UNIVERSITY OF BOLOGNA – ALMA MATER STUDIORUM
MASTER OF SCIENCE IN ENGINEERING AND COMPUTER SCIENCE
“DISTRIBUTED SYSTEMS” COURSE – PROF. ANDREA OMICINI

Workflow Management

Workflow basics and evolution towards coordination

Emilio Fimiani

19/07/2015

Nowadays, the integration between process management and distributed systems has significantly grown. This paper tries to delineate the main features that affected and are going to interest workflow management.
To begin with main process management definitions will be provided, then, there will be a focus on workflow concepts and workflow management technology. Finally, social organization of work, coordination and new agents-based development will give a view of old and new fields that are going to improve workflow management technology.

Summary

Workflow Management

Workflow basics and evolution towards coordination

Introduction

In today's world, the emphasis on innovation, especially process management and networking, has really grown due to new Internet and Intranet developments. As a consequence, enterprises must deal with internal and external newness condition that could determine the organization life to come.

Probably, the most difficult element to cope with is complexity. In fact, organizations must think about several features and handle a lot of issues. For examples, the steady market pressure for cost and lead-time reduction and globalization concept. As a matter of fact, increasingly firms are becoming more internationalized.

Last but not least, globalization leads to a significant customer satisfaction that contributed to customize every little product. The evolving product differentiation has caused a more fragmented market since 1980s. Over this period there were two global developments that changed the political economy of industrial societies: technology change acceleration in production branches (Albrecht, 1982) and transportation and communication means that introduced the constitution of integrated world market and a progressively fierce global competition (Gunn, 1987).

These progresses encouraged ingenious managers and students to switch from Fordism-model to new organizational alternatives. Stemmed from this new approach, at the beginning of 20^{th} century Taylor scientific management was born.

Accordingly, at this moment the division of labor and industrialization induced a focus on routine aspects of work activities. Therefore, it began a systematic investigation of work methods separation of routine tasks from planning to control.
On one hand thanks to this research most of the tasks, roles, rules and procedures regarding offices and production work were determined. On the other hand, it centralized control functions, it caused problems on persons' health and it didn't cope with changing customer demands, innovation in technological products and increasing requirements. These dynamic environment factors were an obstacle to economic and social development. After all, Taylor scientific management target was to control and monitor the work that necessarily produced a loss of flexibility (AWK, 1990).

As a result, a new process management model had to be defined starting from the fundamental change of perspectives. What's more, it had to change from a production centered (transformation) model to a market (language) model. This change shifted business focus from product quality to customer satisfaction.

In 1990s organizational and management alternatives were born due to focus on the structure and behavior of the organizations upon processes rather than on functional responsibilities (Butera, 1993). Over this period there were two main breakthroughs in the organization (Butera and Schael, 1997):

- Processes take command over formal structures and work organization
- Increasing orientation to process performances and to process control tend to reshape organizational structures tasks and work rules

The result of these perceptions was the process management view. This concept reunified simplest and most basic tasks of industrial work into coherent business processes. It was a new approach that allowed to move toward the transformation of traditional bureaucratic organization into market-oriented process organization.

Process and workflow definition

As stated before, a new view was born at the beginning of 1990s. In this section, we're going to describe the main concepts that characterized process management view.

Process definition

A process is a set of rationally linked tasks through which an organization generates value to reach its targets.

Processes have two important features: they cross organizational boundaries, separating the functional units in the hierarchical organization and enterprise being part of network organization, and they have customers, who are internal and external recipients of the processes' output.

There are two kind of processes:

- Primary/operational processes, as material, informational and relational/communicative processes
- Change processes, continuous processes' development and redesign (Butera, Schael, 1997)

There are many positive points on using process working approach, probably the most important is that it improves response times, service and quality, cutting across departmental demarcations.
At the beginning, the most known method to manage processes was the procedure. It involves multiple functions and it progresses step by step as it passes between the functional units. This approach was cost and time consuming. What's more, it didn't fit in the concept of process organizations. So, it was invented a new way for looking at processes: the workflow.

Workflow definition

Workflows were initially developed as an evolution of the office information systems of the 1970s. Then, it has been used in the most different and effective ways.
A workflow is a unit of work within a process generating products or services related to customer satisfaction. In other words, it is a sequence and interrelation of materials, information, activities and communication within a process. It's composed by a main customer served by a supplier or cooperative network that works towards satisfaction of the main or secondary customers of the process. Broadly speaking, in the workflow environment there's a component that supports the workflow specification and another one that supports the workflow execution.

On the whole, work units are assigned to participants, which may link into an organizational model, that must look up at a specific timing given by explicit and accurate representation of the temporal order of the various activities of the business process. It involves both people and software applications.

There are two different types of workflow:

- Case-based workflow, central to organizational business. It consists of standardize workflow, permanent process-team and microstructures
- Ad hoc workflow, useful for specific, temporary needs of unique project teams

What's more, a workflow can be examined from a number of perspectives:

- control-flow perspective, the temporal order of the various tasks in a workflow
- data perspective, how data is defined and passed between workflow elements and/or how the external environment is captured
- resource perspective, how resources are involved in task execution

Clearly, the perspective choice is related to the circumstance.

There are number of benefits of using this approach. In addition to customer satisfaction there are cost reduction, strongly related with productivity increases, and flexibility that allows adaptation to changing circumstances because the process is monitored more closely, and effectively, this is the reason why rapid change in response to evolving market conditions are achieved more easily. Lastly, it improves quality of working life . In fact, while in the functional organizations people worked on pieces of a task in the whole process, in process working people are encouraged to carry out entire tasks from the beginning to the end, monitoring, controlling and handle exceptions themselves, as a consequence a lot of overhead costs are cut.

Workflow Management

Workflow management systems (WfMS) are a class of software that supports business processes by taking on their information logistics, i.e., they ensure that the right information reaches the right person at the right time (van der Aalst and van Hee 2002). It helps organizations to specify, analyze, execute and monitor workflows.
The information logistics of business processes can be captured by a workflow or process modeling language. Different workflow management systems may be implemented supporting the use of different languages.
Workflow management technology will be better explained in the next chapters.

Overview of Main Concepts of Workflow

Workflow Conceptual models were divided into 3 main models: the organizational (OM), the informational (IM) and the process one (PM). They provide an answer to the question: what to define, i.e. what are the entities involved in a workflow?

The organizational model has 2 roles:

- It structures resources in classes of objects with the same characteristics
- It attributes to each resource authorization to perform tasks.

Tasks are assigned to performers in real time so there aren't consequences if a participant is changed. Sometime, organizations don't want to assign a task to an individual resource, in this case, there are Roles and Organizational Units abstraction that supports business process quality.

The second conceptual model, IM, describes the structure of the forms, documents and data that are used and produced by a workflow.
There are many advantages if the structure of these information is known. For example, it can be used to determine whether a task is likely to be executed or not (pre-condition of a task).

The *last model,* PM, defines the component tasks, their coordination, the information and performers involved in each task. Task components and coordination are expressed using a language.
Broadly, tasks consist of: an identity, a pre-condition situation (resources or data availability, event occurrences), an action to perform, and a post-condition configuration to reach in order to consider the task as completed.

The first 2 models affect the last one. The organizational defines and organizes set of potential performers, whereas the informational allows access to the objects to be processed.

Workflow Description Language

Workflow definition language defines how to express the workflow conceptual models, i.e.: How must workflows be expressed? Or, what is the expressive power required for their definition?

Workflow description languages are used to design workflow models in order to capture processes at a level of detail that is sufficient to enable their execution. Outlining complexity of working processes raised new issues and challenges with regard to human activity modeling, for example, due to large component of unstructured and creative work. Consequently, process modeling languages and representations were evaluated in terms of useful representing and reasoning about the various aspects of processes. According to this, were created almost the same number of languages and existing systems. Even if there is a considerable amount of languages, the ideal one should have a very large expressive power to describe the 3 previous models in a uniform way, or at least languages allowing interactions between these models.

Petri-nets was primarily used to model structured systems such as computer programme, factory production lines, etc…. Then, different languages types were born until Functional languages that allow to represent the relationship among inputs and outputs. Moreover, Data description languages like Entity-Relationship, relation or object were usually used to describe information model.

Ultimately, the most actual used are YAWL, a dedicated workflow specification language developed in the academic domain and support most workflow patterns (i.e.: Petri nets), BPEL, a web service language that define executable process and BPMN, like BPEL, a mainstream language for capturing business processes from a practitioner's point of view. It is designed for business process modeling to specify workflow. For process execution it has to be transformed to models specified in an executable language (BPEL or YAWL).

BPMN and BPEL

BPMN is a business processing modeling notation that supports analyst decision making. It's based on scenario analysis, simulation and other techniques. BPMN process models aren't meant to be directly executable. On the other hand, BPEL helps organizations defining class of business processes for Web service interactions. Control-flow dependencies and communication actions interact through constructs corresponding to parallel, sequential, and conditional execution, event, and exception handling.

BPMN and BPEL are used together the first one is used for the process modeling whereas BPEL allows for the specification of executable business processes, and therefore can be used to support the execution of BPMN models.
Obviously, there are obvious drawbacks to modeling and execution separation and it's important to understand if there are more advantages or disadvantages. Sometimes, the languages are based on different paradigms or the modeling language contains potentially complex concepts and little consideration was given to their precise meaning.

YAWL and Its Formal Foundation

As will be mentioned in the next section, the original 20 control-flow patterns were used to evaluate various workflow and process modeling languages, standards, and workflow products.

The evaluation of various workflow and process modeling languages, standards, and workflow products using the original 20 control-flow patterns showed that Petri nets have at least three distinct advantages for being used as a workflow language: formal semantics, state-based instead of (just) event-based, and abundance of analysis techniques (van der Aalst 2000).

They are quite expressive compared to many process languages. In fact, they offer direct support to all state-based patterns. Nevertheless, there are serious limitations when it comes to capturing three categories of patterns: (1) patterns involving multiple instances, (2) advanced synchronization patterns, and (3) cancelation patterns.

For instance, the number of multiple instantiations may be known a priori or not be known until the process proceeds to the next part (at runtime). In high-level Petri nets, it is possible to use advanced constructs to capture multiple instances of a task or a sub process. However, there is no specific support for patterns involving multiple instances, and the burden of keeping track of splitting and joining the various multiple instances is borne by the designer.

The observation of the limitations in Petri nets for capturing certain workflow patterns triggered the development of a new language – YAWL (Yet Another Workflow Language). YAWL took Petri nets as a starting point and introduced mechanisms that provide direct support for the control-flow patterns.

Petri Nets

A Petri net (Murata 1989) is a directed graph composed of two types of nodes: places and transitions. Usually, places are represented as circles and transitions as rectangles. An arc in the net may connect a place to a transition or vice versa, but no arc may connect two nodes of the same type. A transition can have preceding places (input places) and succeeding places (output places).
Places may contain zero or more tokens, which model the flow through the system. The state, often referred to as marking, is the distribution of tokens over places. The state of a Petri net changes when one of its transitions fires. A transition may fire if there is at least one token in each of its input places (enabled transition). When a transition fires, it removes one token from each of its input places and adds one token to each of its output places.

YAWL

In contrast to Petri Nets, YAWL (van der Aalst and ter Hofstede 2005) the syntax of YAWL allows tasks to be directly connected, which helps compress the visual representation of a YAWL model.

Usually, YAWL process definition consists of tasks (i.e., transition-like objects) and conditions (i.e., place-like objects). It starts with a unique input condition and a unique output condition.
A workflow specification in YAWL is a set of workflow nets which forms a directed rooted graph with atomic and composite tasks. Both types of task can also be multiple instance tasks at the same time and thus have multiple concurrent instances at runtime. Each composite task refers to a net that contains its expansion. Atomic tasks correspond to atomic actions, i.e., actions that are either performed by a user or by a software application.
Finally, YAWL provides a notation for removing tokens from a specified region upon completion of a certain task, a generalization of the cancelation patterns.

In addition to the control-flow definition, the YAWL model also allows for the specification of the data perspective and the resource perspective.
The data specification defines data elements and their usage for exchanging information with the environment, for conditional routing, for creating and synchronizing multiple instances, and so on. Data are represented in XML and data manipulation relies on XML-based standards like XPath and XQuery. The resource perspective specifies task-resource allocation for each task within the process. Note that the term "resource" here refers to human resource, e.g., a role or a participant.
YAWL improvements were also possible because this language implemented and used workflow patterns as initial requirements.

Workflow Patterns

Workflow patterns indicate requirements for workflow languages; they are a specialized form of design patterns in the area of software engineering. They refer to recurrent problems and solutions related to the development of process-oriented application. Patterns are an abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts. They provide independence from the implementation technology and the essential domain requirements.

The Workflow pattern initiative was established in late 1990s with the aim of delineating the fundamental requirements that arise during business process modeling on a recurring basis and describing them in an imperative way. They identified 20 and 23 new patterns (Russell et al. 2006B). In total, 43 control-flow patterns classified into 8 categories. This kind of patterns describes structural characteristics of a business process and the manner in which the thread of execution flows through the process model. Then, they identified 40 data patterns (Russell et al. 2005b), series of data characteristics that occur repeatedly in business processes. Furthermore, they identified 43 patterns (Russell et al. 2005a) for the resource perspective into 7 categories. These patterns define various ways in which resources are represented and utilized in business processes. Last but not least, they identified various causes of exceptions and the various actions that need to be taken as a result of exceptions occurring.

The design patterns range from fairly simple constructs present in any workflow language to complex routing primitives not supported by today's generation of workflow management systems. In fact, workflow specifications must be extended to meet newer processing requirements. The control flow constructors require a fundamental insight and analysis.

The primary task of a workflow management system is to enact case-driven business processes by allowing workflow models to be specified, executed, and monitored. *Workflow process definitions* (workflow schemas) are instantiated for specific *cases* (i.e. workflow instances) and defined to specify which *activities* need to be executed and in what order (i.e. the *routing* or *control flow*). Activities are connected through *transitions* and we use the notion of a *thread of execution control* for concurrent executions in a workflow context. Activities are undertaken by *roles* which define organizational entities, such as humans and devices.

Control data are data introduced solely for workflow management purposes, e.g. variables introduced for routing purposes. *Production data* are information objects (e.g. documents, forms, and tables) whose existence does not depend on workflow management. Elementary actions are performed by roles while executing an activity for a specific case, and are executed using *applications* (ranging from a text editor to custom built applications to perform complex calculations).

Each workflow language can be formally described by a set of primitive modeling constructs, syntactical rules for composition, and the semantics of these constructs. It's important to focus on workflow patterns that originate from business requirements. The semantics of these patterns are much less formal because we cannot assume a (formal) language. In fact, patterns are context-oriented, i.e., a workflow pattern typically describes certain business scenarios in a very specific context.

Technology architecture for cooperative computing systems

Heavy use of software layer which takes care of message exchange and routing due to development of workflow management technology with the indicated functions and integration of data, information and communication flows.

Message control is a key component of a technological architecture. It's the most appropriate to cooperative information services. Message control coordinates the passage of message between processing components, this involves routing, translation, sequencing and monitoring.

Workflow Execution Model

The workflow execution model defines how the workflow interpreter behaves once a workflow has been defined, how should workflow models be executed?
There are many alternatives depending on how transaction, tasks and exceptions are managed, so there’s no single algorithm for the execution.
Thus, the workflow interpreter must gather this information to decide, according to the context, what task, when, where and possibly by whom in push mode to perform.

Workflow management Technology

Workflow management technology supports business processes, integrating data flows, information, communication and knowledge management in a working process at any time. As a result it increases productivity and flexibility needed for business competitiveness.

This technology was developed through the years. At the beginning the technology emphasis was on document storing, tracking and retrieval in working processes. Then, workflow software was introduced through integrated office systems as electronic mail and in office information systems. Lastly, thanks to personal computer diffusion, it spread in networked environments.

Nowadays, workflow management systems are applications prototyped by users to determine the flow and define forms and applications to be used. For instance, API (Application Programming Interface) that has to be inserted manually and Back-end operations that are written by computing specialist (developer), e. g. db and technical problems. There isn’t’ an agreed-upon taxonomy of WfS, this is the reason why they are often characterized in terms of the type of process that they support: administrative, production, ad-hoc, and collaborative.

Generation	Distinctive property	Major characteristics
First	application specific	• workflow capabilities expressed in particular applications (e.g., document management, e-mail) • hardcoded process definitions • closed and proprietary
Second	factored application	• workflow capabilities factored out from the application domain • workflow as a separate application, • limited selection of third-party tools • process definition tailorable through script languages
Third (current)	tailorable service	• generic workflow services accessible to other applications through APIs • open standards-based architecture • full integration of third-party tools • tailorable through GUIs • proprietary workflow interfaces and interchange formats
Fourth (next)	embedded enabler	• workflow services fully integrated with other middleware services (e-mail, desktop management, directory) and Internet • standardised interfaces and interchange formats • workflow-enabled applications • ubiquitous but invisible workflow systems

Table 1.

Generations of workflow management technology

(from Abbott & Sarin, 1994:114)

WfMS is usually in connection with 3 types of data:

- created and used during workflow enactment, *Workflow relevant*
- internal to the WfMS and describing the progress of each workflow instance through time, *System and environmental data*
- manipulated by external tools, *Application data*

Workflow management technology definitions

It's important to analyze history, definitions and different ways of workflow management approach because it allows us to understand why it's changed and how to improve tool usage.

Overall, the workflow management is the first method facing the problem of offering integrated support to routine and non-routine work providing their users with some exception handling mechanisms. This is the most general definition but there were number of scientists that provided a personal definition.

In 1991 Hales & Lavery defined the workflow management technology as a proactive computer system which manages the flow of work among participants, according to a defined procedure consisting of a number of tasks

Accordingly to De Michelis and Grasso (1993) workflow management is a technique for process modeling and automation. Hsu & Howard (1994) said that the workflow system is an integrator of disparate mainframe applications.

Then, Then Ellis & Wainer (1994) defined this technology like systems that help organizations to specify, execute, monitor and coordinate the flow of work items within a work group of organization. It coordinates user and system participants, together with the appropriate data resources, directly by the system or off-line, to achieve defined objectives by set deadlines.

Workflow management technology fundamentals

The workflow management technology is measured through standard message based on templates or forms. The crucial point of workflow management technology is the analysis before implementation.

The workflow management target is matching the users' needs. The following sections collect the requirements that should be implemented in workflow management systems.

Functional requirements

Offered functionality of current systems and desired functionalities are quite different, but the gap is going to be reduced with new workflow management systems. The following are the desired functionalities of systems.

1) Every business specialist should effect full-blown applications. So the application development must be customizable in a graphical end-user environment.

2) Steps (actions), sequencing (order of actions), routing (who does the action) and conditions (rules for doing and routing actions) definition must be effective using a flowchart generator. Non-programmers experience suggests use of a picture-oriented approach describing a process (Swenson et al. 1994)

3) There must be simulation facilities, workflow analyzers that automatically check the workflow._E.g. automatical Petri net (Pinci & Shapiro 1993)

4) The workflow can be re-used (library of primitives and existing processes), dynamic and flexible as well as action assignment.

5) The workflow maps (abstraction vs. decomposition) should allow zooming processes. Everyone should understand processes and identify the main steps.

6) There must be a due date for the task accomplishment. Users must be alerted if actions are overdue or pending for a certain period and it must be activated an alternative workflow path for a period of the time-recovery. Workflow applications should notify user that there's a step or action to do.

7) Flows must be monitored. There should be management reports that represent the current state of the process (Swenson 1993:130).

System integration requirements

1) Process management tools should be customizable and easy-to-use by users as well as form generation and modification. There should also be an online help system

2) Data must have the possibility to be connected from different source as well as workflow messages. Offline paper documentation, external (mail, forms, scanned images,…) and also non-electronic resources

3) Link recurrent organization processes with the workflow

Workflow management system architecture and implementation

How to design and implement software supporting the main issues?
The most common architecture includes 3 functions that manage architecture and implementation issues, The WfMS functions are:

- The definition of workflows;
- The execution of workflows;
- The administration of workflows and the monitoring of their executions.

This abstract architecture, followed by most systems, consists of 6 components. The *Workflow Enactment Service that* supports the execution of workflow processes. The *Workflow Definition Tools that* helps the workflow designer in specifying and analyzing workflow models. The *Workflow-Client Applications that* invoke the Workflow Enactment Service and enable the end user to interact with the WfMS. *Invoked Applications* called by the Workflow Engine. *Administration and Monitoring tools* that enable the workflow admin to set parameters, follow workflow executions though time and react to possible problems. *Other WfMS* may be connected to the current one via interface 4.

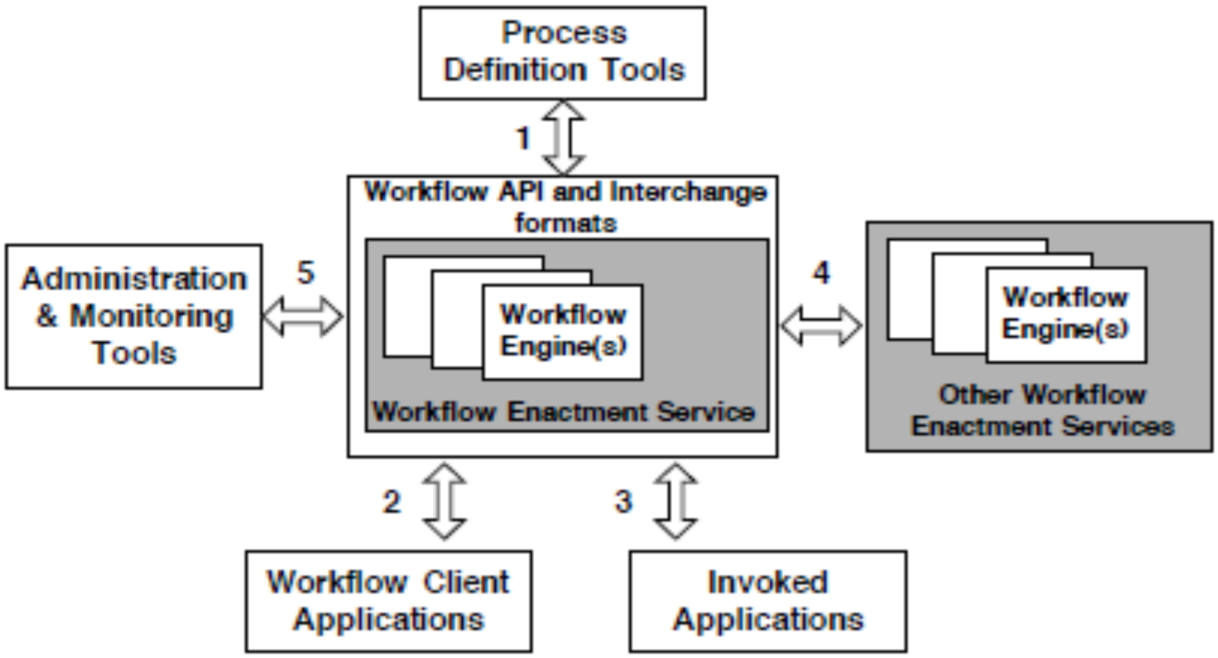

Fig. 1 - The Workflow reference architecture

The Workflow Application Programming Interface (WAPI) allows to unified and standardized interfaces that support communication between components.
WAPI is made of a set of API, interchange formats and protocols.

At the beginning the architecture remains general because there are several choices to make before WFMS work properly. For example, Workflow Enactment Service must define its decomposition into modules (scheduler, transaction-manager,..), the modular definition of these modules with a high-level formalism (Object, Agent, etc), the way these modules share data (message passing, repository, client-server, mediator), their control (e.g., centralized, distributed, event-driven).

Once refined, architecture must be implemented, trying to map conceptual formalism and the implementation language easily. Nowadays, due to its focusing on distribution and accessibility, Web became the most important technical infrastructure.

Comparing workflow management systems

Even if workflow management systems support a lot of patterns, most of them are difficult or impossible to realize using existing workflow management systems.
Generally, company focus on technical issues (which database management systems are supported?), the marketing strategies and other aspects to decide which is the best WFMS for their needs. Consequently, they don’t choose the most appropriate one.
WFMS choice should be taken after organization business processes analysis. In fact, it should support its workflow patterns.

Workflow patterns are also useful because they can be used to compare WFMS.

It's also true that if a pattern is not directly supported or even not supported at all by the workflow management system, this does not imply that it is impossible to realize the functionality. Indeed, it is clear that no tool support all the selected patterns.

First of all, the organization must select a workflow management system that focuses on the most needed patterns for the workflow processes at hand.
Second, the fact that a pattern is not directly supported by a product does not imply that it is not possible to support the construct at all. In fact, many patterns can be supported indirectly through mixtures of more basic patterns and coding.
Third, the patterns should focus not only on the process perspective (i.e., control flow or routing) but also on the other perspectives (e.g., organizational modeling).

The social organization of work

The individuals need to cooperate in order to work together achieving organization or own targets. In 1990 Schmidt argued that work has always been seen with a social and cooperative nature from a sociological perspective. This doesn't mean that the work is interdependent, so that people in a working process are mutually dependent on the activities of others as Popitz and Dahrendorf claimed in 1957 and 1959, but in any case their activities are linked by some mechanisms of interaction (shared resources, coordinated tasks,..).

In Schmidt work concept (1990), cooperative work pertains to the sphere of production and is related to *content* and *production* separating it from social interaction concept that is a bit more general. Schmidt view analysis shows a truly interesting point: o*rganizations exist because they produce something for customers* (whether as products or services does not make a difference for the cooperative nature of work). What's more, cooperation doesn't involve direct communication between agents as someone thinks. In fact, agents cooperate also without knowing each other. Sometimes like transients agents while in other cases as stable formation. It depends on work requirements: task or commitment as continuous or recurrent phenomenon, work environment and technical or human resources availability in a given time, the need to reduce costs, etc..

This chapter tries to introduce concepts of cooperative work and how it can be supported by computer systems. These concepts are related to Computer Supported Cooperative Work (CSCW) research field.

Computer Supported Cooperative Work

Computer Supported Cooperative Work (CSCW) was born in 1980s. Over this period the emphasis on personal computing grew followed by an incredibly positive trend towards networking all the end-computing devices. As a consequence, raised the awareness that networked infrastructure could support group activities in a more effective way.

The specific term CSCW was firstly used to describe the topic of the interdisciplinary workshop in 1984, intended to discuss how to support people in their work arrangements with computers (Greif, 1988).

Then, Bannon in 1991 argued that CSCW has emerged as a new interdisciplinary forum for research into the issues central to the design, implementation and use of technical systems which support people working cooperatively (Bannon *et al.,* 1991).

Group concept as the main unit to focus on is directly related to the term *groupware,* i.e., software for groups. Johansen (1988) defines groupware as:

> "a generic term for specialized computer aids that are designed for the use of collaborative work groups. Typically, these groups are small project-oriented teams that have important tasks and tight deadlines. Groupware can involve software, hardware, services and/or group process support." (Johansen, 1988)

There are several CSCW definitions, the following focuses on the understanding of *cooperative* in order to support the work by taking into consideration the results gained in the design process.
Computer Supported Cooperative Work (CSCW) is an identifiable research field focused on the understanding of nature and characteristics of cooperative work with the objective of designing adequate computer based technologies to support such cooperative work.

The role of computing in CSCW

Nowadays organizations think that *controlling data* means controlling the private or public organization's destiny and spend considerable amount of money to manage information but there is a missing perspective: organizations do not run on information, they run on relationships. Information determines the final result and is only exchanged within these relationships, i.e., customer satisfaction. Organization should invest in communication and knowledge management in order to serve customers by satisfying their needs trough cooperative work. CSCW support men managing the complexity and providing new ways of using traditional tools or new tools.

Developments in the field of CSCW might become the glue for the traditionally separate aspects of computing in work settings: single-user applications (personal computer) and organization-wide systems (transaction processing) (Schael, 1996e). This vision is represented in the figure below.

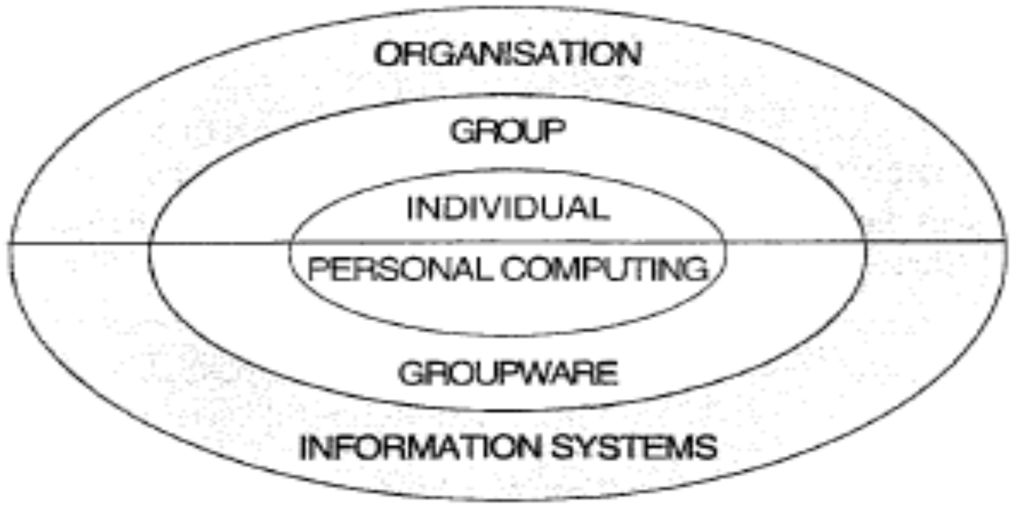

Figure 2 - CSCW in the context of organizations, groups and individuals (from Grudin 1994:94)

As can be seen, there are 3 different rings, each one represents one focus of computer development and of the principle customer or user of the resulting product. The focus of CSCW is the middle ring because it represents groupware, designed with groups in mind. It also interfaces with other 2 rings. The inner ring represents applications designed for individual users while the outer ring represents the entire systems designed to serve organizational goals.

The groups and organizations interactions analysis field study came from network analysis field started in 1980s (Burt, 1982; Knoke & Kuklinski, 1982; Burt & Minor, 1983; Rice & Richards, 1985). It was useful to define cooperative network.
Rice & Richards gave a network analysis definition in 1985:
The goal of network analysis is to obtain from low-level or raw relational data higher level descriptions of a system. The higher-level descriptions identify various kinds of patterns, or test hypothesis about those patterns, in a set of relationships. These patterns will be based upon the way individuals and objects interrelate in a network [...]. (Rice & Richards, 1985: 106)

A network consists of nodes that can be any point capable of participating in a relationship, and connections, used for relations in cooperation.
These 2 basic elements with bounds and behavior are the social network hinges.
It’s important to define the nature of the behavior which is of interest, the level of aggregation of that behavior (across time and units), the level of interaction necessary to constitute a significant relationship, and the bounds to be placed on the system under investigation.

However, cooperative networks description need other four similar dimensions: commitment, nodes, relation, and articulation. The final scope is to identify types of cooperation (coordination, collaboration and co-decision) (De Michelis, 1990; 1996; De Michelis et al., 1992; Schael & Zeller, 1990) as cooperative networks in an organization.

Network behavior is characterized by the articulation of work, work structures and composition necessary to produce the commitment, and synchronization or sharing among its members. Therefore, a cooperative network is a set of (made and completed) network's members and products commitments.
Commitments is a unit of work that must satisfy customer needs. Through language, it defines network member's roles and responsibilities, in other words the bounds of a cooperative network.

Nodes are virtual units defined by their *communication roles* and *communication management.*
They individually participate in a commitment as archivists, managers or performers. They are connected by synchronization or sharing within different degrees of variance among nodes participating in the commitment.

Basic distinctions for cooperative work

Several cooperative settings must be analyzed to identify cooperation types that can be found within a single organization (De Michelis, 1990; 1996; De Michelis *et al.,* 1992; Schael & Zeller, 1990). In fact, during the course of a process, three types of cooperation *(coordination, collaboration* and *co-decision)* may be needed.

Coordination is the first type of cooperative process. Individuals coordinate their actions with those of others, concurring to its completion.
The main issue is represented by *synchronization* of persons, actions and the consistency of the individual actions with respect to the whole process. Thus it's fundamental to break down the process into appropriate component actions.

The second type of cooperation is *collaboration.* Individuals work together to produce a final product that will be an entity as the unified results of all individual contributions of the cooperative ensemble.
The most relevant aspect is the individual's availability to share knowledge. In fact, *t*he success of collaboration depends on having a common understanding of the goal and of the process for achieving it.

The last type of cooperation is *co-decision.* Individuals collect all information needed in order to take a joint decision evaluating all the pros and cons.

So, when considering tools for supporting cooperative work, it is fundamental to understand that the types of coordination need different types of support. Moreover, information systems must be designed and built according to the labor organization and the cooperative activities which they should support.

Computer support for cooperative work

It's possible to identify 2 design spaces for cooperative work that are strictly related to different types and dynamics in cooperation previously discussed: *activity synchronization* and *information sharing.*

Generally, the main effective technique to coordinate work of a cooperative ensemble is to support synchronization of activities among its member providing facilities. On the other hand the main technique for promoting collaboration and co-decision between its members is to give information sharing facilities.

Ultimately, open services must support cooperation in the large, whereas cooperation in the small must be managed by domain services.

Obviously, there's a procedure for designing support for cooperative networks. The steps are the following:

- Identify the balance between activity synchronization and information sharing; its variance depends on the type and dynamics of cooperative networks and the capabilities of the individuals being part of it.
- Assess the current levels of synchronization and sharing
- Identify the support tools which will provide the optimum balance between activity synchronization and information sharing that are paramount in describing the support requirements of cooperative work.

The model to characterize cooperative networks

Commitment, nodes, connection and articulation are sufficient to analyze different the 3 different cooperative networks (coordination, collaboration and co-decision) but also the mixed one. In fact collaboration and co-decision can be mixed with coordination over space or time when there are complex situations or networks *(collaboration/coordination, co-decision/coordination).*

The model can be used to analyze cooperation occurring in a given situation and understand if cooperative network has to be re-designed to meet the real organization needs, the model for different purposes. Indicators for cooperative networks and their components are elaborated according to the structure in the figure below which allows calculation of synthetic indexes at different levels.

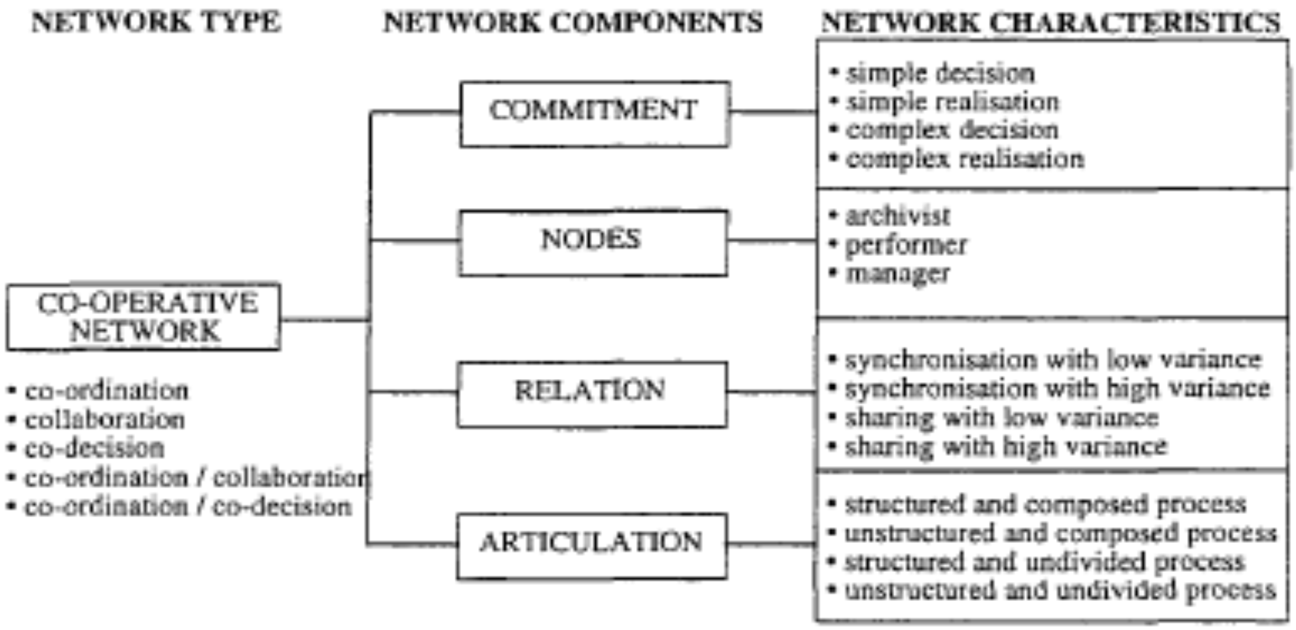

Figure 3 - Types, components and characteristics of cooperative networks

Cooperative networks performance is defined by *communication disciplines* (Petri, 1977, 1979). Although there's a common perception that the best network is the one more quick, reliable and at low cost, communication disciplines offer a theoretical framework for pragmatics in human communication, moving the observer (user) from outside to inside the system.

Shannon & Weaver (1949) developed the classical *communication theory* based on user's view describing communication by information *source, message, transmitter, signal, noise, receiver* and *destination.* The *unit of information* is the *bit.* Performance is defined by speed, robustness and efficiency measured by *channel capacity* and information transmission *rate*.

The most important characteristic the modem approaches aim to realize are flexible communication systems which are able to adjust themselves to the organization of the social system where they are implemented. So new evaluation criteria that represent different and complementary critical factors for cooperation were born: synchronization, identification, addressing, naming, delegation, authorization, reorganization, valuation, copying, cancellation and composition.

The communication disciplines are grouped together concerning:
• execution (synchronization, identification, addressing, naming),
• organization (delegation, authorization, re-organization) and
• administration (valuation, copying, cancellation, composition).

Clearly, the analysis of cooperation technologies characteristics must be evaluated in terms of their capability to cope with a specific set of communication disciplines.

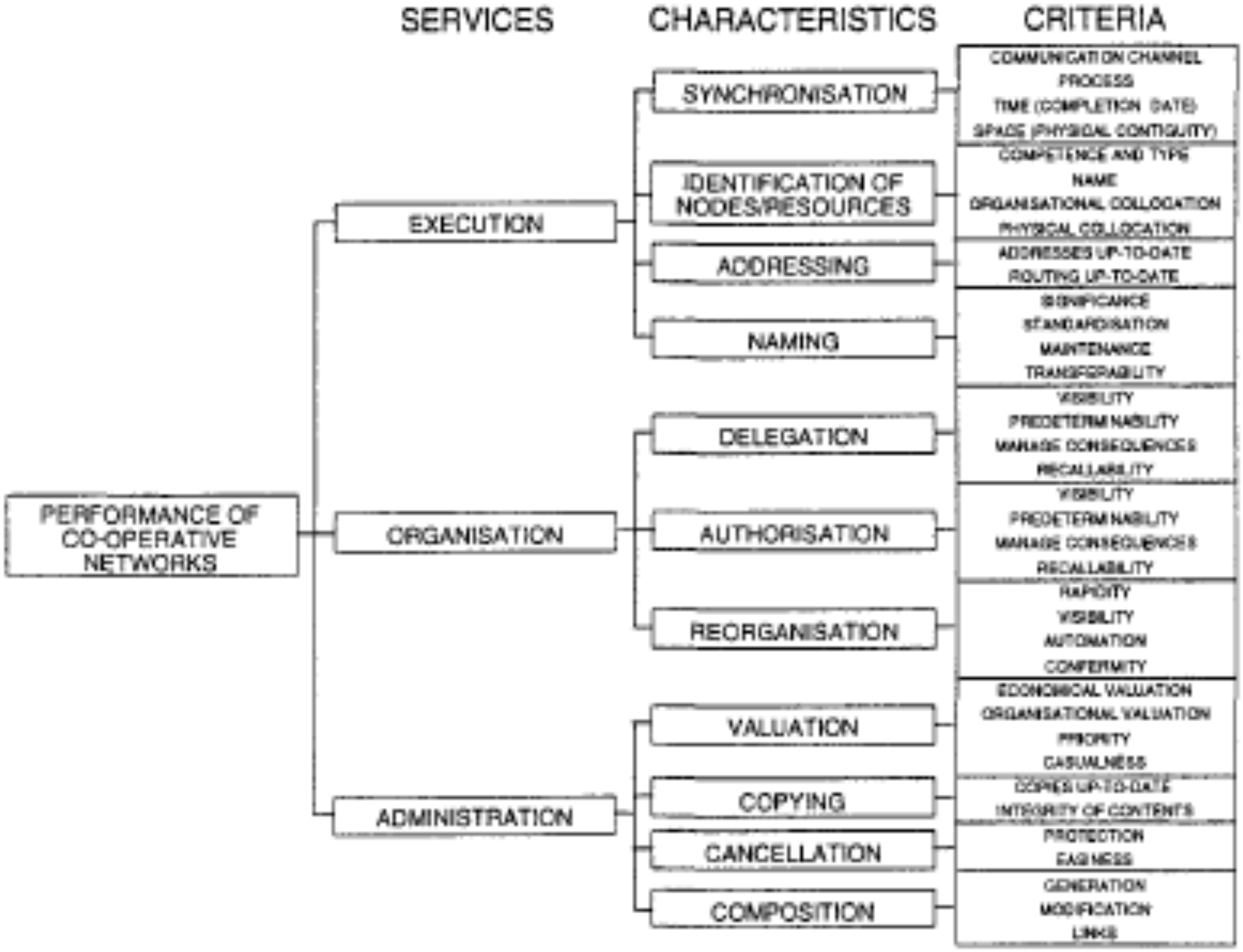

Figure 4 – Characteristics and criteria for cooperative network performance

Synchronization
Getting proper timing restraints. How messages (e.g., speech acts) and any other activity are synchronized among themselves.
Evaluation tools: a measure of time (clock, agenda) or indication of temporal relations which link activities. Partial ordering in terms of causality as opposed to an ordering in time.
Criteria for performance: synchronization of communication channels, process, time (e.g. completion dates for commitments) and space.

Identification of nodes and resources.
Conditions which allow one to recognize the source of messages which are exchanged on the network, and to establish the appropriateness of the message's receiver and destination.

Criteria for performance: identification of nodes' competence and type, identification of resources, its names, and its organizational and physical collocation.

Addressing

Routes and system paths which allow one to address a message on a cooperative network to an identified destination through a net of channels.
Criteria for performance: up-to-date addresses and routings.

Naming

Modality of how names are determined for persons and resources on a network. Bases for a common knowledge and to what extend its ambiguity could be avoided or resolved.
Criteria for performance: significance of names, standardization of naming, possibility to maintain names in time and to transfer them into other contexts.

Delegation

How and when a delegation can be made (delegation of tasks and of responsibilities). Cope with some effects on synchronization, addressing and other disciplines.
Criteria for performance: delegation's visibility on the cooperative network, its possible predetermination, its cancellation and the management of the delegation's consequences.

Authorization

Assigns and represents access rights to resources, priorities going along with these access rights, the rules and obligations for accessing resources and, if necessary, the rules for the supervision of all these activities.
Criteria for performance: authorization's visibility on the cooperative network, its possible predetermination, its cancellation and the management of the authorization's consequences.

Both, delegation and authorization, define the roles performed by persons on the cooperative network.

Reorganization

Rules for updating the cooperative network with regard to changes and adoptions of the network caused by the impact of the other communication disciplines.
Criteria for performance: re-organization being on time, its visibility, its degree of automation and its capacity for conformable adoption.

Valuation

Resolve the problem of limited resources on the cooperative network. All the time algorithms and rules are defined to resolve conflicts among nodes who want to use,

e.g., a communication channel. Nodes might get special rights on the use of resources or a node might be associated with a relative value on the cooperative network.
Criteria for performance: economic and organization (hierarchical) criteria for valuation and rules for priority or casualness.

Copying
Treats aspects of information handling which are related with copying documents. Integrity and coherence of copied information. Copying is related to other disciplines, e.g., delegation and authorization, which define who has sufficient rights to copy certain information.
Criteria for performance: updating of existing copies and the integrity of contents after a partial or complete copying of information.

Cancellation
The possibilities for cancelling information so that the main contents of documents and unique information cannot get lost. Protection and easiness are performance criteria.

Composition
The possibilities for composing a new document which can derive from existing documents or information. This is highly related to copying, but also to other disciplines like delegation and authorization which define who has the right to compose what kind of information.
Criteria for performance: generation, modification and links of information.

Dynamics of cooperative work

Dynamics together with articulation describes what is called *behavior* in network analysis.
Cooperation can occur in 2 different kinds of Dynamics: *in the small and in the large.*

The first one takes place in already *established ensembles* with usually well defined roles, rules and a common history that is shared among members. The boundary between such cooperative networks and the rest of the world is generally well defined. What's more since its past experience influences its future, there is also continuity in the ensemble.

Cooperation in the large is usually characterizes by occasional persons interactions. There aren't very specific roles and it's more complex to identify the right person that have to complete a specific task. Interactions occurring within these ever changing communities are related to the subject and to the goals. Finally, it's impossible to analyze past events to understand how the future could be.

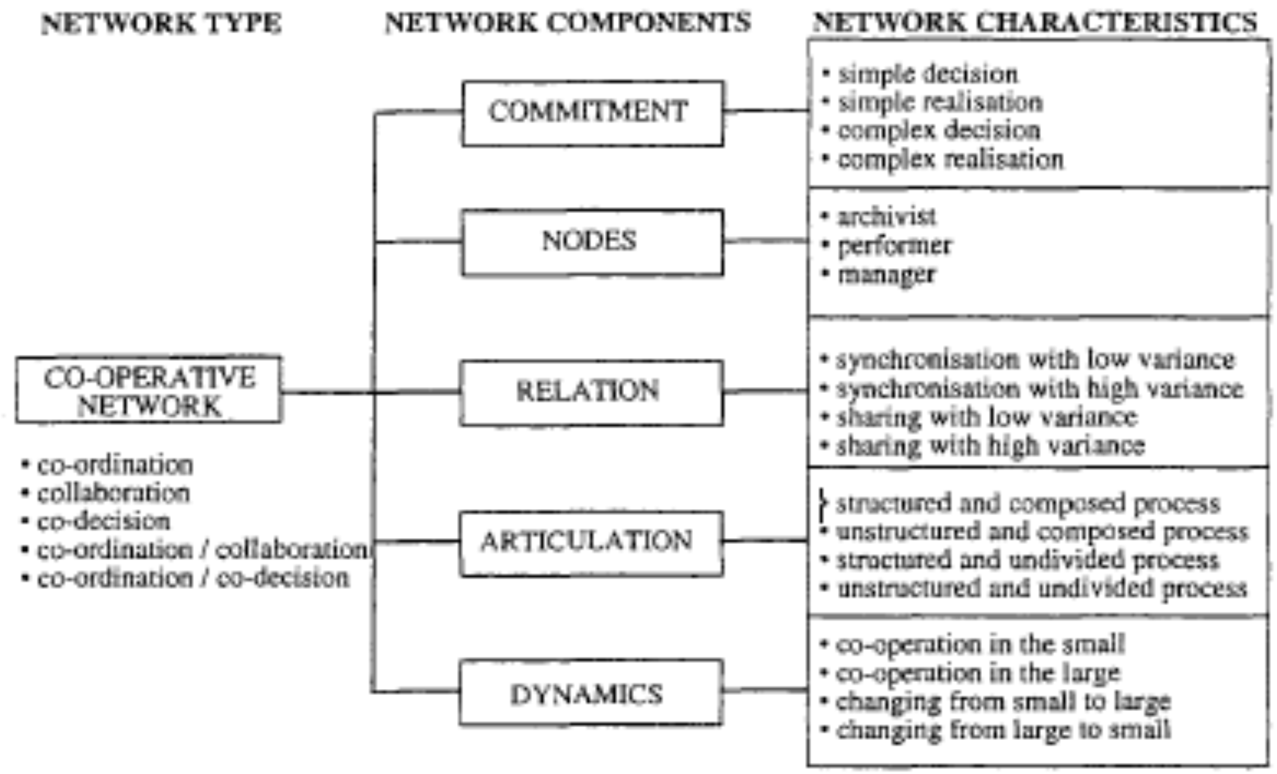

Figure 5 – Types, components and characteristics of cooperative networks including dynamics of coopeation

Inter-Organizational Workflow

Just as coordination is nearly invisible when conducted successfully, the supporting coordination technology works behind the stages in distributed applications on the Internet. In fact, following the user's view, a distributed system is a collection of independent computers that appears to its users as a single coherent system.

Therefore, workflow coordination issues are strongly related with Distributed System main issues: collaboration, through which many entities should work altogether as a single coherent system, and amalgamation that let many heterogeneous entities look altogether as a single uniform system.

A solution? The middleware for Distributed Systems and IOW for Workflows.
They enable meaningful interaction between autonomous distributed components communication issues like syntax, semantics,..
These layers hide differences in technology, structure, behavior, providing a common shared interface for both applications and components like operating systems.

Introduction

The figure below shows some of the features that must be taken into account when coordination technologies are evaluated. The way these tradeoffs are managed affects the design of a coordination manager. For instance, maintenance of a centralized server is easier than in a replicated architecture. In fact, it reduces the risks of inconsistencies regarding changes to the documents and deadlines.

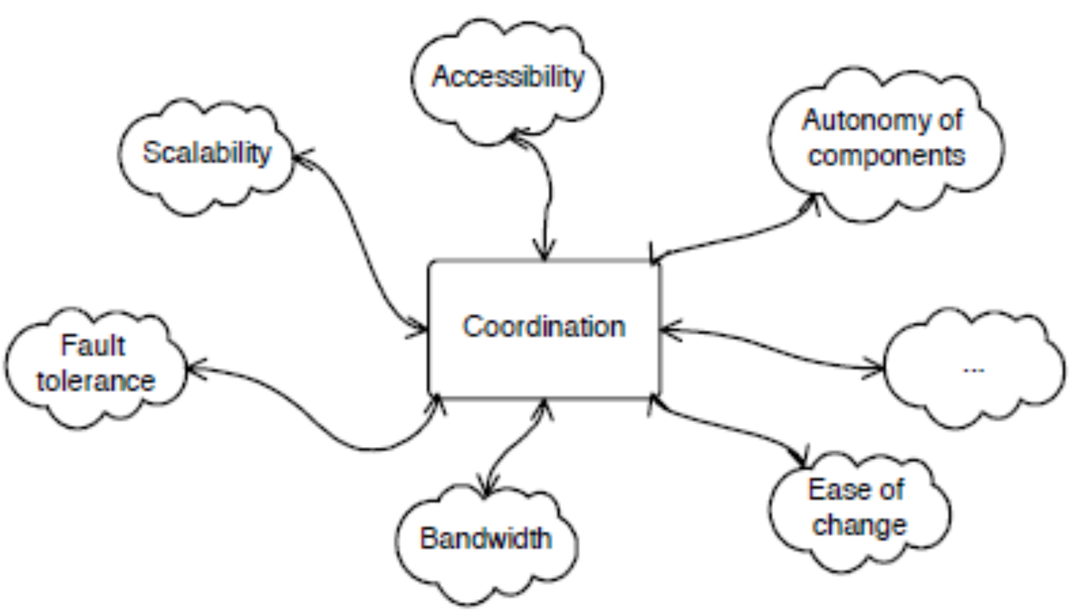

Figure 6 - Tradeoffs on coordination technologies

As stated before, the workflow decreased cost and time consuming. In addition to this, it automates repetitive tasks and promoted the distribution of documents, information and control, coordinating the different actors included in the business process.
Even though many efforts were made to improve this technology, it doesn't support business as well as it could do. This is the reason why, Inter-Organizational Workflow (IOW) technology is fundamental to help organizations with flexible and cooperative activities that must be coordinated to meet customer needs.

Clearly, even if businesses must achieve a common goal, there are several organizational cooperation forms. Nevertheless, it really emerges the need to coordinate the distributed effort. In this case coordination term means the work needed for putting together tasks, task sequences, task clusters, and even the work done in aligning larger units such as subprojects, in order to accomplish the work".
Thanks to recurrent nature of agents' coordination activities, organizations have generated different mechanisms that can support them defining the "expected" workflow. In other words, they have developed mechanisms for cooperating actors in order to describe their expected behavior.

Obviously, coordination complexity rises proportionally with activities distribution (in terms of time, space, knowledge, organizational backgrounds and the like). Since coordination work complexity across organizations is raised, WfMS have a significant relevance in an organization. Although computer-based technology improved the way coordination complexity was handled, there are a lot of problems. For example, WfMS are too rigid and semantic interoperability at the workflow and business level is still a challenge.

Accordingly, flexibility is one of the most important features a workflow must have due to business process evolution adaptation. Indeed, workflow execution must be correct and reliable independently by changes in the process.
In the following, we will refer to these workflows as Inter-Organizational Workflows (IOW).

What's more, because of the set of different forms of cooperation that are defined by different dimensions, IOW flexibility is more complex than other systems. There are numbers of aspects that can affected its flexibility, for example *Time spanning of the cooperation*, i.e. the organizations are joining for a transaction, a single project or are combined into an alliance, t*ypology of actors involved*, *motive for cooperation* (sharing skills, sharing resources).

The development of coordination technology is so relevant that it's become one of the most important dimension that an organization must deal with. First of all, businesses are moving towards an even more coordinated world, i.e. the incredible cloud technology increment. Secondly, issues related to this “new” field are significant, both for his complexity and his newness.

Obviously, IOW systems depend on the type of process they must support. There are two main cases of cooperation forms regarding business process workflow definition: No *definition or detailed definition of a process*. The first one realized coordination through awareness and explicit communication. It has only a common goal with self-organized agents and information sharing is a key to these types of IOWs. While the second one has different degree of tailorability of the tasks by the actors.

IOW Requirements: A Framework for Studying IOW

The *computing context* in which IOW takes place has several distinctive characteristics because of its distributed nature. These systems are usually built to make physical distribution irrelevant from user’s point of view. In fact, there are numbers of different sorts of transparency, that allows to hide non-relevant properties of the system’s components and structure, providing a higher level of abstraction to the users.

Table 2 - Different forms of transparency in a distributed system (ISO, 1995)

Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource is replicated
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource

First of all, as well as Distributed System goals, the computing context must be o*pen*, the composition of an IOW could evolve through time and that it’s not necessarily fixed a priori but should be decided dynamically at runtime. Generally, this characteristic generates 3 different issues: interoperability, that measures how easy / difficult is to make one component / system work with different ones based on some standard-based specification; portability, that measures how much an application (or, a portion of it) can be moved to a different distributed system and keep working; extensibility, that measures how easy / difficult is to add new components and functionality to an existing distributed system.

IOW coordination through MAS

Agents' field technology is developing in a very effective way. One of the biggest improvements regards real-world applications and system, dealing with their complexity, unpredictability and heterogeneity of information, resources and services.

Probably, the most natural application for agent-based models and technologies is Workflow Management system (WfMS); in particular Inter-Organizational Workflows (IOW), In fact, MAS (multiagent system) coordination infrastructures can support WfMS in 2 different ways.

On the one hand they can be artificial participants, being responsible of the automated execution of some tasks, or personal assistants of humans, as intelligent interfaces with respect to the workflow system, encapsulating the execution of tasks and individual activities.

On the other hand they can use coordination artifacts as the target of the agents activities, as computational objects being constructed, manipulated, evolved—, or the tools that agents use either individually or collectively to support their work toward goal achievement.

Generally, coordination artifacts capture the structure and function of workflow engines that coordinate the workflow activities, encapsulating the execution of workflow specifications (workflow rules in terms of coordination laws), encoded in some workflow language (i.e.:BPEL). The inter-connected coordination artifacts network represents the set of workflow that fits the needs for dynamics in IOWs.

Coordination artifacts are useful because they support awareness and coordination. Indeed, they improve mutual knowledge and they manage dependencies and interactions occurring among agents (and humans) involved in some shared activities.

To build flexible, dynamic and scalable workflow management architectures properties defined for the coordination artifact abstraction can be exploited.
First of all, malleability allows the coordinating behavior of workflow engines to be forged and tuned dynamically. Then, inspectability makes it possible to design workflow engines whose coordinating state and behavior are fully observable, thus providing a straightforward support for management activities, in particular for automated forms of on-line analysis of workflow data.

Finally, coordination artifacts must be linkable. As stated before, WfMS is the most natural application for agent-based models and technologies. Consequently, also IOW must deal with agent technology issues.

Probably, the most relevant are distributed workflow engines synchronization and coordination workflow activities coordination across the network.

Linkability scales with workflow activities distribution complexity because it allows coordination artifacts to be composed and interact with each other, by allowing multiple workflow engines (built as coordination artifacts) to be linked together, carrying on parts of the same (distributed) workflow within different execution contexts. So as to create a network of inter-connected coordination flows.

Flexibility and changeability

Nowadays, business scenarios are characterized by frequent changes, versatility and changeability are key elements. This is the reason why integration and coordination of distributed activities are really fundamental. WfMS and virtual enterprises (VE) support the integration of heterogeneous resources, services, and processes.

VE is a temporary aggregation of autonomous, independent and possibly heterogeneous enterprises connected through a network. The main target is to deliver a product or a service in response to customer needs, providing the flexibility and adaptability to frequent changes that characterize the openness of business scenarios.[3]

Obviously, to achieve the global system goal, the autonomous and heterogeneous activities must be managed properly by the VE. Generally, one of the main sources of complexity is the huge amount of dependencies among activities: sharing and exchange of heterogeneous information and resources, task assignments for VE business processes.

So VE activities can be seen and modeled as agents, featuring tasks to be pursued autonomously and pro-actively, typically involving the interaction with other agents and services, with the related coordination problems. In the context of VE, workflow management is the most obvious coordination problem.

The main issues of an infrastructure that supports VE management are the execution of VE processes, exploiting the more resources, services and processes from enterprise they can, and the integration of selected heterogeneous resources provided by single participants.

Therefore it’s obvious that the first requirement is of VE infrastructure is to promote the conceptual homogenization of VE resources, information, and activities.
It should support consistent interactions with shared services among processes, while minimizing VE impact on participant infrastructure. In this context, a VE infrastructure

should also provide abstractions and mechanisms for security, safety and privacy, so as to protect the VE from the outside world, while taking into account the mutual requirements of the VE individual components. Moreover, an infrastructure for VE management should help to govern the dependencies characterizing the execution of business processes, such as task assignments, prerequisite and temporal constraints, which require the development of a WfMS.

The engineering process of a VE requires both a suitably expressive model and a coherent runtime environment embodying and supporting the model. To be addressed in an application-independent way, these requirements typically call for an adequate infrastructure providing general-purpose services (such as coordination) to be configured and tailored according to the specific application needs.

One of the most effective model that is used for designing and implementing flexible agent-based WfMS is the tuple centre model. In fact, it allows to program the behavior of a tuple centre (programmable tuple spaces), defining reactions to interaction, by determining how a tuple centre should react to incoming/outgoing communication events.

Distributing Workflows

Since there are numbers of distributed workflows more or less complex that must be aggregated or combined into more complex workflow structures, it's fundamental to know how to distribute them among different engines.

There are 3 main points that must deal with. The first one is workflow composability. It allows understanding the best way to share information among several independent activities, and to coordinate them over a distributed, asynchronous scenario. The second one is ownership: stakeholders of independent activities within an IOW may not be willing to share everything, and to open up completely to the other participants. The final point is alterity: IOW does not belong in principle to any of the participant organizations. So, there must be an external infrastructure that has to support the main workflow engine, a third-party (an external host, a provider, a shared space) that is not directly owned / controlled by any of the workflow participants that hosts the workflow.

Architecture for Distributed Workflow

As stated before, it's necessary to build an abstract architecture for distributed workflow based on a coordination infrastructure to support the main workflow.
To begin with, infrastructure doesn't need to be pervasive because each of the participants should own and autonomously govern portions of the supporting infrastructure. There are only few processes that have to be shared.

So, each of the workflow nodes may contain one or more workflow engines, built as suitably specialized coordination artifacts. As a consequence, the supporting coordination infrastructure needs to be distributed, and owned by each of the workflow participants in terms of one or more workflow nodes, connected to each other through the network.

Then, composability of workflow engines is a fundamental pre-condition to distribute workflow activities. In fact, while interoperability of nodes is an obvious feature for any distributed infrastructure, the same does not necessarily hold for distributed coordination artifacts.

Instead, the ability to link different stages of separate but related workflows is essential to synchrony and coordinate the activities of multiple autonomous agents in a distributed workflow. Linkability of artifacts is also another essential feature for a coordination infrastructure to support distribution of workflow.

Then, another relevant feature that artifacts should have is alterity which means that a distributed workflow may in principle be hosted by a workflow node (call it the main workflow node, implicitly defining the main scope of the workflow). This node is not required to belong to one of the workflow participants.

Ultimately, an abstract architecture for distributed workflow should consist of a main workflow node that contains the general specification of the distributed workflow, a number of workflow nodes distributed across workflow participants and a network of workflow events exchanged among the workflow engines, in order to maintain global consistency of the distributed workflow activities.

In terms of the supporting coordination architecture, this essentially means that coordination artifacts should be programmable, working as general purpose workflow engines, distributable, so they can be located for each workflow node, and linkable, to reify workflow events, so that distributed workflow engines can synchronize and coordinate across the network, in order to support coordination architecture.

Agent Coordination

One of the most complex issue that a VE and a WfMS must deal with is agents' coordination. Clearly, there must be an infrastructure that could manage this kind of matter; this is the reason why Omicini and Zambonelli in 1999 developed TuCSoN.

TuCSoN is both a model and an infrastructure for the (objective) coordination of Internet agents. It's a collection of agents and tuple centers working together in a possibly-distributed set of nodes based on the notion of (logic) tuple centre.

TuCSoN supports multiplicity of independent environments such as internet nodes or administrative domain of nodes by providing multiple, independent interaction spaces, called tuple centers. They are abstract the role of the environment and can be framed as general-purpose coordination artifacts being customized to provide specific coordination functionality.
Agents (possibly mobile ones) access tuple centers by name, either locally in a transparent way, or globally on the Internet in a network-aware fashion.

Tuple centers can be seen as workflow engines, encapsulating the set of rules that constitute the workflow. As a consequence tuple centers represent workflow rules in terms of coordination rules programmed. So, coordination artifacts represented by tuple centers is the most important feature with respect to workflow management.

How does TuCSoN expressed coordination rules? Using ReSpecT specification language, a general-purpose interpreter for workflow specifications expressed in terms of logic-based workflow language, which are embodied within tuple centers.

TuCSoN as an infrastructure for VE and WfMS

Heterogeneity issue affects TuCSoN both at the model and at the infrastructure levels. From the technology viewpoint, TuCSoN guarantees portability among platforms because it's developed in Java. What's more, it's designed to support different kind of middleware because its architecture is based on a lightweight core.

Another outstanding feature is that agents can interact with other agents without knowing their identity or their position, supporting the integration between heterogeneous information sources. This is fundamental for a peer-to-peer interaction model between WfMS engines and participants, since they make it possible to go beyond point-to-point communication protocols.

TuCSon tuple centers are also more powerful than tuple spaces because they can be programmed to suitably react to incoming/outgoing communication events, by defining the coordination laws ruling agent interactions. This quality is to emphasize because adaptability is one of the most dramatic issue that VE participants must deal with using generic tuple centers.

Last but not least, "the ability of defining the dynamic behavior of tuple centers is also the key feature that allows TuCSoN to fully support heterogeneity at the process level and, more generally, to face the coordination of dynamic aspects inside the VE."

Conclusion

Workflow concepts permeate everyday life even if we don't perceive it.
Since 1970s, when workflows were initially conceived as an evolution of the office information systems, the improvement of this technology is really developed.

Firstly, new workflow-based languages were generated. Then, workflow management tool were invented, integrating specific process management approaches. Ultimately, social behaviors were incorporated in order to represent reality mechanisms more effectively.

As can be seen, workflow evolution involved not only its own technology but other fields as sociology. This discipline is the last innovation frontier.
Nowadays, social network analysis is probably the most studied subject. Reasoning about this, organizations can discover information about global trend or specific individual information. Clearly, there are many benefits of knowing this kind of information, for example, businesses can meet customer needs more precisely than having a direct customer approach.

Social network analysis isn't the only scientific method that affects sociology. As stated in the previous chapters, also agent-based modeling is a new rigorous technique that is expanding significantly and involves social behavior.

In fact, from a technical point of view, computer science and software engineering are undergoing a huge change due to the risen evolution complexity: computing systems will be anywhere, always connected and always active. As a consequence, they will be made by autonomous components that locally interact with each other, immersed in an environment. So, it's necessary to use new tools that make technology able to conceive, design and implement these concepts.

Thus, it's going to take place a "revolution" that will interest technology, process management and workflow in general but also the entire world. In programming languages field is called "the paradigm shift", the change from the previous paradigm (Object-oriented programming) to the new one (Agent-based programming).

Agent is the way!

References

[1]Alessandro Ricci, Andrea Omicini and Nicola Zaghini,
Linkable Coordination Artifacts for Inter-Organisational Workflow
Proceeding of 8th international conference on Coordination Models and Languages
Pages 228-246
Springer-Verlag Berlin, Heidelberg ©2006

[2]Alessandro Ricci Andrea Omicini and Enrico Denti,
The TuCSoN Coordination Infrastructure for Virtual Enterprises,
Proceedings of the 10th International Workshops on Enabling Technologies:
Infrastructure for Collaborative Enterprises (WET ICE .01)

[3]Alessandro Ricci, Andrea Omicini and Enrico Denti,
Virtual Enterprises and Workflow Management as Agent Coordination Issues,
International Journal of Cooperative Information Systems,
Vol. 11, Nos. 3 & 4 (2002) 355-379
World Scientific Publishing Company

[4]Chun Ouyang, Michael Adams, Moe Thandar Wynn and Arthur H.M. ter Hofstede,
Workflow Management [387-412]
Handbook on business process management 1 : introduction, methods, and
information systems / Ed. J. vom Brocke, M. Rosemann. - Berlin : Springer, 2010. -
(International handbooks on information systems). - p. 387-418

[5]M. Divitini, C. Hanachi, and C. Sibertin-Blanc,
Inter–organizational workflows for enterprise coordination
A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf,editors,
1Coordination of Internet Agents: Models, Technologies, and Applications,
Chapter 15, pages 369–398. Springer-Verlag, Mar. 2001.

[6]Thomas Schael,
Workflow Management Systems for Process Organisations,
Springer, second edition

[7]Wil M.P. van der Aalst,
Process Mining,
Discovery, Conformance and Enhancement of Business Processes,
Springer

[8]W.M.P. Van Der Aalst, A.H.M. Terhofstede, B. Kiepuszewski, A.P. Barros,
Workflow Patterns,

Distributed and Parallel Databases, 14, 5–51, 2003, Kluwer Academic Publishers