Evaluating the Economic Implications of Information Systems: A Formal Model of Information Processing Capacity

CHIARA FRANCALANCI, PAOLO GIACOMAZZI
Department of Electronics, Information and Bioengineering
Politecnico di Milano
Piazza Leonardo da Vinci, 32
20133 Milano, Italy
chiara.francalanci@polimi.it, paolo.giacomazzi@polimi.it

Abstract: - The choice of the pattern for cooperation is a traditional issue of the organizational debate on business process design, which typically compares alternative cooperation patters on the basis of cost variables. From an engineering perspective, costs should be contrasted against production capacity as a primary indicator of performance. This paper takes this perspective and proposes a formal model of information processing capacity to evaluate alternative cooperation patterns. Simulation results show how a capacity perspective delivers significantly different results from the cost-oriented organizational view. Traditional design principles for the selection of the most efficient organizational structure (i.e. degree of process networking, specialization, delegation and information overflow) are significantly complemented.

Key-Words: - Information Theory, Value of Information, Information processing capacity.

1 Introduction

Information systems (ISs) are designed to address the fragmentation of information across an organization by means of data and application integration [1]. An information system is often the result of the composition of multiple information processes, which cooperate towards a common set of objectives. Building a coherent logic of an information system's cooperative effort, referred to as cooperative process, is the main goal of IS design [2].

From an organizational perspective, there exist alternative patterns for cooperation and the choice of the pattern for cooperation is a primary issue of the debate on organizational performance, which is central to economic theory and, hence, to a multitude of past research contributions [3], [4], [5]. By embedding the business logic (or the logic of organizational processes), enterprise integration technologies necessarily implement a specific pattern for cooperation.

In the system engineering literature, this design step is tackled during requirements analysis and, more specifically, during early requirements analysis when cooperative processes are analyzed, modeled and reengineered [6], [7]. Research on early requirements analysis provides methodologies and tools to elicit requirements by analyzing organizational processes, to then represent relevant characteristics of processes and to verify the completeness, correctness and understandability of process models [7], [8], [9]. However, previous contributions from the economic literature on the choice of cooperation patterns are only marginally considered.

The goal of this paper is to categorize the cooperation patterns enabled by enterprise integration technologies, by taking an engineering view to evaluate their performance. In the organizational literature, the evaluation of cooperation patterns is typically based on economic theory which compares cooperation alternatives on the basis of cost variables [10], [11], [12]. Costs describe the efficiency of a process at using its input resources, but need to be complemented by an evaluation of the output that can be obtained with different cooperation patterns in order to assess performance [13]. From an engineering perspective, production capacity is usually considered a primary indicator of performance, as a measure of the output to input ratio of production processes [13]. The general effect of technology is a reduction of costs and a growth of capacity and an enterprise integration framework should have a similar effect, leading to an increase of the ratio of capacity to costs.

This paper takes this perspective and proposes a formal model of capacity to evaluate alternative cooperation patterns. The paper's definition of capacity is original in that it incorporates the characteristics that differentiate information from other organizational resources and, therefore, computer-supported cooperative processes from traditional production processes. Results from a capacity-based analysis of cooperation patterns are then compared with previous cost-based findings in the organizational literature.

The presentation starts from a review of the economic and organizational literature on cooperative processes. The model of information processing capacity is then presented in Section 3. Section 4 summarizes the testing approach and illustrates the main findings. A final discussion derives preliminary capacity-oriented EI design guidelines and IS managerial implications.

2 Describing the Information Processing Capacity of Cooperative Processes

Our first goal in reviewing the literature is to classify alternative cooperation patterns and discuss the traditional organizational principles for the design of cooperative processes (Sections 2.1-2.5). In the organizational literature, the very activity of organizing is defined as the design of the rules for cooperation, which, from a software standpoint, constitute the design rules of the cooperative processes embedded in EI technologies [14].

From an engineering perspective, although cooperation is a costly activity, it also delivers benefits. A potential source of benefits originates from the specific characteristics of information as an organizational resource. Different from other resources, such as raw materials or product components, information is non-depletable [15], [16], i.e. it is not destroyed by use. As a consequence, information can be shared among a virtually infinite number of tasks without any risk of scarcity. Information is also self-generating [16]: By accessing and transforming information, individuals produce additional knowledge that can be made increasingly customized to the needs of process tasks. This customization process improves the utilization of the information resource by making it more easily accessible to individual executors and has the potential to increase their overall information processing capacity. In this respect, information sharing is not only an inevitable and costly consequence of the division of labor, but also an opportunity to create customized information and increase capacity [17]. In the following, the main organizational alternatives in the design of cooperation patterns are reviewed and the corresponding opportunities for the creation of customized information are discussed.

2.1 Task Parallelism

Organizational studies observe that a sequential execution of tasks removes information processing resources from execution to be conveyed to information sharing activities. Therefore, networking among tasks is regarded as a costly solution that should be implemented only when necessary as a consequence of the need for cooperation [18].

From an engineering perspective, this process design principle is contrasted by the opportunities for information customization created by a sequential coordination among tasks. Customization delivers benefits by simplifying the execution of tasks that share increasingly useful information. Tasks can use information created by other tasks only if they are executed sequentially or, at least, they decrease their degree of parallelism by synchronizing when information sharing is required. For example, product reengineering should be based on feedback information from the marketing function documenting customers' perceptions. In turn, a lower degree of parallelism and, conversely, a higher degree of networking among the marketing and production functions can enhance information processing capacity and provide benefits. Consequently, a first research question is whether a higher degree of networking among tasks can enhance capacity, although increasing costs.

2.2 Task Specialization

A second traditional question is what degree of specialization maximizes organizational performance. In answering this question, the literature starts from a traditional process design principle that associates the introduction of non-computer-based technologies with a higher degree of task specialization [19].

According to the information-processing perspective of organizational theory, the opposite holds for computer-based technologies. Computers automate part of the decision-making work and can extend the boundaries of individual rationality [20], [21], [18]. Computer-supported individuals can be assigned more complex tasks and play a less specialized role.

From an engineering perspective, greater specialization can also be associated with benefits. The opportunities for taking advantage of self-generation and non-depletability through information customization are in fact augmented by task specialization. Splitting tasks into more specialized sub-tasks may enable the creation of new intermediate information that can be customized to the needs of other tasks. Enterprise Application Integration (EAI) technologies may further enhance the benefits from task specialization by supporting communication and, thus, reducing the overhead costs of sharing customized information.

2.3 The Delegation of Decision-Making

The debate on task specialization has prompted a parallel debate on the delegation of decision-making. Within an organization, decision-making responsibilities are organized hierarchically, ranging from top management at the highest level to operations management at the lowest level. Delegation is defined as the downwards shift of decision-making responsibilities in the organizational hierarchy [4]. This reallocation of responsibilities modifies cooperation patterns, since it changes the organizational units involved in the execution of decision-making tasks.

The traditional process design principle suggests to minimize the degree of delegation by centralizing decisions within high-level organizational units [19], [22]. According to the information-processing perspective of the organizational literature, delegation brings decisions closer to decision-making information and may reduce the costs of sharing information [11]. However, delegation also involves additional costs. Decision makers can behave opportunistically and take advantage of greater responsibility to their own benefit [11].

With a cost-oriented view, the potential benefits of delegation are overlooked, as its influence on the customization of information is not considered. Within a rigid hierarchical system, when exceptions occur and decision-making is required, all the information must be conveyed upwards and peer communication takes place only indirectly through higher hierarchical levels. Delegation moves decision-making responsibilities towards lower hierarchical levels and, consequently, favors direct peer communication in exceptional circumstances [21]. If it is conveyed upwards without this direct interaction, information is usually synthesized to simplify higher-level managerial work, but not accurately customized [23]. As a consequence, delegation may have a positive effect on information processing capacity, once again related to greater opportunities for customization.

2.4 The Size of Cooperative Processes

The reallocation of decision-making within the organizational hierarchy is not the only form of delegation. Organizations can also outsource part of their production and related decision-making activities to other organizations, such as customers, suppliers, consultants or commercial partners. Organizational studies indicate that there is continuum of cooperation forms between hierarchies and markets, basically depending on the stability of the relationship among cooperating actors. The position along this continuum should selected on the basis of a cost trade-off [10], [12]. It has also been theorized that computer applications shift this trade-off towards greater market coordination and, as a consequence, can cause a reduction of the average size of organizations in the economic system [10], [12].

As noted before, from an engineering perspective, although involving costs, coordination is beneficial to the creation of cumulative knowledge by means of increasingly customized information. This process of information customization provides different outcomes when executed through hierarchical or market forms of cooperation. Customization is an inherently iterative process that requires continuous adjustments to build cumulative knowledge. As a consequence, it may be more efficiently achieved if the relationship between interacting parties is more stable, that is closer to hierarchical forms of coordination.

2.5 The Risk of Information Overflow

Although delivering benefits, the non-depletability and self-generation features of information also involve a risk, known as information overflow. Not only information is not destroyed by use, but it can constantly grow in volume through processing. Over time, the risk for organizations is to be challenged by an increasing information-seeking effort to filter out useful knowledge from a relentlessly expanding information base [17]. Information overflow occurs when organizations are unable to transform their inputs into useful knowledge and cannot implement a learning process that eliminates data of little use and retain a manageable quantity of valuable information.

It can be hypothesized that different cooperation patterns are affected by information overflow to varying degrees, as they are responsible for the breadth and frequency of information exchanges among individuals. The effect of customization on the risk of information overflow at a process level has not been explicitly addressed in the organizational literature and the impact of different cooperation patterns on customization and, hence, on information overflow at a process level is still an open issue.

3 A Formal Model of Information Processing Capacity

3.1 Information processing tasks

An information processing task T_i is characterized by the input information flow, $I_{input,i}$, and the output information flow, $I_{i,output}$. The main characteristic of information processing tasks is that input information is non depletable and can be reused multiple times to produce output information. Therefore, $I_{i,output}$ can be greater than, equal to, or smaller than $I_{input,i}$. Whenever $I_{i,output}$ is greater than $I_{input,i}$, tasks generate more information than they receive, by taking advantage of the self-generation feature of information. Each task is associated with an *information customization factor*, λ_i, that represents the average fraction of input information that is processed by the node to produce each unit of output information. By definition, λ_i ranges between 0 and 1 ($\lambda_i \in [0,1]$), since at most the whole amount of input information is used to produce each unit of output. In general, a low λ_i indicates that part of the input information is not used or is reused a few times by the processing task and is consequently less customized.

In our model, we assume that a task produces the output information in one time unit. As an example, Figure 1 shows a task receiving 20 units of input information per time unit and producing 15 units of output information per time unit. Since λ_i is equal to 0.1, a fraction equal to 10% of the input information is used to produce each unit of output information. Thus, in order to produce each unit of output information, $I_{input,i}\lambda_i$ = 2 units of input information are required. To produce $I_{i,output}$ units of output information, the node processes $(I_{input,i}\lambda_i)$ $I_{i,output}$=30 units of information per time unit. In general, a task processes a total amount of information equal to $I_{input,i}\lambda_i I_{i,output}$ per time unit. This quantity is referred to as the Information Processing Capacity (IPC_i) of the task and is defined as:

(1) $IPC_i = I_{input,i}\lambda_i I_{i,output}$.

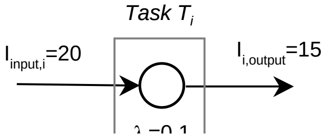

Fig 1. Example of information processing task.

Input (output) information is the quantity of information entering (exiting) the task in a time unit. Let us assign a conventional unit to information, [I/T], and a conventional unit to time, [T]. With reference to Figure 1, task T_i receives 20 [I/T] (that is, 20 units of information in a time unit) and produces 15 [I/T] (that is, 15 units of information in a time unit). Since $IPC_i = I_{input,i}\lambda_i I_{i,output}$ is the quantity of information processed by the task in a time unit, IPC_i is measured as [I/T]. Therefore, by performing a simple dimensional analysis on Equation 1, it is obtained:

(2) $[IPC_i] \equiv [I_{input,i}][\lambda_i][I_{i,output}] \Rightarrow [I/T] \equiv [I/T][\lambda_i][I/T] \Rightarrow [\lambda_i] \equiv [T/I]$.

3.2 Cooperation among information processing tasks

Cooperation among tasks is modeled as a directed graph, where each node T_i represents a task in the process and oriented edges connecting pairs of nodes (T_i,T_j) represent information exchanges between corresponding tasks. A cooperative process is defined as a set of cooperating tasks.

In the graph describing a process, the weight of the directed edge connecting node T_i to node T_j represents the quantity of information I_{ij} passing from T_i to T_j in a time unit (if I_{ij} = 0, node T_i is not connected to node T_j). A node T_i can exchange information with multiple nodes. T_i can also exchange information with the external environment, that is, with tasks that belong to other processes. Information exchanges of a node T_i to and from the external environment are indicated as $I_{i,ext}$ and $I_{ext,i}$, respectively. Overall, the total input information of node T_i is $I_{input,i} = \sum_{j=1}^{j=n} I_{ji} + I_{ext,i}$ and the overall output information is $I_{i,output} = \sum_{j=1}^{j=n} I_{ij} + I_{i,ext}$.

By applying Equation 1 to a generic node T_i in a graph, the information processing capacity, IPC_i, of that node can be defined as the overall quantity of information processed in a time unit, as in (1). IPC_i

is measured as information per time unit, that is, as [I/T]. Similar to individual tasks, the information processing capacity of a cooperative process, indicated as IPC, is defined as the total quantity of information processed by the corresponding set of cooperating tasks in a time unit.
The total amount of information processed by a set of cooperating tasks may not coincide with the total input from the external environment. Input information from the external environment may be used multiple times by tasks and, therefore, the total amount of information processed may be greater than the total input information of the cooperative process, defined as $I_{input} = \sum_{i=1}^{i=n} I_{ext,i}$.

Furthermore, in a cooperative process with multiple tasks, the total amount of information processed may not coincide with the summation of the total information processed by each node, referred to as Summation of Information Processing Capacities (SIPC) and formally defined as:

(3) $$\text{SIPC} = \sum_{i=1}^{i=n} I_{input,i} \lambda_i I_{i,output} .$$

This is shown by means of the following example. Let us consider the sequence of two tasks reported in Fig. 2. The external information $I_{ext,1}$ is used $\lambda_1 I_{12}$ times in order to produce I_{12}, which, in its turn, is used $\lambda_2 I_{2,ext}$ times in order to produce $I_{2,ext}$. As a consequence, $I_{ext,1}$ is indirectly used $\lambda_1 I_{12} \lambda_2 I_{2,ext}$ times in order to produce $I_{2,ext}$ and, thus, the total information processed by the sequence of tasks is $I_{ext,1}(\lambda_1 I_{12} \lambda_2 I_{2,ext})$.

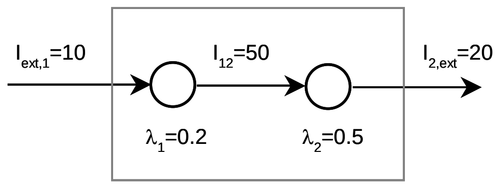

Fig. 2. The information processing capacity of a sequence of tasks.

The advantage from dividing the process into two subsequent tasks is that task processors, either humans or computers, will be able to accomplish simpler individual tasks (IPC_1=100, IPC_2=500), while a one-task process would require the direct processing of $I_{ext,1}(\lambda_1 I_{12} \lambda_2 I_{2,ext})$ =1000 units of information per time unit.

Let us define X_i as the total amount of information from the external environment used by node T_i. Since node T_i can process external information either directly, $I_{ext,i}$, or indirectly through other nodes communicating with T_i in the process, X_i is formally defined as follows:

(4) $$X_i = I_{ext,i} + \sum_{j=1, j \neq i}^{j=n} X_j \lambda_j I_{ji} ,$$

where n is the total number of nodes in the cooperative process. The term

(5) $$\sum_{j=1, j \neq i}^{j=n} X_j \lambda_j I_{ji}$$

represents the information indirectly entering from the external environment through flows from other nodes T_j sending information I_{ji} to T_i.

In a general case, X_i can be calculated from a set of n linear equations of which the X_i constitute the n unknown terms (Equation 4). The total quantity of information processed in a time unit by the set of cooperating tasks can be calculated with the following summation:

(6) $$IPC = \sum_{i=1}^{i=n} X_i \lambda_i I_{i,ext} .$$

Note that tasks with a null $I_{i,ext}$ give no contribution to the summation. However, they contribute to the production of process output $\sum_{i=1}^{i=n} I_{i,ext}$ indirectly and this indirect contribution is already accounted for by the terms X_i. In the example reported in Fig, $X_2 \lambda_2 I_{2,ext}$=1000 would be the only term in the summation and would represent the total information from the external environment processed by the sequence of tasks, consistent with former intuitive analyses.

The *information customization factor* λ (measured as [T/I]) of a cooperative process is defined as the average fraction of input processed per unit of output by the entire process. Therefore, the customization λ of a process can be expressed as:

(7) $$\lambda = \frac{\sum_{i=1}^{i=n} X_i \lambda_i I_{i,ext}}{\sum_{i=1}^{i=n} I_{ext,i} \times \sum_{i=1}^{i=n} I_{i,ext}}$$

Since nodes are associated with an execution time, processes have a dynamic behavior. In general, a node's X_i at time k depends on other nodes' X_j at preceding time steps. This can be modeled as a discrete dynamic system, by assuming that the information processed by a node at step k depends

on the information processed by other nodes during the previous step, *k-1*.
In order to discuss the dynamic behavior of a cooperative process, it is important to recall that input and output information can be a function of time and, therefore, they are formally represented by $I_{ext,i}(k)$ and $I_{i,ext}(k)$, respectively, where k is a natural number representing the discrete time index. For example, if a node T_i receives at each time unit the same amount of external input information, equal to 2 units, starting from the beginning of the process (that is, from time $k = 0$), then $I_{ext,i}(k) = 2\ u_1(k)$, where $u_1(k)$ is the unit step function in the discrete-time domain. According to the definitions discussed in Section 3.2.1, the total information processed by task T_i at step $k + 1$ would evaluate to:

(8) $$X_i(k+1) = I_{ext,i}(k+1) + \sum_{j=1}^{j=n} X_j(k)\lambda_j I_{ji}\,.$$

Accordingly, the information processing capacity of the process at step k would be:

(9) $$IPC(k) = \sum_{i=1}^{i=n} X_i(k)\lambda_i I_{i,ext}(k)\,.$$

The terms $X_i(k)$ describe the status of the process as a dynamic system, while IPC(k) represents the output in terms of information processing capacity, as a function of time. $X_i(k)$ and IPC(k), describing the dynamic behavior of a process, can also be expressed in the classical matrix form. By defining |X(k)| as the *n*x1 column vector whose element in row *i* is $X_i(k)$, ||A|| as the *n*x*n* square matrix whose element (*i*, *j*) is $\lambda_i I_{ij}$, and |B(k)| as the *n*x1 column vector whose element in row *i* is $I_{ext,i}(k)$, |X(k)| can be written as:

(10) $|X(k)| = ||A||^T\ |X(k-1)| + |B(k)|$,

Where T is understood as the transposition operator. Moreover, by defining |H| as the 1x*n* row vector whose element in column *j* is $\lambda_j I_{j,ext}$, IPC(k) can be calculated as:

(11) IPC(k) = |H| |X(k)|.

3.3 The ratio of IPC to SIPC as a Measure of Process Performance

IPC represents the total quantity of information that a set of cooperating tasks can process in a time unit. The summation of their individual capacities, $SIPC = \sum_{i=1}^{i=n} IPC_i$, represents the actual processing work performed by tasks. In summary, $\sum_{i=1}^{i=n} IPC_i$ represents the effort required to obtain IPC. The ratio of IPC to $\sum_{i=1}^{i=n} IPC_i$ represents a measure of the efficiency of a process at employing its processing resources to provide an overall capacity.

If tasks are independent, that is they execute their processing activity in parallel, the outcome of their processing coincides with their effort, that is IPC can be demonstrated to coincide with $\sum_{i=1}^{i=n} IPC_i$. By definition, $IPC = \sum_{i=1}^{i=n} X_i \lambda_i I_{i,ext}$, where, in parallel processes such as the one reported in Fig, the terms X_i evaluate to $I_{ext,i}$ and, as a consequence, IPC can be expressed as $IPC = \sum_{i=1}^{i=n} I_{ext,i}\lambda_i I_{i,ext}$, which represents the summation of the information processing capacities of individual nodes, SIPC.

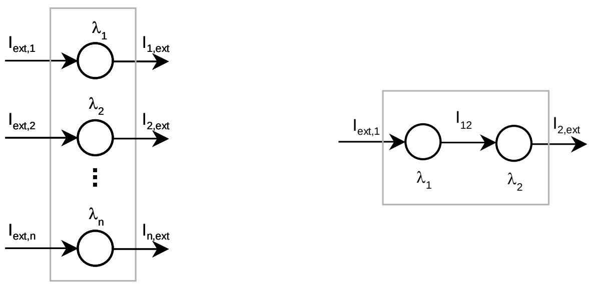

Fig.3. Sample parallel process and generic two-task sequential process.

If tasks cooperate by exchanging information, the total quantity of information that they collectively process in a time unit can be greater than the summation of the quantities actually processed by each one of them, that is IPC can be greater than $\sum_{i=1}^{i=n} IPC_i$. As an example, let's consider a two-task sequential process (Figure 3). The condition for process IPC to be greater than the summation of individual capacities is:

$$I_{ext,1}\lambda_1 I_{12}\lambda_2 I_{2,ext} > I_{ext,1}\lambda_1 I_{12} + I_{12}\lambda_2 I_{2,ext}\,.$$

By eliminating I_{12} on both sides, the former condition becomes:

$$\lambda_1 > \frac{\lambda_2 I_{2,ext}}{I_{ext,1}(\lambda_2 I_{2,ext} - 1)}.$$

In Section 2, it has been hypothesized that cooperation can deliver benefits by increasing the overall information processing capacity of processes. Parallel tasks do not cooperate with each other and do not share information; the sequential execution of tasks is necessary to exchange information. The model verifies that sequential tasks can achieve an overall IPC greater than parallel tasks, thus confirming the positive effect of coordination on capacity.

3.4 Instability and information overflow

In order to discuss the stability of the solution of the system defined in Equation 11, we switch to the Z transform domain. In the Z domain, all functions of discrete time *k* are represented as functions of the complex variable *z*, according to the transformation rule $F(z) = \sum_{k=0}^{k=\infty} f(k) z^{-k}$:

(12) $|X(z)| = (||I|| - z^{-1}||A||^{T})^{-1}|B(z)|$,

where $||I||$ is the identity matrix. Moreover, IPC(*z*) can be written as:

(13) $IPC(z) = |H|\,|X(z)|$.

The poles of the process are the eigenvalues of the characteristic matrix associated to $||A||^T$. Thus, the poles of the process are defined by the following equation:

(14) $\det(||I|| - z^{-1}||A||^{T}) = 0$.

The steady-state solutions can be obtained by solving the following system of linear equations:

(15) $(||I|| - ||A||^{T})\,|X| = |B|$.

Technically, the system is stable if its poles are included inside the unit circle in the complex plane. When the system is unstable, the X_i do not converge and, consequently, IPC does not reach a steady-state. The mathematical reason why the X_i terms diverge is that the influence of past inputs from the external environment does not decrease over time. As a consequence, the calculation of IPC at a given time step requires all previous inputs from the time when the set of cooperating tasks has started its processing activities and over time, past inputs have an increasing weight in the determination of IPC with respect to recent information.

Note that when processes can be modeled as DAGs (directed acyclic graphs) that is, have no feedback information flows, the corresponding dynamic system can be demonstrated to be always stable. Let us consider a generic process with a single input $I_{ext,i}$ to one of its nodes. $X_i(k)$ is a function of $I_{ext,i}$ according to the impulse response of the dynamic system h(k), that is:

$$X_i(k) = \sum_{j=0}^{j=\infty} h(k-j) \times I_{ext,i}(j).$$

If the process is a DAG, h(k) is both finite and bounded. Therefore, $X_i(k)$ is also bounded, since it satisfies the following condition:

$$|X_i(k)| = \left| \sum_{j=0}^{j=\infty} h(k-j) \times I_{ext,i}(j) \right| \leq \sum_{j=0}^{j=\infty} |h(k-j)| \times |I_{ext,i}(j)|.$$

If $X_i(k)$ is bounded, the system is stable. These considerations can be easily extended to the general case of multiple $I_{ext,i}$, thus demonstrating that DAGs are stable. Feedback flows are therefore responsible for the possible creation of instability conditions, by conveying too much information back into the processing cycle. This is consistent with the conceptual definition of information overflow as the uncontrolled growth of an organization's information base due to excessive information sharing provided in Section 2.5.

4 Simulation Results

Four cooperation patterns, from fully centralized to fully decentralized with different degrees of delegation have been analyzed. Figure 4 reports these patterns in the case of 7-node processes. Pattern (1) represents a fully centralized pattern where one node is responsible for final decisions and information exchanges with the external environment. Lower-level nodes do not communicate directly with each other and can exchange information only with the central node. Pattern (2) represents a first step of delegation, as the central node allows direct information exchanges among lower-level nodes, while retaining a general supervisory role and control over information exchanges with the external environment. When communication with the external environment is also delegated, pattern (3) is obtained. Pattern (4) has a fully decentralized structure, with no central node and no supervision over operating nodes.

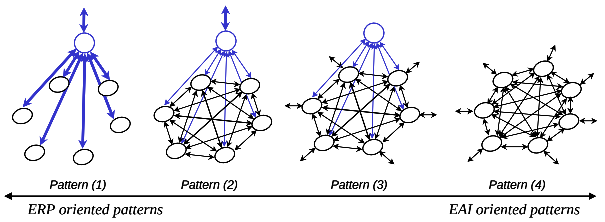

Fig. 4. Alternative cooperation patterns with 7-node size.

Cooperation patterns (1) to (4) correspond to actual alternatives in real organizations which have been extensively discussed in the organizational literature [24]. As a real instance of process alternatives (1) to (4), let us consider the case of the supply-chain management in a manufacturing company. The cooperative information framework will have to integrate and coordinate different applications and different organizational units or organizations.

For example, if supply-chain management activities are organized according to pattern (1), customers have a single reference point inside the organization, which is also in charge of coordination among different responsibilities. Alternatively, different organizational units can coordinate autonomously through direct information exchanges, but communicate to a central node the information necessary to manage customer relationships, thus implementing pattern (2). With pattern (3), customers would interact with different units depending on the status of their order. Pattern (4) would delegate supervision to operating organizational units which would mutually control their behavior. In this case, customers may indirectly influence this mutual control by providing organizational units with information on other units' behavior.

Since research questions address process characteristics and do not make a distinction among tasks, model parameters are assigned the same value for all nodes. Similarly, $I_{input,i}$ is always attributed the same value as $I_{i,output}$. Given a value of size, different processes are assigned identical $I_{input,i}$ and λ_i and, as a consequence, they have identical SIPC. This implies that edges will have different weights in different patterns, but they are all attached the same weight within the same pattern. The hierarchy of tasks is built by calculating the number of per-node branches in the hierarchy of nodes that maximizes the number of nodes at the operating (lowest) level. Simulations have been performed by means of an *ad hoc* tool developed in Java, as the use of general-purpose mathematical applications has proved time inefficient for large processes.

As an example, Figures 5 to 8, report simulation results with varying $I_{input,i}$ and λ_i. The other parameter is assigned a constant value. In Figures 5 to 8, n=22 in order for processes (1), (2) and (3) to

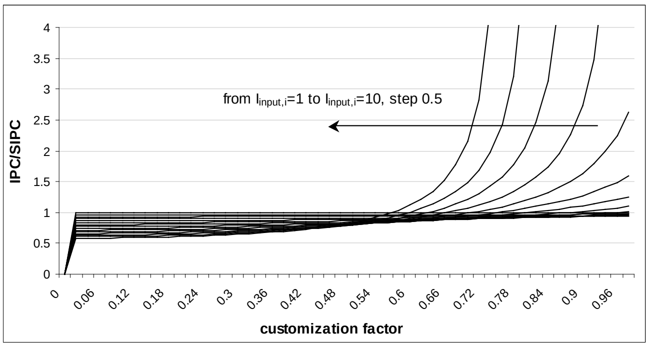

Figure 5 – Efficiency of pattern (1) as a function of λ_i and $I_{input,i}$.

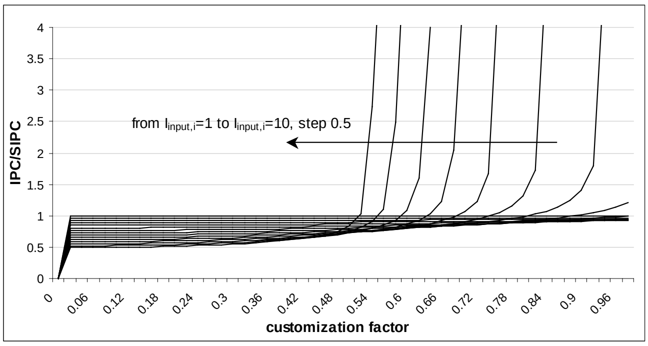

Figure 6 – Efficiency of pattern (2) as a function of λ_i and $I_{input,i}$.

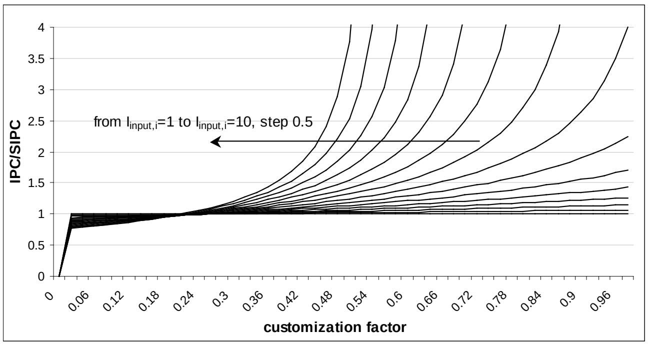

Figure 7 – Efficiency of pattern (3) as a function of λ_i and $I_{input,i}$.

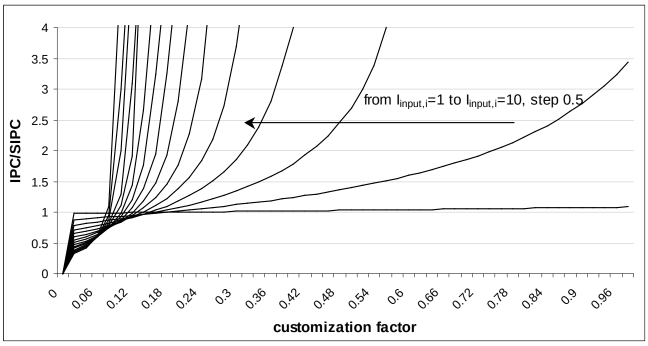

Figure 8 – Efficiency of pattern (4) as a function of λ_i and $I_{input,i}$.

have at least two hierarchical levels. These have been selected as representative values, although simulations have been performed with similar results for the entire range of values of λ_i from 0.02 to 1, with $I_{input,i}$ ranging from 0.1 to 100 and with size ranging from 3 to 1000 nodes.

5 Conclusions

The first question, inquiring whether greater parallelism among tasks can enhance capacity, has been already provided a mathematical answer in Section 3.3. It has been shown how in a two-task process sharing information through a sequential coordination between tasks, as opposed to their parallel execution, can increase information processing capacity above the summation of individual capacities. However, this occurs only if customization exceeds a minimum value. Simulations confirm this requirement on customization. Figures 5 to 8 report the ratio of IPC to SIPC as a function of customization for varying quantities of the information exchanged among nodes. For low values of customization, the ratio of IPC to SIPC is smaller than one, that is, the sequential coordination among tasks does not provide benefits; as the customization factor grows, the ratio of IPC to SIPC increases above one. However, high levels of customization can cause instability.

Different cooperation patterns show different levels of information processing capacity. Size tends to reduce the sensitivity of processes to customization. This would confirm the traditional process design principle recommending smaller size as a way to increase external delegation through outsourcing and improve performance [10], [12]. However, size reduces process efficiency less rapidly in more delegated patterns. Accompanying growth with a higher degree of hierarchical delegation can mitigate the negative effects of size and may even result into an overall increase of information processing capacity. This is consistent with the discussion in Section 2.4, where the downwards shift of decision making along hierarchical levels has been described as a form of delegation that could lead to more efficient and larger organizations. Delegation shows a general positive effect on capacity, but it also increases the risk of information overflow. This identifies a relationship between the choice of the cooperation pattern and the risk of information overflow.

The quantity of information exchanged among nodes decreases process efficiency when customization is low, while it increases efficiency as customization grows (Figures 5 to 8). This indicates that task specialization has a positive effect on capacity only when customization is low. Simulation findings indicate that if customization is already high, specialization is not needed and has in fact a negative impact on capacity. In these cases, specialization involves costs with no corresponding benefits and the traditional process design principle recommending a lower degree of specialization seems to be more appropriate. Overall, simulation results indicate that a capacity-oriented perspective can deliver process design principles different from the traditional cost-oriented approach.

References:

[1] J. Lee, K. Siau, S. Hong. 2003. Enterprise Integration with ERP and EAI. Communications of the ACM, 46, 2 (2003) 54-60.

[2] Y. Jiang, J. Zhao. 2014. Co-creating business value of information technology, Industrial Management and Data Systems, 114 (1), pp. 53-69, 2014.

[3] T.W. Malone, K. Crowston. 1994. The Interdisciplinary Study of Coordination. ACM Computing Surveys, 26(1), 87-119.

[4] W.R. Scott. 1992. Organizations: Rational, Natural, and Open Systems. Prentice Hall, 3rd ed.

[5] M.F. Balcan, A. Blum, Y. Mansour, The Price of Uncertainty, ACM Trans. on Economics and Computation, 1(3), pp. 15-29, 2013.

[6] G. De Michelis, E. Dubois, M. Jarke, F. Matthes, J. Mylopoulos, M. Papazoglou, K. Pohl, J. Schmidt, C. Woo, E. Yu. 1997. Cooperative Information Systems: A Manifesto, Cooperative Information Systems: trends & Directions, M. Papazoglou, G. Schlageter eds.

[7] A.G. Sutcliffe, N.A. M. Maiden, S. Minocha, D. Manuel. 1998. Supporting Scenario-Based Requirements Engineering. IEEE Trans. on Software Engineering, 24(12), 1072-1088.

[8] I. Sommerville, P. Sawyer, S. Viller. 1999. Managing Process Inconsistency Using Viewpoints. IEEE Trans. on Software Engineering, 25(6), 784-799.

[9] A. Lamsweerde, E. Letier. 2000. Handling Obstacles in Goal-Oriented Requirements Engineering. IEEE Trans. on Software Engineering, 26(10), 978-1005.

[10] T.W. Malone. 1987. Modeling Coordination in Organizations and Markets. Management Science, 33(10), 1317-1332.

[11] V. Gurbaxani and S. Whang. 1991. The Impact of Information Systems on Organizations

and Markets. Communications of the ACM, 34(1), 49-73.
[12] E. Brynjolfsson, T.W. Malone, V. Gurbaxani, A. Kambil. 1994. Does Information Technology Lead to Smaller Firms? Management Science, 40(12), 1628-1645.
[13] R.H. Hayes, S.C. Wheelwright, K.B. Clark. 1988. Dynamic manufacturing: Creating the learning organization. New York: Free Press; London: Collier Macmillan.
[14] J.R. Galbraith. 1977. Organization Design. Addison-Wesley Publishing Company, New York (NY).
[15] R. Glazer. 1993. Measuring the value of information: The information-intensive organization. IBM Systems Journal, 32(1), 99-110.
[16] G.P. Huber. 1990. A Theory of the Effects of Advanced Information Technologies on Organizational Design, Intelligence, and Decision Making. Academy of Management Review, 15(1), 47-71.
[17] F. Tsung. 2000. Impact of Information Sharing on Statistical Quality Control. IEEE Trans. on Systems, Man and Cybernetics – PART A: Systems and Humans, 30(2), 211-216.
[18] A. Pete, K.R. Pattipati, D.L. Kleinman, Y.N. Levchuk. 1988. An Overview of Decision Networks and Organizations. IEEE Trans. on Systems, Man and Cybernetics, 28, 172-192.
[19] F. Taylor. 1947. Scientific Management. Harper & Brothers, New York (NY).
[20] M.D. Giudice and V. Maggioni. 2014. Managerial practices and operative directions of knowledge management within inter-firm networks: A global view, Journal of Knowledge Management, 18(5), 841-846, 2014.
[21] J. Bolton, K. Kim, R. Vertegaal. 2012. A Comparison of Competitive and Cooperative Task Performance Using Spherical Flat Displays, Proc. of CSCW 2012, 529-538, Seattle (US), Feb. 2012.
[22] M. Weber. 1968. Economy and Society: An Interpretive Sociology. Bedminister Press, New York (NY).
[23] J.A. LePine, J.R. Hollenbeck, D.R. Ilgen, J. Hedlund. 1997. Effects of individual differences on the performance of hierarchical decision-making teams. Journal of Applied Psychology, 82(5), 803-811.
[24] P. Bolton and M. Dewatripont. 1994. The Firm as a Communication Network. Quarterly Journal of Economics, 109(4), 809-838.