process dependencies d_S , d_R take real values in the interval [0,1].

In the case of elementary dependencies d_r , d_a and d_{so} each dependency-object of one process, say P_1 , can only depend on one dependency-object of the other process P_2 .

Therefore the dependency matrices among objects are simplified, and we can use a simplified aggregation function:

$$d_k(p_1, p_2) = \frac{\sum_{i=1}^n \sum_{j=1}^m d_k(O_i, O_j)}{\max(n, m)}$$

Finally, with “Business Object Sharing” d_o , a different function is used in order to apply the event dependency d_{ev} as a weight to re-enforce d_o .

Organisational Urbanism of Information Systems

$k = \max(i, j)$ and for formula described $k = i$, so $d_{ev}(O_k) = \max_{j=1..m} d_{ev}(O_i, O_j)$ then

$$d_o(p_1, p_2) = \frac{\sum_{k=1}^n d_{ev}(O_k) \cdot \max_{j=1..m}(d_o(k, j))}{\sum_{k=1}^n d_{ev}(O_k)}$$

Once the aggregation has been accomplished, to calculate all the process dependencies d_S , d_R , d_A , d_{SO} and d_F (and considering all the processes within the cope of the study), a classical weighted mean is used to provide the three main dependencies D_A , D_R and D_I . The weight coefficients (w_1 , w_2 , w_3 , w_4 , w_5) chosen for this aggregation can be fine tuned depending on the targeted diagnosis.

$$D_A = \frac{w_3 * d_A + w_4 * d_S + w_5 * d_R}{w_3 + w_4 + w_5}$$

$$D_{RE} = \frac{w_1 * d_{SO} + w_2 * d_F}{w_1 + w_2} \quad D_I = d_o$$

Figure 1 summarises the different steps of the dependency measure computation described above. The computation of the three main dependencies produces a process graph where the nodes represent the processes and the edges represent the dependency between the processes. This graph will be represented as a matrix and analysed to identify relevant clusters, as explained in the following section.

[Insert figure 1 about here]

4. Clustering of processes

Our approach will now use a clustering method in order to generate a representation of the Information System based on consistent and highly dependent sets of processes,

while taking into account change management criteria. These sets will constitute basic management units to implement changes. The diagnosis will emphasise the transformation consistence with regards to the existing systems. Transformation consistence should consider both the stability and the flexibility of the system.

Figure 2 presents an overview of the decision-support procedure. First, process maps - in their technological and organisational environment - are generated. These enterprise models contain all the company data required for the proceeding steps. These models are built upon a meta-model specified in (Chapron *et al.*, 2004). Data collected at this stage is used to evaluate the process dependencies, according to procedures in section 3. This step generates a process dependency graph. Then, mathematic clustering is applied to this dependency graph. The goal is to extract groups of processes which are coherent with change management criteria. Cluster maps provide concise and relevant information in order to supply decision-making supports.

[Insert figure 2 about here]

4.1. Selection of a clustering approach

Clustering consists in grouping objects, evaluated on a matrix of distance (or dissimilarity) between each pair of objects, into a limited number of homogeneous subsets. Clustering reveals a structure in a collection of objects. There is a large number of clustering approaches, mainly based on mathematical theories of graph analysis (Jain *et al.*, 99). Clustering algorithms fall into two broad groups: hierarchical classification and partitional clustering. Partitional algorithms have a strong constraint. Initially, they

Organisational Urbanism of Information Systems

require fixing the final number of clusters. Hierarchical algorithms produce clustering where sub-clusters can be contained in larger clusters. But, the distance between each pair of objects or object clusters has to be known, in order to build a partition.

The objective of the method here is to obtain distinct clusters of processes, considered as consistent organisational units to achieve change goals: all the processes within a given cluster have to be managed together when there is a situation of change. This objective induces the use of a hard, non incremental, and determinist method. Moreover, the measure of distance between two processes (i.e. dependency measure) is a non-oriented and aggregated distance (which means it is a unique criterion; two nodes of the graph are only connected with one non-oriented edge). The size of the data computed is quite small (even when considering hundreds of processes) compared to sophisticated algorithms specialised in the treatment of large databases. These criteria put forth the classification methods.

Two approaches characterise most of the classification methods: agglomerative algorithms (bottom-up) and divisive algorithms (top-down). Divisive methods have rarely been applied, especially because they consume more calculation-time than the agglomerative algorithm. Therefore, the Agglomerative Hierarchical Classification (AHC) was selected.

4.2. The clustering procedure

The dependencies defined in section 3 have to be calculated for each pair of processes considered, and result in a dependency matrix providing the inputs for mathematical

analysis.

Then, before applying the AHC algorithm, a multi-dimensional scaling method (MDS) is used. The goal is to detect meaningful underlying dimensions that explain the observed distance between the investigated objects. In the case of this method, the MDS plays the role of pre-treatment, like a filter while preserving the useful information. The result is a better partition, by creating more homogeneous clusters when applying the classification method (Lebart *et al.*, 1997).

Thirdly, the AHP method is finally applied. The principle of the classification algorithm is to create a partition at each step. This partition is obtained by aggregating the nearest elements two by two. What are called “elements” can be the objects to classify as well as the clusters generated by the algorithm (Berkhin, 2002). A partition is a set of clusters defined at a certain level in the tree. The algorithm does not only give a partition of the elements but a partition hierarchy which is usually called a dendrogram (see Figure 3). The dendrogram is then cut to get the final partition, which will be represented as a cluster map.

[insert here figure 3]

The mathematical criteria used to position the final “cut” of the dendrogram has to be precisely defined. Here, the partitions are made to minimise the information loss associated with each cluster. In this objective, Ward’s hierarchical method was used (Ward, 1963). Two clusters are aggregated in such way that the inter-class inertia is the least, so that the classes remain homogeneous.

Organisational Urbanism of Information Systems

More details on algorithms and mathematic criteria in use can be found in (Chapron, 2006).

4.3. Several cluster maps

As explained previously, three main dependencies between processes are considered. Each dependency (actor, resource, information) can be used in the clustering method to provide a specific cluster map. Furthermore, these three types of dependencies can also be aggregated in a global dependency which supplies a cluster map of the aggregated dependency between the processes of the enterprise.

Thus the approach can produce four different maps, either applied to the existing or targeted information system. Which map will be used depends on the point of view of the manager. Each map can be used in the strategic and/or tactic approaches of the Organisational Urbanism methodology (see 2.2). These maps can also be compared one with another to highlight internal consistence problems, among the distinct dependencies. This will be illustrated in section 5. The added-value as decision-support tool is situated at the pragmatic level: (Williams and Lance, 1967) explains that it is not a question of reaching a true or false result, whether probable or improbable, but only profitable or non-profitable.

5. The case study – ST Microelectronics Company

5.1. Context and objectives of the case study

This approach was applied to the Franco-Italian electronic chip manufacturing company

Organisational Urbanism of Information Systems

ST- Microelectronics. Like any firm in this domain, STMicroelectronics is faced with frequent technological and organisational upheavals and has strong needs for the joint management of both the transformation of the information systems and the organisational structures. Indeed, this industry is characterised by a twofold challenge. First, the rapid evolution of semi-conductor technologies requires constant and heavy investment for a short product life cycle. Second, market fluctuations with 5 to 7 year cycles, make it difficult to forecast the production workload. Microelectronics is a highly specialised industry where IT tools are essential. With the manufacturing process of a wafer requiring more than 300 steps and production units of several thousand employees, the underlying information system has a very high degree of complexity. This complexity makes almost impossible the analysis and management of information systems without decision-support tools for change management.

On the operational level, the objective is to build a decision-support tool where the precise dependency and clustering calculations are automated and thus non-apparent to the end-user. The successive steps of calculation are embedded in the tool, which improves user-friendliness. The responsibility of the user is to input correct data on the enterprise and its processes. A meta-model specifying all the enterprise modelling concepts required to implement the approach was proposed in (Chapron, 2004). The enterprise models cover the processes, their organisational context, and technological environment. STMicroelectronics has chosen the software and modelling environment MEGA Process and MEGA Architecture® to implement the Meta-model. All the data

necessary to calculate the dependencies and generate the clusters are extracted from enterprise models.

This is a feasibility study. The objective was to test and apply the method on a limited number of processes, in order to test and validate the applicability of the different steps of the method. The experimental case presented gathers fifteen distinct processes. This first industrial test trial will illustrate the generation of cluster maps (section 5.2), as well as some Information System diagnoses that can be deduced (section 5.3). The company is currently deploying the method to an even larger portion of the firm.

5.2. Dependencies computation and clustering sequence

Table 2 presents the fifteen processes concerned.

[Insert table 2 about here]

The data collected on these processes is sufficient to obtain the cluster maps. In the following sections, the various steps which generate these maps will be described.

However, the data presented are anonymous to guarantee the enterprise confidentiality.

5.2.1. Dependency calculation steps

In this section, the procedure to calculate the *actor elementary dependency* is explained, using the example of two processes: P_2 – Process Change Review Board and P_7 – Scrap Management. Of course this procedure must be repeated for each additional pair of processes, as well as for the two other main dependencies: resource and information dependencies.

Organisational Urbanism of Information Systems

a. Actor structural dependency – d_s

The company organisation chart is used to identify the depth of the *Lowest Common Ancestor* for each pair of actors. Table 3 shows the calculation results for the structural dependency applied to each pair of actors. The structure of the organisational chart has been simplified for the case study. In this context, the value of the variable α is 4 (see 3.4), which means that two actors separated by more than four hierarchical levels are not considered as dependent.

[Insert table 3 about here]

The aggregation function (see section 3.5) applied to the actor structural dependency matrix provides the value 0.44.

b. Actor role dependency – d_R

The matrix of Table 4 identifies the different roles for the processes P_2 and P_7 .

[Insert table 4 about here]

The aggregation function applied to the actor role dependency matrix gives the value 0.37.

c. Actor sharing – d_A

The two processes P_2 and P_7 share two actors – The Device Engineer and the Process Engineer (see Table 3). The calculation procedure is the same as for the role dependency. The actors of the two processes are compared in a matrix, later aggregated using the same aggregation function as for role dependency (section 3.5). d_A takes the value 0.25.

d. Actor dependency – D_A

The actor main dependency aggregates the elementary dependencies using the following formula:

$$D_A = \frac{1 * d_A + 0,5 * d_S + 2 * d_R}{3,5} = 0,35$$

The coefficients of aggregation applied to the elementary dependencies must be defined depending on the objectives of the analysis. In this study, the roles are highlighted (high coefficient) because they represent functional dependencies between the processes.

Such dependencies are considered by the manager as crucial in situations of evolution.

5.2.2. Cluster maps building

The previous dependencies are calculated for the three main dependencies (actor - D_A , resource - D_R and information - D_I) and for all the processes. Table 5 represents the actor dependencies for all the processes studied:

[Insert table 5 about here]

This matrix provides the input for the mathematical clustering analysis. To execute the multi-dimensional scaling method and the agglomerative hierarchical classification, the software XLStat, developed by the company Addinsoft, was used. First, the multi-dimensional scaling method was applied to improve the quality of the data. Then the agglomerative hierarchical classification (AHC) was employed. In the AHC method, the calculation of the aggregation criterion is based on the Ward method. This approach calculates the loss of inertia for each new group composition. The cut on the dendrogram must be placed when the inertial loss is the most important. Figure 4

Organisational Urbanism of Information Systems

represents the results of the analysis for the actor as well as resource dependencies.

[Insert figure 4 about here]

5.3. Diagnosis applied to the case study

At this stage of the method, the initial goal has been reached: the generation of cluster maps. However, the important point is to facilitate the use of such maps for decision-supports applied to information systems diagnosis and transformation management. In (Chapron, 2006) several levels of decision-support have been developed. However, it is not possible to present all of them. Therefore, in section 5.3.1, only one example of diagnosis will be discussed (the analysis of consistence between distinct cluster maps). A brief overview of other decision-support systems will be given in section 5.3.2.

5.3.1. Consistence among the three cluster maps

The clustering sequence has generated a map for each of the three dependency dimensions. But, as shown in Figure 4 for the case study, these cluster maps are rather different. The Information System diagnosis presented below consists in comparing these three maps and identifying potential inconsistencies. The comparison can be automated. The potential inconsistencies have to be further analysed with contextual information from the company, to confirm if they point out (or not) any organisational weaknesses, which could later lead to remedial actions. An example of this diagnosis is presented via STMicroelectronics case study.

First, a potential organisational weakness can occur when different processes belong to the same cluster in both the actor and resource maps, but not in the information one.

Organisational Urbanism of Information Systems

These processes are dependent because of the actors and the software in use, but with a low level of information sharing. This situation has to be further analysed to check if there is a lack of information sharing or technology deployment for the processes. In the case study, this situation was identified for processes P₂, P₇, P₈. A contextual analysis by STMicroelectronics has confirmed the necessity to improve information technology: two major projects for processes P₂ and P₈ have been launched.

A second potential inconsistency appears when several processes belong to the same cluster in the resource map, but to different clusters in the information and actor maps. This situation could point out a lack of cohesion between the information systems and the actor/information organisation: the same software answers the needs corresponding to several of the enterprise's sectors. Sometimes, using different applications or specifically configured software can lead to performance gains. However, in the case of STMicroelectronics, this situation occurs with several processes because the Manufacturing Execution System, at the heart of the manufacturing information system, is transversal to a lot of processes. Therefore, this situation has been justified by the context and there is no inconsistency here.

A third example of potential inconsistency occurs when several processes belong to the same clusters in the actors and information maps, but not in the resource map. The process actors are interdependent from each other and share information, but they do not use common software. This situation can emphasise a need of information technology at the interface between several processes. In the case study, this situation was identified

Organisational Urbanism of Information Systems

for the pairs of processes $\{P_{11} ; P_{12}\}$ and $\{P_9 ; P_{10}\}$. However, in both cases, a software at the interface of the processes already existed, but the processes were not in the same resource cluster because of other dependency links considered during the clustering.

Thus, this situation is not inconsistent, and outlines the limits of the clustering support.

These three examples show the importance of the contextual factors. Each of them has led to a different conclusion: (i) confirmation of inconsistency, (ii) rejection because the situation is accepted, and (iii) rejection because the results of the clustering approach are not sufficient. Via these examples, three types of inconsistency have been put forward.

However, other types of inconsistencies have been identified in (Chapron, 2006), and their diagnosis has been systematised. A generic grid of most potential inconsistencies has been proposed. Each inconsistency corresponds to a given gap among the three cluster maps. The clusters are represented in a matrix so as to calculate these gaps.

Then, the generic grid is applied to signal potential inconsistencies. Of course the contextual analysis remains necessary to confirm these alarms.

5.3.2. Complementary diagnoses: perspectives

Other diagnoses, based on the use of the cluster maps, have been proposed in (Chapron, 2006). They are briefly introduced herein to illustrate the added-value of the approach.

At a primary level, some decision-support has been proposed to evaluate the existing system:

- The *consistence analysis* applied to the three dependency dimensions of the information system, which has been explained in the previous section.

Organisational Urbanism of Information Systems

- A *gap analysis*, between the structure of the computer system and the organisational clusters. The objective is notably to evaluate the consistence between organisational and technological urbanism.
- A *maturity analysis* on the clusters, used to build targets and transformation scenarios for the information system. Referring to the standard Capability Maturity Model (Carnegie Mellon University, 2002), the maturity of the process clusters is assessed. This information is further used to ease management of change.
- An *analysis of the strategic weight* of the clusters. Based on the evaluation of the contribution of each process to the strategy of the firm, the overall strategic weight of the clusters is calculated. This indicator complements the maturity level, in helping manage IT evolutions.

At another level, some complementary decision-support focuses on the comparison of the alternative solutions for the information system transformations:

- An *analysis of the quality* of the cluster maps. This analysis is based on the definition of three quality indicators: the level of separation between clusters of a same map, the internal degree of cohesion of a cluster, the number of clusters in a map. These indicators can evaluate the quality of the clustering. But overall, they are used to characterise the transformation between two configurations of the information system (pre and post change).
- An *analysis of the transformation effort*. Considering two configurations of the

Organisational Urbanism of Information Systems

same information system, the dependency gaps are identified among the cluster maps from one configuration to the other. Then, the two indicators “extension” and “amplitude” of the transformation are calculated. This information is useful in selecting among several change alternatives.

6. Conclusion and perspectives

This paper has presented the dependency approach and clustering method used to implement Organisational Urbanism of Information Systems. Based on models formalising knowledge of the business processes and their technical/organisational environment, Organisational Urbanism provides process cluster maps, which present new organisational views. The method tends to be generic. The models were developed for only one enterprise, but offer the necessary flexibility to be adapted to other enterprises and domains.

This method makes use of simplified enterprise representations. Of course, these models cannot integrate all the complexity and exhaustiveness of the existing interactions among the various components of the systems. The definition of the dependencies could thus be refined, by describing more precisely the existing links (for example, the inter-services links or the trans-functional co-ordinations in the company organisational chart). Nonetheless, the cluster maps are synthetic decision-support systems that managers can deploy and adapt to their needs. At each level, the different dependency maps provide tools that, when in the hands of the managers, can forecast highlight and

Organisational Urbanism of Information Systems

control the strengths and the weaknesses of the possible evolutions of the information system.

The implementation of this methodology on fifteen processes of the company ST-Microelectronics has confirmed the feasibility of the approach. The successive steps of calculation are embedded in the tools which generate the clusters. Meanwhile, the case study has underlined the fact that each implementation of the methodology requires fine tuning to fit with the internal enterprise context and structure. Beyond the feasibility study, the real benefits of the method will appear with an even broader application.

Indeed, Organisational Urbanism should provide a real added-value when the number of processes and interactions considered is high and justifies a systematic and mathematical analysis, as proposed. After the success of the first tests described here, ST-Microelectronics is currently generalising process modelling on several production sites. Furthermore, other decision-support systems using process clusters have been suggested, notably: (i) identifying the inconsistency between two information systems when any company has to be interconnected with an industrial partner, and (ii) managing inter-plants benchmarks, and process transfers from one production plant to another.

References

Albino, V., Pontrandolfo, P., Scozzi, B., Analysis of information flows to enhance the coordination of production processes. *International Journal of Production Economics*, 2002, **75**, 7–19.

Organisational Urbanism of Information Systems

Archimate Project, Concepts for Architectural Description, European Project, 2002. Available online at: <http://www.telin.nl/projecthome.cfm?id=48&language=en> (accessed 29 09 2005).

Attie, P.C., *et al.*, Specifying and Enforcing Intertask Dependencies, in *19th International Conference of VLDB*, 1993.

Berkhin P., Survey of clustering data mining techniques, Technical Report, *Accrue Software*, San José, Californie, 2002.

Camarinha-Matos L.M., Afsarmanesh H., Virtual Enterprises: Life cycle supporting tools and technologies, *Handbook of Life Cycle Engineering: Concepts, Tools and Techniques*, A. Molina, J. Sanchez, A. Kusiak (Eds.), pp. 1-30, 1998, (Chapman and Hall: London).

Carnegie Mellon University, Software Engineering Institute, Capability Maturity Model® Integration (CMMISM), Version 1.1, CMMISM for Software Engineering (CMMI-SW, V1.1) Staged Representation, *CMU/SEI-2002-TR-029*, Pittsburgh, 2002.

Carter, B.M., Lin, J.Y.-C., Orłowska, M.E., Customizing Internal Activity Behaviour for Flexible Process Enforcement, in *15th Australasian Database Conference*, 2003.

Carvalho, A., Evolution du Système d'Information fondée sur l'Urbanisation - Application au contexte hospitalier. PhD thesis, University Paris VI, 2002.

Castano, S., De Antonellis, V., Melchiori, M., A methodology and tool environment for process analysis and reengineering, *Data & Knowledge Engineering*, 1999, **31**, 253-278.

Castro, J., Kolp, M., Mylopoulos, J., Towards requirements-driven information systems engineering: the Tropos project. *Information Systems*, 2002, **27**, 65-389.

Chapron, J., Boucher, X., Lebrun, P., Managing the information system evolution: the microelectronics case study, in *15th Information Resources Management Association Conference*, 2004.

Organisational Urbanism of Information Systems

Chapron J., L'urbanisme organisationnel : méthode et aide à la décision pour piloter l'évolution du système d'information de l'entreprise. PhD Thesis, Ecole Nationale Supérieure des Mines de Saint Etienne, September 2006.

Chelli, H., *Urbaniser l'entreprise et son système d'Information. Guide des entreprises agiles*, 2003 (Vuibert: Paris).

Chen, S., Lin, L., Decomposition of interdependent task group for concurrent engineering. *Computers & Industrial Engineering*, 2003, **44**, 35-459.

Club-Urba, *Pratique de l'Urbanisme des systèmes d'information en entreprise*, 2003 (Publibook: France).

Crowston, K., A coordination theory approach to organisational process design, *Organisational Science*, 1997, **8**, 157-175.

Dahlgren, J., Söderlund, J., Managing inter-firm industrial projects on pacing and matching hierarchies. *International Business Review*, 2003, **10**, 305-322.

Dursun, D., Perakath, C.B., Towards a truly integrated enterprise modelling and analysis environment, *Computers in Industry*, 2003, **51**, 257-268.

GERAM, Generalised Enterprise Reference Architecture and Methodology. *The ISO 15704 Requirements for Enterprise Reference Architecture and Methodologies*, 2000.

Fujii S., Kaihara T., Sashio K. and Seki T., Agile manufacturing system with autonomously operated square arrayed machining cells, Preprints of INCOM'2006, *12th IFAC Symposium Information Control Problems in Manufacturing*, May 17-19 2006, Saint Etienne, France, pp. 517-522.

Harrison R., Ming Lee S., Huey Ong M. and West A.A., Distributed engineering of modular reconfigurable automation systems, Preprints of INCOM'2006, *12th IFAC Symposium Information Control Problems in Manufacturing*, May 17-19 2006, Saint Etienne, France, pp. 553-558.

Organisational Urbanism of Information Systems

Hee-Woong, K., Business process versus coordination process in organizational change. *The International Journal of Flexible Manufacturing Systems*, 2000, **12**, 275-290.

Hoffmann W. H., Schlosser R, Success Factors of Strategic Alliances in Small and Medium-sized Enterprises—An Empirical Survey, *Long Range Planning*, 2001, **34**, N° 3, 357-381.

Hwang, S.-H., Yang W.-S., On the discovery of process models from their instances. *Decision Support Systems*, 2002, **34**, 41-57.

ISO 19440, PR EN ISO 19440, *Enterprise Integration - Constructs for Enterprise Modelling*, september 2005 (ISO/DIS 19440:2005).

Jain, A.K., Murty, M.N., Flynn P.J, Data Clustering: A Review. *ACM Computing Surveys*, Vol. 31, N°3, 1999, pp.264-323.

Kuziak, A., Wang, J., Decomposition of the Design Process. *Journal of Mechanical Design*, 1993, **115**, pp.687-695.

Lebart L., Morineau A., Piron M., *Statistique exploratoire multidimensionnelle*, 1997 (Dunod: Paris).

Lee, H.-S., Automatic clustering of business processes in business systems planning. *European Journal of Operational Research*, 1998, **114**, 354-362.

Lim, S.S., Park K.H., ECEM : an event correlation based event manager for I/O-intensive application. *The journal of Systems and Software*, n° 74, 2005, 229-242,.

Le Roux B., Desbertrand L., Guérif P., Tang X., Tixier J., Verger P., *Urbanisation et Modernisation du SI»* , 2004 (Hermès: Paris).

Longépé, C., *Le projet d'urbanisation du système d'information*, 2001 (Dunod: Paris).

Malhéné, N., Gestion des processus d'évolution des systèmes industriels : conduite et méthode. PhD thesis, University Bordeaux 1, 2000.

Organisational Urbanism of Information Systems

Malone, T.W., Crowston, K., The interdisciplinary Study of Coordination. *ACM Computing Surveys*, 1994, **26**, 87-120.

Manufuture, Report of the European DG Research, *European Conference Manufuture'2003, European Manufacturing of the Future*, Milan, Italy, Dec. 2003.

Ould, M.A., *Business processes: modelling and analysis for reengineering and improvement*, 1995 (John Wiley & Sons).

Papadacci E., Salinesi C., NENO: une approche d'aide à l'arbitrage par l'évaluation qualitative et quantitative de la valeur métier du SI durant la phase amont du processus d'urbanisation, *2nd Workshop ECI "Ingénierie et Gestion des Processus d'Entreprise"*, Paris, 2005.

Pollalis, Y.A., Patterns of co-alignment in information-intensive organisations: business performance through integration strategies. *International Journal of Information Management*, 2003, **23**, 469-492.

Strader T.J., Lin F.R., and Shaw M.J., Information infrastructure for electronic virtual organization management, *Decision Support Systems*, **23**, pp. 75-94, 1998.

Vernadat, F., *Techniques de Modélisation en Entreprise: applications aux processus opérationnels*, 1999 (Economica: Paris).

Ward, J.H., Hierarchical grouping to optimize an objectimize function. *Journal of the American Statistical Association*, 1963, **58**, 238-244.

Williams, W. T., Lance, G.N., A general theory of classificatory sorting strategies - Hierarchical system. *The Computer Journal*, 1967, **9**, 373-380.

Williams, T. J., Rathwell, G.A., and Li, H, A handbook on master planning and implementation for enterprise integration programs, 1998 (Purdue University West Lafayette: Indiana 47907).

Yu, E., Mylopoulos, J., Modelling Organisational Issues for Enterprise Integration, in *2nd International Conference on Enterprise Integration and Modelling Technology*, 1997.

	Elementary Dependency Link	Basic description	Abbreviation - values
Actor dependency (D_A)	Actor sharing	The actors A_i and A_j of the processes P_n et P_m may or may not be the same	$d_a(A_i, A_j) = 0$ or 1
	Structural dependency	Computation of the structural dependency of the actors A_i and A_j based on the enterprise organisational chart	$d_s(A_i, A_j) = 0$ or 1
	Role dependency	Two roles performed by two distinct actors A_i and A_j may or may not be the same.	$d_r(A_i, A_j) = 0$ or 1
Resource dependency (D_{RE})	Software sharing	The software S_i, S_j used in the processes P_n et P_m may or may not be the same.	$d_{so}(S_i, S_j) = 0$ or 1
	Data flow dependency	Processes dependency based on the software in use and referring to the technical-oriented clusters generated by information system urbanism.	$d_f(S_i, S_j) = 0$ or 1
Information dependency (D_I)	Business object dependency	The business objects O_i and O_j used in the processes P_n et P_m may or may not, be the same.	$d_o(O_i, O_j) = 0$ or 1
	Event dependency	Coefficient expressing the use of the business objects as a temporal or logical coordination between the two processes.	$d_{ev}(O_i, O_j) = 1$ or 2

Table 1. Synthesis of the dependency measures

Organisational Urbanism of Information Systems

	Process Name	Process description
P ₁	Execution planning	Preparation of the manufacturing plan defining the manufacturing priorities.
P ₂	Process Change Review Board	Procedure of validation applied to process, equipment or quality change (every change concerning manufacturing).
P ₃	Quarter Capacity Planning	Quarterly planning process to define the manufacturing capacity allocated to each plant, for each product and each type of technology for a period of two years.
P ₄	Capital Approval Request	Process dedicated to the management of the investment request concerning the front-end chip manufacturing.
P ₅	Capacity Requirement Planning	Monthly planning process to define the manufacturing capacity allocated to each plant, for each product and each technology for a period of six months.
P ₆	Customer Demand	Customer order planning process with a delivery date.
P ₇	Scrap Management	Procedure for the management of manufacturing defect on wafers, covering the analysis of defects and proposal of remedial actions.
P ₈	Management of non-conform manufacturing lots	Procedure for the analysis and remedial actions on non-conform manufacturing lots
P ₉	Planning to maintain resources	Planning of required operations to maintain the manufacturing resources.
P ₁₀	Operations to maintain resources	Proceed to the operations necessary to maintain the manufacturing resources
P ₁₁	Employees training	Vocational training for the employees, notably on new manufacturing equipments.
P ₁₂	Teacher training	Training for the employees who will take in charge to teach to other employees.
P ₁₃	Recipe management	Equipment recipe procedure for the modification of equipment recipe. The recipe is the procedure to follow inside the equipment.
P ₁₄	Equipment transfer	Procedure to be followed when transferring manufacturing resources from on plant to another one.
P ₁₅	RAT process	Management of the lithography material utilised for wafer manufacturing.

Table 2. List of processes

Organisational Urbanism of Information Systems

	P2-Process Change Review Board							
	Actor 1	Actor 2	Actor 3	Actor 4	Actor 5	Actor 6	Actor 7	Actor 8
P7-Scrap Management								
Actor 1 (Manufacturing tech.)	0	0	0	0	0	0	0	0
Actor 2 (Control Engin.)	0	0	0	0	0	0	0	0
Actor 3 (Manufacturing tech.)	0	0	0	0	0	0	0	0
Actor 4 (Development Engin.)	1	1	1	0	1	1	1	1
Actor 5 (Quality expert)	1	1	1	0	1	1	1	1
Actor 6 (Manufacturing Engin.)	1	1	1	0	1	1	1	1

Table 3. Matrix of the structural dependencies between actors

For Peer Review Only

Organisational Urbanism of Information Systems

	Device	Division	Equipment Engineering	Manuf.	Process Control	Process Engineer	Production Control	Quality Assurance
Device	1	0	0	0	0	0	0	0
EWS Process Tech	0	0	0	0	0	0	0	0
Plant Scrap Champion	0	0	0	0	0	0	0	0
Plant Technician	0	0	0	0	0	0	0	0
Process Engineer	0	0	0	0	0	1	0	0
Production Control	0	0	0	0	0	0	1	0

Table 4. Role dependency matrix

Organisational Urbanism of Information Systems

	P ₅	P ₄	P ₆	P ₁₄	P ₁	P ₈	P ₁₁	P ₉	P ₁₀	P ₂	P ₃	P ₇	P ₁₃	P ₁₅	P ₁₂
P ₅	1	0,07	0,28	0,00	0,09	0,00	0,00	0,00	0,00	0,00	0,44	0,00	0,00	0,00	0,00
P ₄	0,07	1	0,03	0,00	0,04	0,02	0,01	0,01	0,01	0,06	0,11	0,05	0,01	0,00	0,00
P ₆	0,28	0,03	1	0,00	0,26	0,03	0,00	0,00	0,02	0,01	0,14	0,00	0,01	0,00	0,00
P ₁₄	0,00	0,00	0,00	1	0,00	0,00	0,01	0,18	0,00	0,08	0,00	0,00	0,00	0,00	0,10
P ₁	0,09	0,04	0,26	0,00	1	0,00	0,00	0,00	0,00	0,00	0,26	0,02	0,00	0,00	0,00
P ₈	0,00	0,02	0,03	0,00	0,00	1	0,16	0,03	0,15	0,30	0,00	0,17	0,31	0,02	0,10
P ₁₁	0,00	0,01	0,00	0,01	0,00	0,16	1	0,03	0,00	0,05	0,00	0,15	0,23	0,01	0,27
P ₉	0,00	0,01	0,00	0,18	0,00	0,03	0,03	1	0,21	0,24	0,00	0,04	0,19	0,00	0,14
P ₁₀	0,00	0,01	0,02	0,00	0,00	0,15	0,00	0,21	1	0,08	0,00	0,02	0,18	0,00	0,00
P ₂	0,00	0,06	0,01	0,08	0,00	0,30	0,05	0,24	0,08	1	0,00	0,35	0,10	0,00	0,14
P ₃	0,44	0,11	0,14	0,00	0,26	0,00	0,00	0,00	0,00	0,00	1	0,00	0,00	0,00	0,00
P ₇	0,00	0,05	0,00	0,00	0,02	0,17	0,15	0,04	0,02	0,35	0,05	1	0,02	0,01	0,02
P ₁₃	0,00	0,01	0,01	0,00	0,00	0,31	0,23	0,19	0,18	0,10	0,00	0,02	1	0,02	0,13
P ₁₅	0,00	0,00	0,00	0,00	0,00	0,02	0,01	0,00	0,00	0,00	0,00	0,01	0,02	1	0,12
P ₁₂	0,00	0,00	0,00	0,10	0,00	0,10	0,27	0,14	0,00	0,14	0,00	0,02	0,13	0,12	1

Table 5. Actor dependency matrix for each pair of processes.

Organisational Urbanism of Information Systems

Figure 1. Aggregation procedure for the actor dependency

Figure 2. Clustering approach

Figure 3. Agglomerative hierarchical clustering method

Figure 4. Actor and resource dependency dendrograms

For Peer Review Only

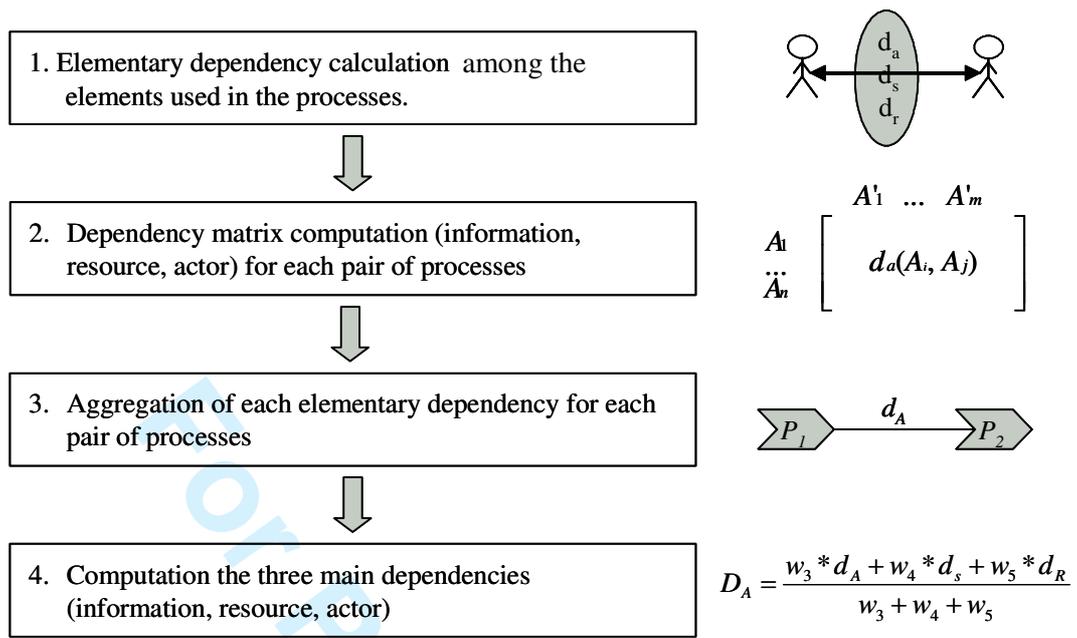


Figure 1. Aggregation procedury for the actor dependency

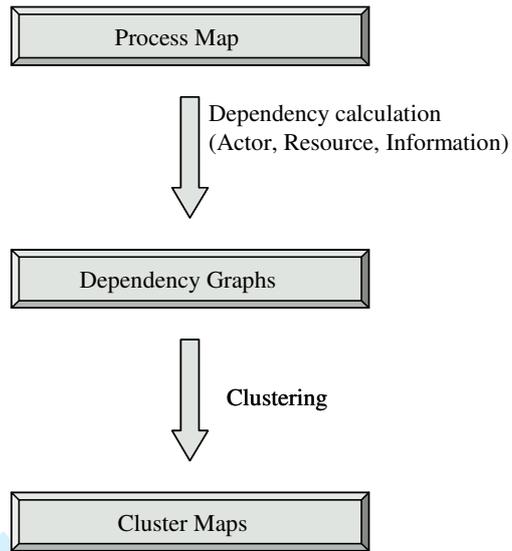


Figure 2. Clustering procedure

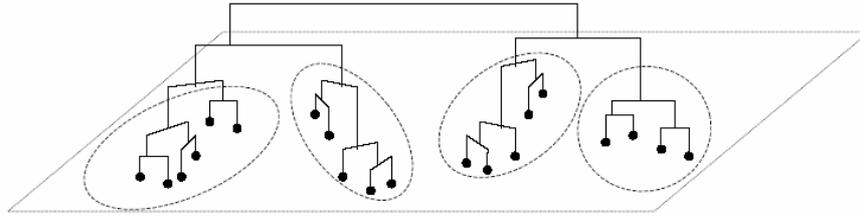


Figure 3. Agglomerative hierarchical clustering method

For Peer Review Only

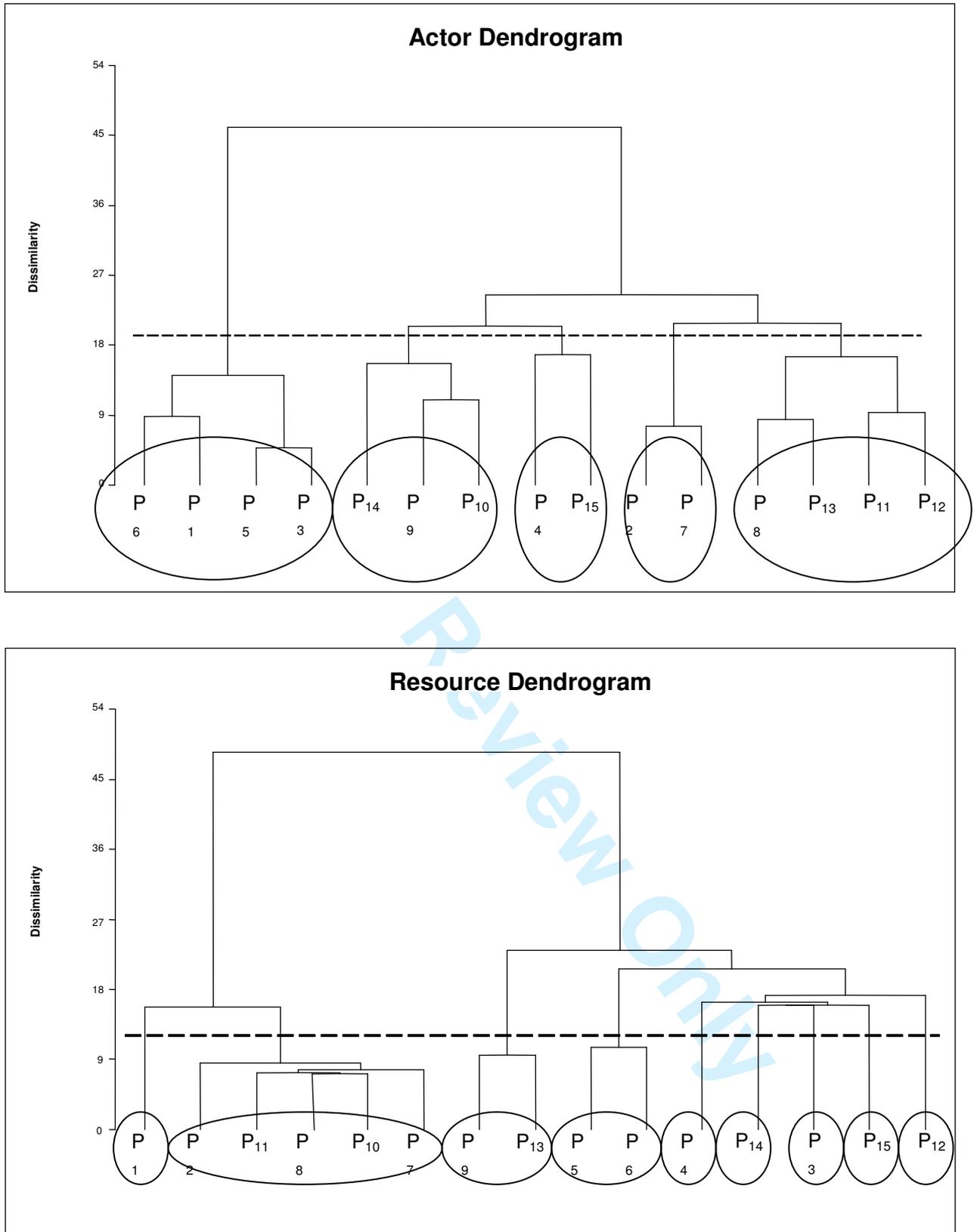


Figure 4. Actor and resource dependency dendrograms