UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

A MULTI-VIEW FRAMEWORK FOR ORGANIZATIONAL PATTERNS

Enzo Colombo and John Mylopoulos

March 2006

Technical Report # DIT-06-017

A Multi-View Framework for Organizational Patterns

Enzo Colombo and John Mylopoulos

[1]Dipartimento di Elettronica e Informazione – Politecnico di Milano, Via Ponzio 34/5,
20133 Milano, Italy – phone +39-2-2399.3477 – colombo@elet.polimi.it
[22]Dept. of Computer Science, University of Toronto, 40 St. George Street, Toronto, Canada
M5S 2H4 – jm@cs.toronto.edu

Abstract. From the original work proposed by Gamma, design patterns have provided an important contribution to the development of well-structured software systems. This paper studies patterns supporting requirements analysis and, in particular, it discusses organizational patterns supporting the conceptual specification of business conversations among organizations within a virtual enterprise. These patterns are elicited from coordination and transaction costs economics theories and they are validated through a wide variety of non-trivial case studies. Organizational patterns are provided using three different concurrent views: an *intentional view*, a *strategic view* and a *process view*. Each view addresses a specific set of concerns of interest to different stakeholders in the system and, as a consequence, it has its own particular notation, rationale and constraints. Additionally, since these views are not fully independent, the paper discusses how elements of one view are connected to elements in other views.

1. Introduction

In information systems and workflow management systems literature, the impact of the structure of the organization on process management is not considered. The emphasis is on managing the execution of activities in processes, but not on their control and coordination, other than relating a sequence of execution of activities to a given goal [6].

In the cooperative information system manifesto, technologies for cooperation of agents with the same goal and acting towards the fulfillment of these goals are discussed, analyzing *systems*, *group collaboration*, and *organizational facets* [13]. Flexible systems and change management are emphasized, in particular with reference to organizational change. However, the organizational facet focuses on possible changes of goals in the organization and therefore on the identification and modeling of organizational objectives and on enterprise integration.

In workflow literature, the structure of the organization is represented only to indicate the roles of the agents executing activities. The problem of control in WfMS is studied from the point of view of guaranteeing a correct sequence of execution of activities. Therefore the research work focuses on methodologies for modeling workflow processes [4] and on activity scheduling [30].

Recent work on e-services [24, 27] has focused on representing the interfaces of services provided by different organizations in a cooperative environment in terms of exchanged messages, data and control dependencies, and e-service state evolution. Little attention is paid to the problem of designing these interactions according to patterns of interactions between organizations. However, modeling and understanding the organizational context within which cooperating relationships are deployed has been widely recognized as an important task of the requirements engineering process [18, 32]. This is especially true for cooperative and adaptive information systems where there is the necessity to reduce the effort needed to design, build and maintain cross-application business conversations. [7, 12]. Accordingly, this work studies and formalizes organizational styles providing a framework that captures experiences from literature and case studies into patterns supporting requirements analysis.

In the remainder of the paper, motivations to this work are presented depending on pattern-based requirements engineering and coordination theory literatures. Next, a multi-view framework for organizational patterns is presented. Patterns are then analyzed and formalized according to case studies from literature and the Italian district of Matera, Italy. Finally, conclusions are drawn and future work is discussed.

2. Pattern-based Requirements Engineering

Traditionally, the experience in system design has shown that experts working on a particular problem usually tend to capture existing, well-structured solutions reusing best practices for their needs [16].

This paper focuses on pattern-based requirements engineering, i.e. the branch of software engineering concerned with the real-world goals for, function of and constraints on software systems. In the last decade, lot of works have stressed the importance of patterns supporting requirements engineering. Patterns are provided at different level of concerns: they supports requirements elicitation, specification, analysis and validation. For example, a set of late requirements patterns for embedded software systems have been discussed in [19]. Patterns supporting goal refinement and operationalization are studied to support KAOS formal specifications [11]. These patterns are motivated by the observation that goal decompositions made by hand are usually incomplete and sometimes inconsistent. Moreover, recurrent patterns of task, conversation, physical actions and artefact usage have been observed during meetings among stakeholders aimed at establishing system requirements [21].

This works concerns with organizational patterns supporting early requirements analysis and extends part of the work conducted in the Tropos project. The main goal of Tropos is to provide methods, models and tools supporting the development of multi-agent information systems. In this context, researchers formalized agent-based software architectures inspired to organizational styles grounded on organization and strategic alliance theories [5]. Organizational styles are specified through strategic relationships according to the *i** social model and recently formalized through the Formal Tropos language [18].

In general, the adoption of social models facilitates goal classification supporting the specification of casual interactions among goals and linking goals to tasks aimed at their achievement. Moreover, social models as *i** are consistent with coordination theory [23] that constitutes the conceptual background for modeling business conversations among organizations. However, social specifications alone are inadequate when are required to model data and control flows and message exchange typical of business conversations [4].

Recently, some attempts have been conducted to embed business process semantics within *i** [14]. However, exceptional flows derived from goal violations are not considered and parallel tasks are not modeled explicitly. Moreover, this approach suffers of poor separation of concerns [20] since it is provided a single specification model that attempts to capture the overall system requirements. As a consequence, the contribution of this work is to study organizational patterns according to different views in order to address a specific set of concerns of interest to different stakeholders in the system.

3. Theoretical Background from Coordination Theory

In the organizational literature, the very activity of organizing is defined as the design of the rules for control and coordination [15]. It is straightforward how coordination and control rules are unnecessary when a single individual can accomplish all the tasks needed to reach a given set of objectives. On the contrary, when tasks increase in complexity, multiple individuals or organizations need to cooperate and, hence, to *control* and *coordinate* with each other.

Literature suggests a wide variety of definitions for control and coordination [2,3]. The diversity of these definitions illustrates the difficulty of defining control and coordination, and also the variety of possible starting points for studying these concepts.

However, in summary, these definitions can be reconciliated into two main perspectives, i.e. a *social perspective* and a *process perspective*. From a social perspective, control and coordination are defined as the act of managing interdependencies among actors in order to achieve goals [22]. On the other hand, from a process perspective, control and coordination concern with *(i)* task design and allocation to different actors, either individuals or organizational units and *(ii)* management of control and resource dependencies among tasks [31, 2]. Accordingly, different choices in task design and allocation translate into alternative business process patterns. Moreover, social and process perspectives are driven by high-level objectives of actors reasoning at a strategic level (*intentional perspective*).

Traditional literature distinguishes between two fundamental control and coordination styles: organizational *hierarchies* and *markets*. Within organizational hierarchies, actors are organized hierarchically, ranging from top management at the highest level to operations at the lowest level. Functional specialization is usually the criterion for specialization and organizational units are built around the specific set of functional competencies that they develop, such as research, engineering, production or marketing [26].

Organizations can also outsource part of their production and related decision-making activities to other organizations, such as customers, suppliers, consultants or commercial partners. A relationship between distinct organizations is implemented through the execution of economic transactions, defined as exchanges of economic goods and services ruled by a price system [10, 29]. This cooperation through economic transactions is referred to as market control and coordination and is considered an alternative to the hierarchical control and coordination among task executors within a single organization [23].

Organizational hierarchies and markets constitute the opposite ends of a continuum of coordination and control mechanisms [23]. Hierarchical and market coordination and control can mix and generate different organizational styles depending on the degree of *delegation*. Organizational styles include market relationships [29], long-term agreements (e.g. *comakership*) [25], vertical quasi-integration [3], relationships based on equity exchange [15] and vertical integration (hierarchy) [23].

Different styles are adopted depending on the minimization of control and coordination costs. Either *product or service complexity*, *environment uncertainty*, *frequency of business conversation*s, *asset specificity* and implementation of either *complementary* or *similar services* are some of the strategic variables that decision makers have to deal with when they adopt an organizational style in order to cooperate.

The focus of this paper is on inter-organizational styles and, in particular, on networks of juridical independent agents such as *virtual enterprises*. As a consequence, organizational styles based on equity exchange and on vertical integration are not considered in this work. Hence, Section 5 formalizes *market*, *comakership* and *vertical quasi-integration* according to different views also consistent with different perspective of analysis of control and coordination mechanisms.

4. A Multi-View Framework for Organizational Patterns

An organizational structure defines the way in which interrelated groups of actors manage their relationships in terms of control and coordination mechanisms. Typically, organizational structures are specified according to three levels of abstraction, i.e. *strategic*, *decisional* and *operative* [1]. Each abstraction involves different stakeholders in the organization and, as a consequence, requires its own particular notation. Figure 1 shows the relationship between *stakeholders*, *abstraction levels* and *modeling views* supporting the specification of organization structures.

4.1 The Framework

Modelers specify organizational structures using several specification methods in well-chosen forms. In this section, these specification methods are grouped into views and the contribution of each view is singularly discussed.

Intentional View. The intentional view supports reasoning on strategic objectives. A board of directors refines high-level strategies through a set of key abstractions. These abstractions are intentional elements such as softgoals, goals, task and resources [5, 32]. *Goals* represent requirements to be fulfilled (⬭ = goal); s*oftgoals* are similar to goal but their fulfillment is not clearly defined (☁= softgoal). A *task* is a structured sequence of decisions and actions aimed at producing an added value transformation of inputs into outputs (▭ = task) and, finally, information *resources* represent inputs to tasks (□= resource).

An *intentional diagram* shows a set of intentional elements and their logical relationships: *decomposition* (—+), *contribution* (→) and *means-end* (➝) links. Directors defines their high-level strategies and then, following a refinement process, elicit the set of tasks (and the corresponding resources) that should be performed to achieve their goals (and softgoals).

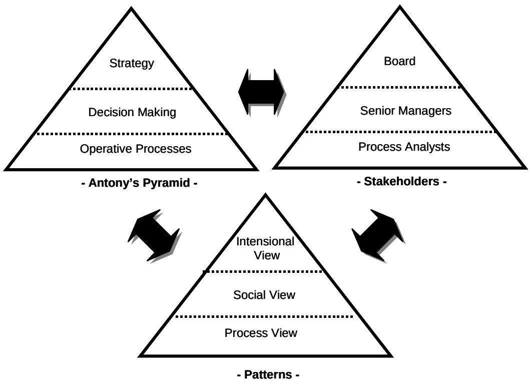


Figure 1. Relationship between views, stakeholders and the corresponding level of abstraction.

Social View. The social view concerns with the specification of social dependencies among organizations. Managers at the decision making level specify social dependencies through a model of strategic relationships [5, 32] complemented with the Formal Tropos (FT) language [14].

The *strategic relationships model* is a graph where each node is represented by an organization and each link between two actors describes a dependency in terms of intentional entities. A dependency formalizes an agreement between two organizations, i.e. a depender and a dependee (*depender* →— *int. entity* →— *dependee*). The type of dependency defines the nature of the agreement.

A goal (or softgoal) dependency represents the delegation of responsibility over the fulfillment of a goal (or softgoal) from a depender to a dependee. A task dependency represents the delegation of responsibility over the execution of a task from a depender to a dependee. With respect to goal (or softgoal), a task dependency is stronger since the depender also specifies how the task needed to fulfill a goal (or a softgoal) must be implemented. Finally, a resource dependency represents the need for an input that must be provided to a depender by a dependee. We note that actors' boundaries may embed intentional elements from the intentional view if they are involved in some way in the social relationships.

Formal Tropos (FT) is a linear-time temporal logic modeling actors, intentional entities and dependencies. It complements the strategic relationships model allowing the formalization of cardinalities and strategic policies. The formalization of cardinalities is essential to decide whether an instance of the specification is allowed or not. On the other hand, policies describe the social behavior of participating organizations depending on their strategic relationships. For example, FT is used to model policies implemented when a seller is able to provide more than one offer satisfying a delegated goal. Examples of FT specifications are provided in Sect. 5.1.2.

Process View. The process view takes into account some non-functional requirements such as flexibility, adaptability, self-tuning and controllability of business conversations. Process analysts at the lower level of our conceptual model (see Figure 1) describe the process view at several level of abstraction, each addressing a different concern.

At a highest level, the process view is a business conversation among organizations in terms of activities and control flows, resource assignments and information flows. This *process model* is a particular instance of statechart [17] enriched with (i) compensation actions derived from [7] and (ii) predicates around the correct execution of tasks and violation of goals [8]. Accordingly, the model formalizes both the standard and exceptional behavior of a business conversation where exceptions are typically triggered by goal violations and compensated at run-time.

Compensation actions are grouped into classes, i.e. delay (e.g. wait for, delay, …), informative (e.g. notify, urge, …), re-execute (e.g. re-execute, skip,…), re-negotiate (e.g. relax, tighten, …) and re-transact (e.g. delegate execution,…). On the other hand, predicates reason around (i) either the fulfillment or the violation of goals/softgoals, i.e. **Achieved**(*actor*, *goal*/*softgoal*); (ii) either the execution or the fulfillment of a compensation action, i.e. **Done**(*temporal condition*, *actor*, *task*), **Fulfilled**(*temporal condition*, *actor*, *task*).

At the lowest level, a specification of business conversation is complemented with a set of properties satisfied by instances of the model and, where possible, with a set of properties linking instances of the model over time. Accordingly, a process model is specified together with a set of liveness and safety properties that analysts have verified on process instances. Examples are process termination, maximum lead-time and properties around either the correct sequence of fulfillment of goals or control and information flows (see Sect. 5.1.3). Moreover, process analysts document their specification choices through a blueprint discussing:

- Rules supporting the refinement of intentional elements when they are mapped from a strategic relationship model to a model of business conversation. Refinement policies involve motivations for the decomposition of intentional elements and motivation around the way they are finally operationalized.
- Rules supporting the management of residual rights of control. These rules identify the group of actors that retains decisional power around compensation management where unexpected exceptions occur [4].
- Rules supporting the management of abort. They specify how organizations within a business conversation should behave when the conversation aborts.
- Rules supporting the management of time-outs.

4.2 Correspondence among Views

Views are not fully independent and, as a consequence, elements of one view are connected to elements in other views according to rules and heuristics. Accordingly, the goal of this section is to highlight these correspondences.

From intentional to social view. Only part of the intentional model is shared between the board of directors and the pool of managers at a decision making level. In particular, the board shares with managers intentional elements involved in the strategic relationships with sellers. Accordingly, the correspondence between these views is achieved either delegating or taking responsibility of intentional elements on the basis of different alternatives complying with the intentional diagram.

From social to process view. Goals and softgoals within the social model are mapped into pre- or post-conditions of tasks. Goals/softgoal that are delegated are translated as post-conditions of tasks fulfilling the goal/softgoal. Goals/softgoals that are retained are mapped into either pre- or post-conditions on corresponding tasks, which are executed by the owner of the goal.

Resources within the social model are mapped into input-output parameters for the evaluation of pre- and post-conditions on the availability of resource necessary to execute a task. Tasks are mapped into states of the process model. Moreover, pre-conditions of the considered task label ingoing transitions, on the contrary, post-conditions of the considered task label outgoing transitions. If the considered high-level task is refined into subtasks, subtasks are mapped as sub-states of the hierarchical state associated with the high-level task. Finally, actors are mapped into state labels in order to preserve knowledge of the organization who takes responsibility for the execution of an operative task. In particular, the actor executing a task is determined by observing the actor boundaries in the social model.

Once correspondence rules are applied, several specification choices must be addressed by process analysts in order to obtain a complete process model:

- Specification of the standard flow of activities either through sequences of states or AND/OR-states.
- Specification of the exceptional flow as a consequence of either goal or softgoal violation. Accordingly, modelers have to decide what class of compensation must be applied.

5. Organizational Patterns

In this section we describe three organizational patterns, i.e. market, vertical quasi-integration and comakership. The former is studied in detail and presented according to our multi-view framework. Vertical quasi-integration and comakership are instead presented shortly. However, a complete formalization of these patterns together with their possible deviations is studied through more case studies in [9]. In particular, [9] shows that a market pattern allows two deviation in its process view and that a comakership pattern allows one deviation in both its social and process view. Vertical quasi-integration does not allow deviations.

5.1 Market Pattern

A *market conversation* is defined as the exchange of economic goods and services ruled by a price system [10]. In particular, in a market system (see Fig. 3.(b)), a buyer aims at discovering a commodity minimizing its price. Coordination among cooperating actors is spontaneous and not planned beforehand [29]. In the following we describe a cooperating scenario involving an organization that provides services according to a market paradigm. The example also discusses the perspective of the potential buyer.

5.1.1 Case Study

Beca S.n.c. Beca is an organization producing polyurethane (a material derived from oil) with revenues about 4.5 mil/euro per year. When Beca receives an order, it schedules production and notifies the customer whether the order can be supplied or not according to lead-time requirements. Moreover, in order to improve quality of service (QoS), Beca implemented a customized information system to monitor production and obtained the ISO9002 certification. Each lot of polyurethane is therefore provided with a technical document certifying the quality of the product with respect to a set of quality parameters (e.g. granularity). This production process is organized as follows. Orders of polyurethane are received and automatically scheduled according to lead-time requirements. According to the production plan, pieces of raw polyurethane are cut either manually or with a semi-automatic machine. The polyurethane is shaped with a numerically controlled machine and glued together. Final quality control is performed according to ISO9002 norms.

Before performing an order, Beca *negotiates an agreement* with its potential buyer (i.e. Sofaland). Negotiation on price is based on polyurethane parameters such as granularity, pressure, inflammability, toxicity and resistance to traction. Moreover also lead-time requirements and ordered quantities impact on the price of polyurethane proposed to the buyer.

Sofaland S.r.l. (in a market system). Sofalands is part of the Natuzzi Group S.p.A., one of the primary organizations producing sofas worldwide. The Natuzzi Group exports around 92% of its production, supplying 3500 customers located in 137 countries. The success of Natuzzi is a consequence of a well defined strategic plan focused on high QoS, minimization of production costs, wide variety of sofa models and a particular attention to international markets.

The quality of polyurethane needed by Sofaland varies according to the model of sofas. Medium quality polyurethane (e.g. medium granularity) is used to stuff cheap sofas for the U.S.A. market. On the contrary, high quality polyurethane is used for the European market. During procurement, Sofaland gathers offers submitted by multiple potential sellers (one is Beca) that should comply with polyurethane requirements. Selection is finally ruled by the better price and the seller is committed with Sofaland to comply with the signed agreement. Moreover, Sofaland does not adopt a free-pass policy and it controls polyurethane samples for each supply. If polyurethane does not comply with the agreement, Sofaland is authorized either to require a new lot of polyurethane or to re-negotiate the agreement. Re-negotiation (within predefined intervals) is allowed since this polyurethane could be used to stuff sofas for the USA market anyway. Payment is therefore performed either if control activities do not show violations or if violations are successfully compensated.

5.1.2 Intentional View

Sofaland's board of directors defines the general strategy to buy polyurethane. In particular, the board of directors decides that a market system is the better control and coordination mechanism to interact with its partners (e.g. Beca S.n.c.). Indeed, this mechanism allow the minimization of conversation costs since the frequency of conversation is low, there is no need to deploy specific assets and the environment is not uncertain since polyurethane suppliers as Beca are well known for their high quality of service [29]. Moreover, polyurethane is a complementary but dissimilar good for Sofaland, and accordingly, it is cheaper to outsource polyurethane production than implementing its internalization [28]. The overall intentional schema supporting this evaluation process is provided in Figure 2.

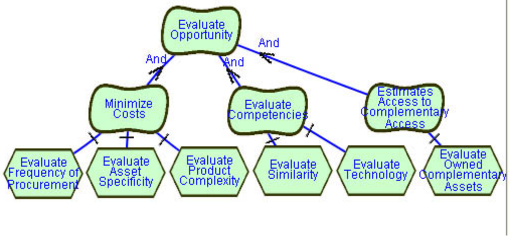

Evaluate Opportunity
And
And
And
Minimize Costs
Evaluate Competencies
Estimates Access to Complementary Access
Evaluate Frequency of Procurement
Evaluate Asset Specificity
Evaluate Product Complexity
Evaluate Similarity
Evaluate Technology
Evaluate Owned Complementary Assets

Figure 2 – Evaluation of opportunities associated with an organizational structure.

Moreover, Figure 3.(a) shows the intentional model associated with the delegated *"Buy Polyurethane on the Market"* softgoal. In order to fulfill this softgoal, it is necessary to *"Procure Polyurethane at the Better Price"* and *"Satisfy Customers"*. The former is achieved by Sofaland discovering polyurethane producers, evaluating offers of polyurethane suppliers to obtain the better price and managing polyurethane purchases. The latter is fulfilled either trusting the supplier or controlling the polyurethane lot before production. Figure 3.(b) shows the intentional view of a market pattern derived from the Sofaland – Beca scenario.

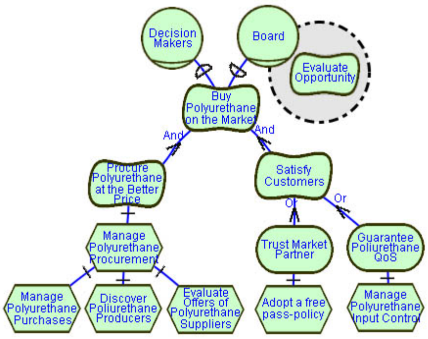

Decision Makers
Board
Evaluate Opportunity
Buy Polyurethane on the Market
And
And
Procure Polyurethane at the Better Price
Satisfy Customers
Or
Or
Manage Polyurethane Procurement
Trust Market Partner
Guarantee Poliurethane QoS
Manage Polyurethane Purchases
Discover Poliurethane Producers
Evaluate Offers of Polyurethane Suppliers
Adopt a free pass-policy
Manage Polyurethane Input Control

Figure 3.(a) –Intentional reasoning of Sofaland in a market system.

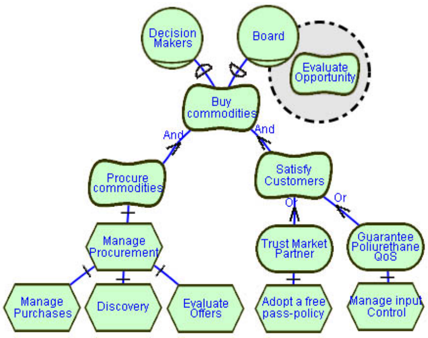


Figure 3.(b) – Intentional view of a market pattern.

5.1.3 Social View

Figure 4 shows the strategic relationships between Sofaland and Beca within a market system. Part of the intentional view in Figure 3.(a) is reported into Sofaland's boundaries. In particular, the *"Discover Polyurethane Producers"* task requires from Beca an offer complying with polyurethane requirements. We note that offers are generated in order to *"Fulfill Beca's Marketing Policies"* and *"Comply with Polyurethane Requirements"*. Moreover, the *"Manage Polyurethane Purchases"* task provides the final order to the *"Polyurethane Production & Delivery"* task according to the agreement.

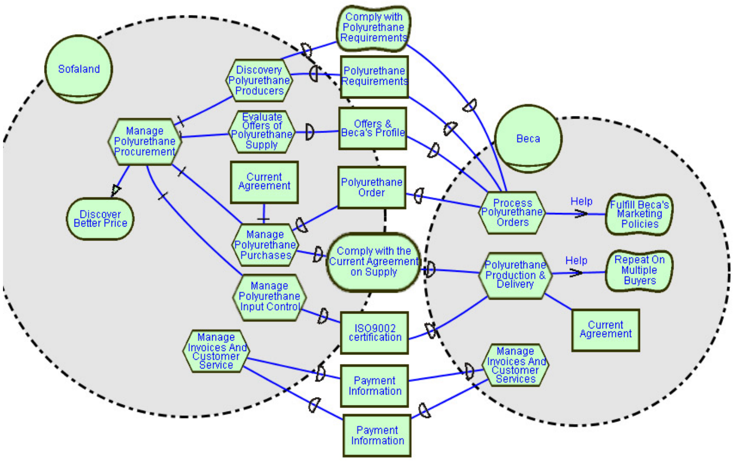
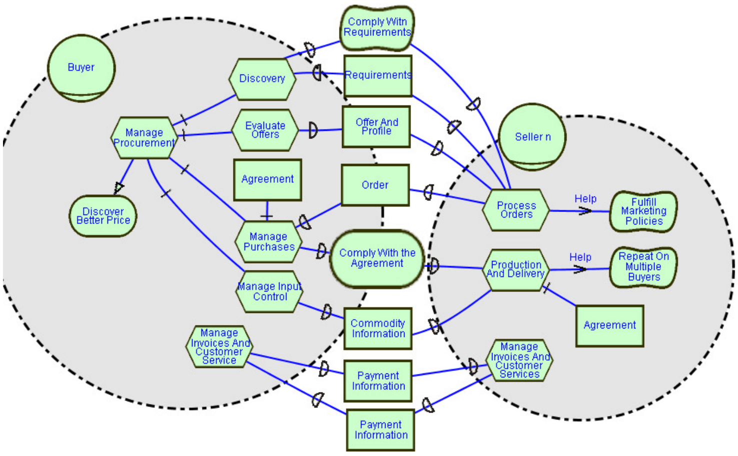

Sofaland
Comply with Polyurethane Requirements
Discovery Polyurethane Producers
Polyurethane Requirements
Manage Polyurethane Procurement
Evaluate Offers of Polyurethane Supply
Offers & Beca's Profile
Beca
Current Agreement
Polyurethane Order
Discover Better Price
Process Polyurethane Orders
Help
Fulfill Beca's Marketing Policies
Manage Polyurethane Purchases
Comply with the Current Agreement on Supply
Polyurethane Production & Delivery
Help
Repeat On Multiple Buyers
Manage Polyurethane Input Control
ISO9002 certification
Current Agreement
Manage Invoices And Customer Service
Manage Invoices And Customer Services
Payment Information
Payment Information

Figure 4 – Market relationship between Sofaland S.r.l. and Beca S.n.c.

The *"Polyurethane Production & Delivery"* task produces polyurethane in order to *"Comply with the Current Agreement on Polyurethane Supply"*. The agreement between Beca and Sofaland is therefore used to set the assembly line performing polyurethane production. Besides, the execution of the *"Polyurethane Production & Delivery"* task over time with different buyers allows Beca to improve its production process thus contributing positively to the fulfillment of the *"Repeat on Multiple Buyers"* softgoal. Finally, the *"Manage Polyurethane Input Control"* task receives ISO9002-compliant certifications together with the polyurethane supply and performs source inspections.

Figure 5 shows the *social view* derived from the example of market relationship between Sofa and Beca. This view specifies an actor playing the role of buyer and one playing the role of seller. Moreover, the social view models an abstraction of the vertical cooperating relationships between Beca and Sofaland obtained by generalizing the example from actors to roles and from domain dependent to domain-independent intentional elements.

Moreover, in the following, examples of structural properties constraining the representation of Figure 5 are specified with Formal Tropos (FT) [14]. The FT outer-layer of a market pattern is provided in [9].



* Roles and intentional elements are the parameters of this pattern
Figure 5 – Social view of a *market pattern*.

Each instance of the market relationship pattern includes exactly one Buyer actor
$\exists b: \text{Buyer}(b) \wedge \forall b1, b2: \text{Buyer}(b1=b2);$

Multiple instances of both the "Requirements" and the "Offer & Profile" resource (different instances have a different identification).
$\forall$ re1, re2: Requirements (re1≠re2 $\leftrightarrow$ re1.id $\neq$ re2.id)
$\forall$ op1, op2: Offer&Profile (op1≠op2 $\leftrightarrow$ op1.id $\neq$ op2.id)

The "Production & Delivery" task performed by a seller and the "Manage Purchases" task performed by the buyer share the same agreement
$\forall$ mp: ManagePurchases, p: Production&Delivery (mp.agm=p.agm)

If more then one offer fulfills the softgoal "Comply with Requirements", the offer associated with the lower price is selected thus satisfying the "Negotiate Better Price" goal .
$\forall$ b: Buyer, s1, s2: Seller, cr1, cr2: ComplyWithRequirements, nbp: NegotiateBetterPrice, op1, op2: Offer&Profile, or: Order
(**Fulfilled**(cr1) $\wedge$ **Fulfilled**(cr2) $\wedge$ (cr1.dependee = s1) $\wedge$ (cr2.dependee = s2) $\wedge$ (cr1.depender = b) $\wedge$ (cr2.depender = b) $\wedge$ (op1.dependee = s1) $\wedge$ (op2.dependee = s2) $\wedge$ (nbp.betterprice = op1.price) $\wedge$ (nbp.betterprice < op2.price) $\rightarrow$ (or.depender = s1) $\wedge$ (or.pspec = op.pspec) $\wedge$ (or.price = op1.price) $\wedge$ **Fulfilled**(nbp))

If a seller can supply two different "Offer & Profile" that "Comply with Requirements", the alternative satisfying the "Fulfill Marketing Policies" softgoal is submitted to the buyer.
$\forall$ s: Seller, cr1, cr2:ComplyWithRequirements, fmp1, fmp2:FulfillMarketingPolicies, op: Offer&Profile
((**Fulfilled**(cr1) $\wedge$ **Fulfilled**(cr2) $\wedge$ **Fulfilled**(fmp1) $\wedge$ ($\neg$**Fulfilled**(fmp2)) $\wedge$ (fmp1.pspec $\neq$ fmp2.pspec) $\wedge$ (cr1.dependee = fmp1.**actor**) $\wedge$ (cr2.dependee = fmp2.**actor**) $\wedge$ (fmp1.actor=s) $\wedge$ (fmp2.actor=s) $\rightarrow$ (op.pspec = fmp1.pspec))

5.1.4 Process View

Figure 6 shows the standard and exceptional flows of activities between Sofaland and Beca. In particular interactions are organized according to the classical *matchmaking*, *negotiation*, *execution* and *post-settlement* phases of a market conversation [7]. Figure 6 shows how Sofaland discovers two potential suppliers on the market (i.e. Beca and Argo) and sends them *"Polyurethane Requirements"*. If the counterparts correctly receive polyurethane requirements, the negotiation process is started. Accordingly, the transition between Matchmaking (MM) and Negotiation (NEG) is triggered through the following ECA rule.

End(MM)[**Received**([1,7]$_{\text{days}}$, Sofaland, Beca, *"Polyur. Requirements"*) ∧
Received([1,7]$_{\text{days}}$, Sofaland, Argo, *"Polyur. Requirements"*)]|ξ

The potential sellers generate offers in parallel. If Sofaland receives at least an offer, a comparative evaluation is performed and finally, in our scenario, Beca is selected as the final polyurethane supplier. Note that if at the end of the *"Evaluate Offers of Polyurethane"* task (EOP), offers do not *"Comply with Polyurethane Requirements"*, new offers are required, thus implementing the typical bargaining interaction of mutual adjustment of negotiation. The transition that implements this bargaining process is labeled as follows:

End(EOP)[¬**Achieved**(Beca, *"Comply with Polyurethane Requirements"*) ∧
¬**Achieved**(Argo, *"Comply with Polyurethane Requirements"*)]|ξ

The negotiation process is typically time-bounded on 7 days, thus this value is specified as the maximum residence time of negotiation.

Once the agreement between Beca and Sofaland is reached, control is transferred to Sofaland's Purchase Office that formalizes the *"Polyurethane Order"* and then handles all supporting activities needed to terminate the purchase process. In the meantime, Beca schedules, produces and delivers the polyurethane to Sofaland. If the *"Polyurethane Order"* is not received by the *"Process Polyurethane Order"* task (PPO) within 30 days from the agreement, Beca first waits for 3 days from the deadline, then urges the submission of an order.

Begin(PPO)[¬**Received**([1,30]days, Beca, Sofaland, *"Polyurethane Order"*)]|
Sequence(**Wait-for**([1,3]days *"Polyurethane Order"*);
Urge([1,3]days, Sofaland, *Polyurethane Order*")

If Sofaland does not acknowledge the request, the market conversation reaches a pending state that requires a manual compensation. Finally, source inspections and then payment are performed. In particular, source inspections need some information from Beca about the lot of polyurethane supplied. If this information is not provided and the urge compensation fails, the transition towards the pending state is triggered. Moreover if source inspections on polyurethane discover a violation of the agreement, Sofaland requires Baca to relax the price of the supply.

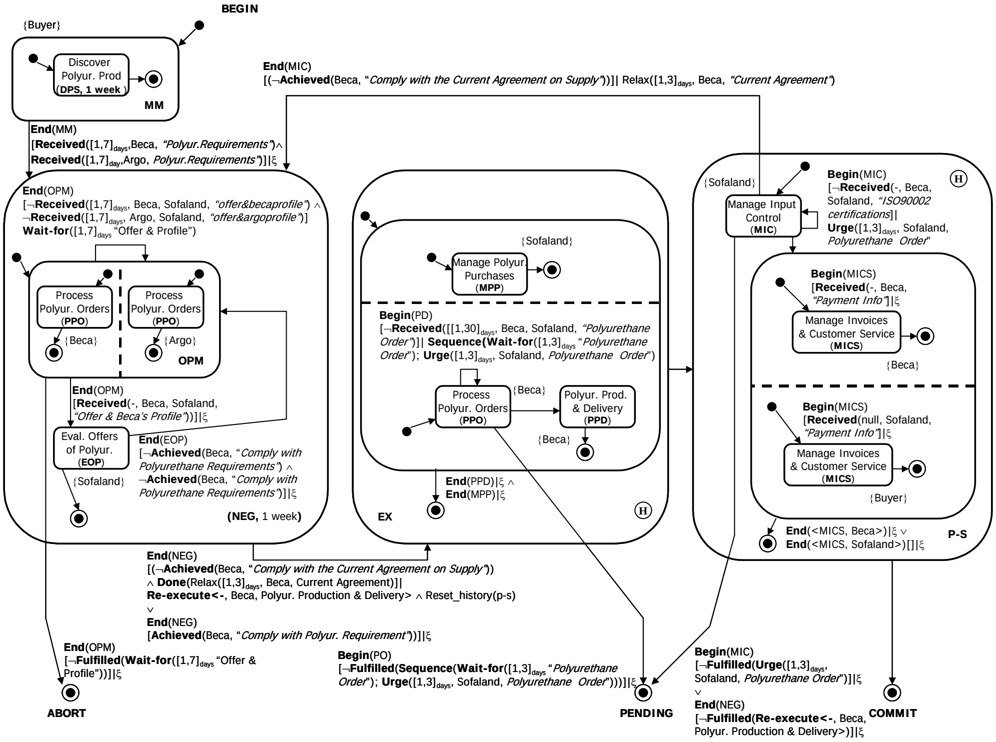

BEGIN
{Buyer}
Discover Polyur. Prod (DPS, 1 week)
MM
End(MIC)
[(¬Achieved(Beca, "Comply with the Current Agreement on Supply"))]| Relax([1,3]days, Beca, "Current Agreement")
End(MM)
[Received([1,7]days,Beca, "Polyur.Requirements")∧
Received([1,7]day,Argo, Polyur.Requirements")]|ξ
End(OPM)
[¬Received([1,7]days, Beca, Sofaland, "offer&becaprofile") ∧
¬Received([1,7]days, Argo, Sofaland, "offer&argoprofile")]
Wait-for([1,7]days "Offer & Profile")
Process Polyur. Orders (PPO)
{Beca}
Process Polyur. Orders (PPO)
{Argo}
OPM
End(OPM)
[Received(-, Beca, Sofaland, "Offer & Beca's Profile"))]|ξ
Eval. Offers of Polyur. (EOP)
End(EOP)
[¬Achieved(Beca, "Comply with Polyurethane Requirements") ∧
¬Achieved(Beca, "Comply with Polyurethane Requirements")]|ξ
{Sofaland}
(NEG, 1 week)
{Sofaland}
Manage Polyur. Purchases (MPP)
Begin(PD)
[¬Received([[1,30]days, Beca, Sofaland, "Polyurethane Order")]| Sequence(Wait-for([1,3]days "Polyurethane Order"); Urge([1,3]days, Sofaland, Polyurethane Order")
Process Polyur. Orders (PPO)
{Beca}
Polyur. Prod. & Delivery (PPD)
{Beca}
End(PPD)|ξ ∧
End(MPP)|ξ
EX
H
{Sofaland}
Manage Input Control (MIC)
Begin(MIC)
[¬Received(-, Beca, Sofaland, "ISO90002 certifications]|
Urge([1,3]days, Sofaland, Polyurethane Order"
H
Begin(MICS)
[Received(-, Beca, "Payment Info"]|ξ
Manage Invoices & Customer Service (MICS)
{Beca}
Begin(MICS)
[Received(null, Sofaland, "Payment Info"]|ξ
Manage Invoices & Customer Service (MICS)
{Buyer}
End(<MICS, Beca>)|ξ ∨
End(<MICS, Sofaland>)[]|ξ
P-S
End(NEG)
[(¬Achieved(Beca, "Comply with the Current Agreement on Supply"))
∧ Done(Relax([1,3]days, Beca, Current Agreement)]|
Re-execute<-, Beca, Polyur. Production & Delivery> ∧ Reset_history(p-s)
∨
End(NEG)
[Achieved(Beca, "Comply with Polyur. Requirement"))]|ξ
End(OPM)
[¬Fulfilled(Wait-for([1,7]days "Offer & Profile"))]|ξ
ABORT
Begin(PO)
[¬Fulfilled(Sequence(Wait-for([1,3]days "Polyurethane Order"); Urge([1,3]days, Sofaland, Polyurethane Order")))]|ξ
PENDING
Begin(MIC)
[¬Fulfilled(Urge([1,3]days, Sofaland, Polyurethane Order")]|ξ
∨
End(NEG)
[¬Fulfilled(Re-execute<-, Beca, Polyur. Production & Delivery>)]|ξ
COMMIT

Figure 6 – Specification of the standard and exceptional flows of activities between Sofaland and Beca.

Note that the use of history here is critical to model a correct behavior. If the counterparts agree on reducing the price of polyurethane, the business conversation must evolve into the *"payment"* state. By marking both execution and post-settlement with history, this behavior is easily modeled since the automaton enters the execution state and immediately leaves it since history points to the final sub-state within execution. Leaving execution, the automaton enters post-settlement but since the *"Manage Polyurethane Control"* state has been already visited, history points to the *"payment"* state as expected.

On the other hand, marking post-settlement with history could generate a wrong behavior since, after the re-execution of *"Production Polyurethane & Delivery"*, the automata skips quality control. The effect is that Sofaland receives the second lot, does not perform source inspections and pays the full price. This behavior is corrected by specifying a *"Reset_history"* action together with the *"re-execution from component production"* as follows (see also Figure 6).

End(NEG)

[(¬**Achieved**(Beca, *"Comply with the Current Agreement on Supply"*))

∧ **Done**(Relax([1,3]days, Beca, Current Agreement)]|**Re-execute<-**, Beca, Polyur. Production & Delivery> ∧ Reset_history(p-s)

Finally, if the re-execution from *"Production Polyurethane and Delivery"* fails, the business conversation reaches the pending state.

Figure 7 shows the *process view* of a market pattern as a generalization of the exemplification in Figure 6. Note how the use of universal and existential quantifiers within ECA (Event-Condition-Action) rules allows the substitution of a specific actor (i.e. Beca) with a general token belonging to a set modeling a role ($s \in \text{Seller}$). Moreover, the following rules typical of a market system complement the process view shown in Figure 7.

– **Refinement.** Tasks cannot be refined since the market pattern already provides the maximum view on internal business processes. Goals, softgoals and resources can be instead refined further. Moreover, the *"Requirements"* resource can embed information around price, product quality, delivery time, reliability of delivery and product specification.

– **Management of residual rights of control** (*on pending*)**.** When a market transaction reaches a pending state, the actor forcing pending has the responsibility to take control of the conversation and execute a recovery procedure.

– **Management of abort.** When a market conversation reaches an abort state, each actor has the responsibility for its recovery actions.

– **Management of time-outs.**
A time-out violation during negotiation brings the whole transaction into abort.
$\exists t\,((t>\text{time-out}) \wedge (l_q =<\text{NEG, time-out, ->}) \rightarrow l_{q+1} = <\text{abort, null, null}>))$
A time-out violation during matchmaking brings the whole transaction into abort.
$\exists t\,((t > \text{time-out}) \wedge (l_q =<\text{MM, time-out, ->}) \rightarrow l_{q+1} = <\text{abort, null, null}>))$

An example of properties satisfied by the *process view* in Figure 7 follows. Properties are formalized according to a notation that complies with FT and the process view provided in Sect. 4 [8]. Let *s* be an actor and *g* a goal in the social model, the following equivalence maps a goal condition of a process view into a FT formula:

Achieved(s, g) $\equiv_{def}$ (g.actor= s) $\wedge$ **Fulfilled**(g)

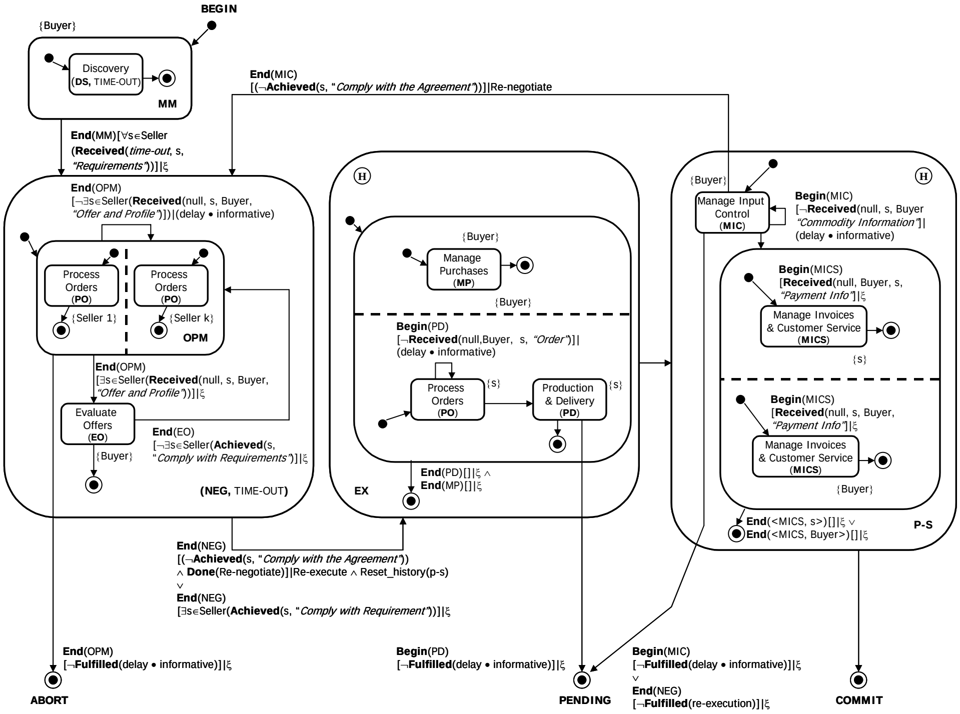
The "Comply with the Agreement" softgoal can be satisfied if the "Comply with Requirements" softgoal has been satisfied in the past.
$\forall$ s: Seller, ca:ComplyWithTheAgreement, cr: ComplyWithRequirements
((ca.actor=s) $\wedge$ **Fulfilled**(ca)) $\rightarrow$ ((cr.actor=s) $\wedge$ **O**(**Fulfilled**(cr))

Only the actor receiving the "Order" performs "Production & Delivery" and "Manage Invoices & Customer Services" tasks
$\forall$ s: Seller, pd: Production&Delivery, mics: ManageInvoices&CustomerService, o: Order
((pd.actor=s) $\wedge$ (mics.actor=s) $\wedge$ **Received**(-, s, o) $\leftrightarrow$ **Done**(pd) $\wedge$ **Done**(mics))

The "Manage Purchases" task and the "Production" task are executed in parallel
$\forall$ mp: ManagePurchases, p: Production&Delivery **And**(mp, pd)

A market transaction is always correct, i.e. each run terminates within either an abort or a commit or a pending state.

Let n be the number of transition needed to reach the final state q_n and τ our transition function, correctness is formalized as following.
$[(\tau\ (\text{begin}, -, -)=\text{MM}) \leftrightarrow \exists\ q_{n-1}\text{: Q }, q_n\text{: F } ((\tau(q_{n-1}, l_{n-1}, -)=<q_n, l_n>) \wedge$
$((l_n = <\text{commit, null, null}>) \vee\ (l_n = <\text{pending, null, null}>) \vee (l_n = <\text{abort, null, null}>)))]$


BEGIN
{Buyer}
Discovery (DS, TIME-OUT)
MM
End(MIC) [(¬Achieved(s, "Comply with the Agreement"))]|Re-negotiate
End(MM)[∀s∈Seller (Received(time-out, s, "Requirements"))]|ξ
End(OPM) [¬∃s∈Seller(Received(null, s, Buyer, "Offer and Profile")])|(delay • informative)
Process Orders (PO)
{Seller 1}
Process Orders (PO)
{Seller k}
OPM
End(OPM) [∃s∈Seller(Received(null, s, Buyer, "Offer and Profile"))]|ξ
Evaluate Offers (EO)
{Buyer}
End(EO) [¬∃s∈Seller(Achieved(s, "Comply with Requirements")]|ξ
(NEG, TIME-OUT)
H
{Buyer}
Manage Purchases (MP)
{Buyer}
Begin(PD) [¬Received(null,Buyer, s, "Order")]|(delay • informative)
Process Orders (PO)
{s}
Production & Delivery (PD)
{s}
EX
End(PD)[]|ξ ∧
End(MP)[]|ξ
H
{Buyer}
Manage Input Control (MIC)
Begin(MIC) [¬Received(null, s, Buyer "Commodity Information"]|(delay • informative)
Begin(MICS) [Received(null, Buyer, s, "Payment Info"]|ξ
Manage Invoices & Customer Service (MICS)
{s}
Begin(MICS) [Received(null, s, Buyer, "Payment Info"]|ξ
Manage Invoices & Customer Service (MICS)
{Buyer}
End(<MICS, s>)[]|ξ ∨
End(<MICS, Buyer>)[]|ξ
P-S
End(NEG) [(¬Achieved(s, "Comply with the Agreement")) ∧ Done(Re-negotiate)]|Re-execute ∧ Reset_history(p-s) ∨ End(NEG) [∃s∈Seller(Achieved(s, "Comply with Requirement"))]|ξ
End(OPM) [¬Fulfilled(delay • informative)]|ξ
ABORT
Begin(PD) [¬Fulfilled(delay • informative)]|ξ
PENDING
Begin(MIC) [¬Fulfilled(delay • informative)]|ξ ∨ End(NEG) [¬Fulfilled(re-execution)]|ξ
COMMIT

* State and transition labels are parameters of this pattern

Figure 7 – Process view of a *market pattern*

The partial satisfaction of all delegated goals is required to commit the business transaction.
$\exists q_n \in F\ (l_n = <\text{commit, null, null}> \rightarrow$
[∃s:Seller, cr: ca:ComplyWithTheAgreement, cr: ComplyWithRequirements
((cr.actor=s) ∧ **Fulfilled**(cr)) ∧ (((ca.actor=s) ∧ **Fulfilled**(ca)) ∨ **Fulfilled**(**Re-negotiate**(s, ca))]

5.2 Vertical Quasi-Integration

By cooperating according to a vertical quasi-integration, organizations can benefit the advantages of hierarchical coordination without dealing with the typical risks of ownership [3, pp. 253]. Vertical quasi-integration is an organizational structure that embeds operating relationships typical of ownership and preserves juridical independence among cooperating actors.

Organizations are vertical quasi-integrated when they coordinate each other through strategic dependencies negotiated within an environment where decisional power is not symmetrically distributed among the cooperating counterparts [3]. Accordingly, in a vertical quasi-integration, a seller strongly depends from an enterprisewide organization and it is typical required to comply with well defined assets. Sellers are therefore unable to organize cooperation with more potential buyers because of site, physical, human and time asset specificity [29].

5.2.1 Case Study

F.lli Cornacchia. F.lli Cornacchia is an organization supplying Sofaland with high-quality sofa backbones. This strategic relationship results into significant technological investments involving the backbones production process. F.lli Cornacchia guarantees low transformation costs, an high production capacity and an excellent quality of the final product. This innovation process is monitored by Sofaland which provided detailed specifications on physical and technological assets.

However, F.lli Cornacchia can decide how to organize supporting activities in order to improve the performance of the production process. Accordingly, F.lli Cornacchia handles procurement according to a *just-in-time* policy organized as follows: *orders* are received and panels necessary to assemble backbones are immediately required to a trusted third party. Panels are then temporally stored in a warehouse and the backbone production process is scheduled and executed. All supporting activities such as *Finance & Administration*, *Human Resource Management*, *Information System Management*, *Shipping Resource Management*, are internalized.

Sofaland S.r.l. (vertical quasi-integration). Sofa backbones must comply with a strict system of international norms. Therefore, Sofaland supervises F.lli Cornacchia’s primary activities by collecting QoS information at run-time and provides a set of strong specification on the overall production process. For example, in order to reduce production costs, Sofaland re-engineered F.lli Cornacchia’s business process to produce sofa backbones from panels of white fir thus reducing lead-time from 900-1200 to 100-150 minutes. However, Sofaland completely delegates the management of supporting activities to F.lli Cornacchia.

5.2.2 Intentional View

Sofaland’s board of directors decides to buy sofa backbones according to a vertical quasi-integration since the procurement of sofa backbones represent a strategic activity for the company. Indeed, the frequency of conversation per year is high, specific assets are required to produce backbones and the environment is uncertain (see Figure 2). Moreover, in order to minimize conversation costs, Sofaland have to control seller’s production at run-time thus reducing the period required to compensate failures. On the other hand, Sofaland guarantees to F.lli Cornacchia the saturation of it production. Figure 8 shows the intentional view of a vertical quasi-integration pattern.

5.2.3 Social View

Figure 9 shows the social view of the pattern complying with the Sofaland – F.lli Cornacchia scenario. Sofaland supervises F.lli Cornacchia’s primary activities while delegating the management of supporting activities. In exchange, Sofaland guarantees to F.lli Cornacchia the saturation of its production capacity by taking responsibility of the fulfillment of the *"Saturate Production Capacity"* goal.

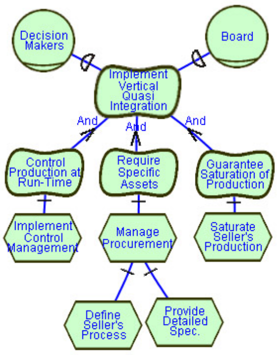
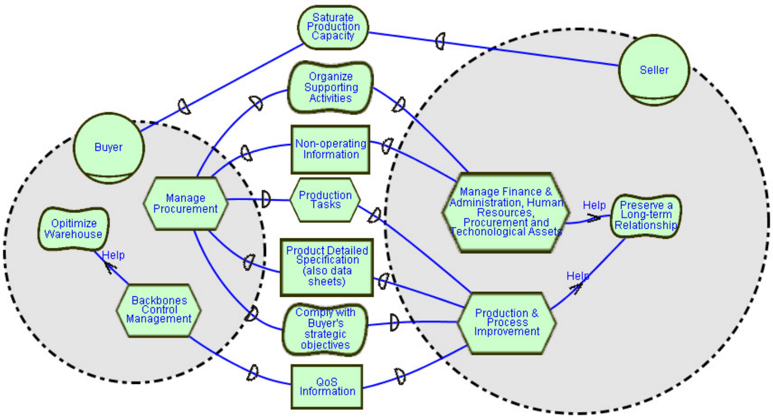

Decision Makers
Board
Implement Vertical Quasi Integration
And
And
And
Control Production at Run-Time
Require Specific Assets
Guarantee Saturation of Production
Implement Control Management
Manage Procurement
Saturate Seller's Production
Define Seller's Process
Provide Detailed Spec.

Figure 8 – Social view of a vertical quasi-integration pattern.


Saturate Production Capacity
Seller
Organize Supporting Activities
Buyer
Non-operating Information
Manage Procurement
Production Tasks
Manage Finance & Administration, Human Resources, Procurement and Techonological Assets
Help
Preserve a Long-term Relationship
Opitimize Warehouse
Help
Product Detailed Specification (also data sheets)
Help
Backbones Control Management
Comply with Buyer's strategic objectives
Production & Process Improvement
QoS Information

*Roles and intentional elements are the parameters of this pattern
Figure 9 – Social view of a vertical quasi-integration pattern.

In particular, the complete delegation of supporting activities is specified by delegating the *"Organize Supporting Activities"* goal to the *"Manage Finance & Administration, Human Resource, Procurement & Technological Assets"* task. Moreover, *"Non-operating information"* must be provided to F.lli Cornacchia in order to execute supporting activities efficiently.

On the other hand, F.lli Cornacchia is delegated the responsibility of providing backbones which *"Comply with Buyer's Strategic Objectives"*. Moreover, *"Backbones Production Tasks"* are delegated to F.lli Cornacchia's *"Produces Backbones & Manage Process Improvement"* task in order to guarantee that backbones production follows the set of strict specifications provided by Sofaland. The set FT properties complementing this pattern is detailed in [9].

We note that differently from a market pattern, we need to add properties formalizing that in a vertical quasi-integration conversation are not isolated from each other[1]. On the contrary, in a market, the commitment of a conversation does not depend on the previous interaction with the same supplier. An example follows.

The current fulfillment of the "Organize Supporting Activities" and "Comply with Buyer's Strategic Objectives" softgoals requires also their fulfillment in the past transactions

$\forall$ s: Seller, osa: OrganizeSupportingActivities, cbs: ComplyBuyersStrategicObjective

$[(\text{osa.actor}=s) \wedge \mathbf{Fulfilled}(\text{osa}) \wedge (\text{cbs.actor}=s) \wedge \mathbf{Fulfilled}(\text{cbs})] \rightarrow$

$[(\text{osa.actor}=s) \wedge \mathbf{O}(\mathbf{Fulfilled}(\text{osa}) \wedge (\text{cbs.actor}=s) \wedge \mathbf{O}(\mathbf{Fulfilled}(\text{cbs})]$

[1] We use the term *isolation* softly as a means to indicate that the achievement of goals in the current conversation also depends from the achievement of goals in the previous conversation.

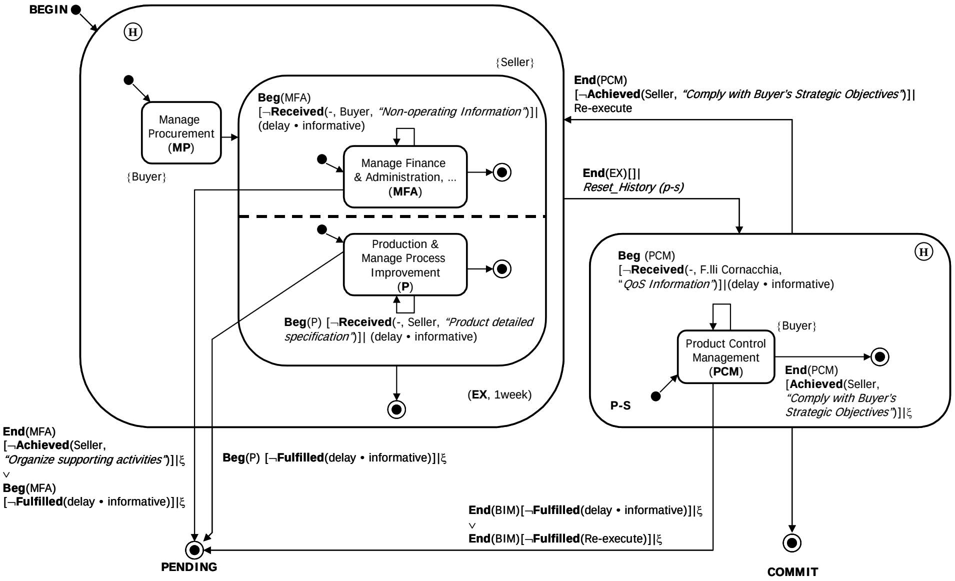
5.2.4 Process View

Figure 10 shows the process view of a quasi-vertical integration pattern modeling the Sofaland – F.lli Cornacchia scenario. In particular, we note that, differently from a market transaction, these interactions are organized into the *execution* and *post-settlement* phases only [7]. This behavior is quite intuitive since the Sofaland-F.lli Cornacchia scenario models a long-term cooperating relationship where F.lli Cornacchia supplies a complex and strategic product (i.e. sofa backbones). Sofaland does not need an automatic mechanism to support discovery and negotiation since cooperation is planned and negotiated *"face-to-face"* when requirements on either physical and technological assets or service characteristics are specified. When an agreement is reached and required assets are deployed, discovery and negotiation are no more executed and Sofaland and F.lli Cornacchia cooperate according to the model in Figure 10. Moreover, the following rules apply:

- **Refinement.** In a vertical quasi-integration a buyer has the power to retrieve control information from the seller's production process according to the desired granularity. The *"Production & Process Improvement"* task can be therefore brought down into subtasks, and, accordingly, subtasks can be further brought down until the desired granularity is reached. We note that differently from a market conversation, the pattern of a vertical quasi-integration specifies very high level tasks that can be brought down according to a desired granularity. On the contrary, a market pattern already specifies the lowest level of detail and task refinement is not allowed.
- **Management of residual rights of control.** When a vertical quasi-integrated conversation reaches a pending state, the buyer has the responsibility to take control of the conversation and execute a recovery procedure. Accordingly, in a vertical quasi-integration the buyer retains the decisional power and delegates but supervises the execution of operating activities.
- **Management of abort.** Abort is not allowed unless another supplier providing the same strategic service is available on the market. As a consequence, on pending, the buyer usually perform recovery actions in order to bring the conversation into commit.
- **Management of time-outs.** A time-out violation during execution brings the whole transaction into pending.

5.3 Comakership

Coordination of *dissimilar* activities cannot be effectively managed according to a vertical quasi-integration but, on the other hand, cannot be handled through market mechanisms. Therefore, when cooperating organizations need to coordinate *dissimilar* but *complementary* activities, direct supervision is overcome by a more flexible mechanism of mutual adjustment known as *comakership*. A comakership is an organizational style grounded on a strong peer-to-peer cooperation among business partners and provides an effective solution to the problem of coordinating dissimilar activities.


BEGIN
H
{Seller}
Manage Procurement (MP)
{Buyer}
Beg(MFA) [¬Received(-, Buyer, "Non-operating Information")]| (delay • informative)
Manage Finance & Administration, ... (MFA)
Production & Manage Process Improvement (P)
Beg(P) [¬Received(-, Seller, "Product detailed specification")]| (delay • informative)
(EX, 1week)
End(PCM) [¬Achieved(Seller, "Comply with Buyer's Strategic Objectives")]| Re-execute
End(EX)[]| Reset_History (p-s)
Beg (PCM) [¬Received(-, F.lli Cornacchia, "QoS Information")]|(delay • informative)
H
{Buyer}
Product Control Management (PCM)
End(PCM) [Achieved(Seller, "Comply with Buyer's Strategic Objectives")]|ξ
P-S
End(MFA) [¬Achieved(Seller, "Organize supporting activities")]|ξ ∨ Beg(MFA) [¬Fulfilled(delay • informative)]|ξ
Beg(P) [¬Fulfilled(delay • informative)]|ξ
End(BIM)[¬Fulfilled(delay • informative)]|ξ ∨ End(BIM)[¬Fulfilled(Re-execute)]|ξ
PENDING
COMMIT

* State and transition labels are parameters of this pattern
Figure 10 – Process view of a vertical quasi-integration pattern

5.3.1 Case Study

Matera Cinghie S.r.l. Matera Cinghie is a small craftsmen organization localized within the Italian district of Matera. It works on sofa backbones adding springs and straps necessary to assemble sofas with polyurethane and staffed pillows. In particular, this production process is organized as follows: production is scheduled according to either weekly or daily orders, then backbones are withdrawn from Sofaland's warehouse and directly delivered to Matera Cinghie's assembly line. Finally, after a strict quality control, the final output is directly delivered to Sofaland's assembly line without source inspections (*free pass policy typical of just-in-time*). Moreover, production settings differ each other according to the *comfort level* required by Sofaland.

Sofaland (comakership - operative integration). Adding springs and straps to sofa backbones is not a strategic activity for Sofaland. Moreover, this transformation process does not involve specific assets such as know-how, highly skilled human resources or advanced information systems. Nevertheless, to be consistent with its global strategy, Sofaland decided not to outsource the adding of springs and straps according to market coordination.

Sofaland selects Matera Cinghie as its business partner after a strict selection process (*i.e. vendor rating*) based on the evaluation of total cooperation costs, process capability, quality insurance, technological maturity and human resources. Moreover, Sofaland monitors these indicators over time comparing Matera Cinghie's performances with the average of the market. Differently, from a vertical quasi-integration, we note that performances are not elicited at run-time but at the end of the transformation process. Moreover, if performances results under the average, Sofaland helps its partner to regain a level of excellence by sharing knowledge about possible improvements at an operative and a strategic level.

The reduction of sofas production lead-time is obtained by delegating control activities to Matera Cinghie who takes complete responsibility for the fulfillment of the agreement. Moreover, the vendor rating process implemented during cooperation guarantees that deviations from the agreement are readily reported to buyer's procurement process. Finally, Sofaland pursues a *total quality* philosophy, i.e. a policy of continuous improvement of quality of product and product prices.

5.3.2 Intentional View

Sofaland's board of directors decides to outsource the adding of springs and straps according to a comakership since Matera Cinghie have to manage a strategic good such as sofa backbones. The implementation of a comakership is based on the adoption of a policy of continuous improvement, on the reduction of process lead time and on trust. Continuous improvement is achieved performing vendor rating, requiring information around the performance of the process, sharing knowledge (*know-how*) with the seller and requiring a continuous improvement of the product and the overall process. Lead-time is reduced through a stricter integration with the business partner and implementing weekly orders on the basis of current needs. Finally, Sofaland trust in the policy of continuous improvement performed by its partner and, accordingly, updates their contract according to market trends. Figure 11 shows intentional view of a comakership pattern.

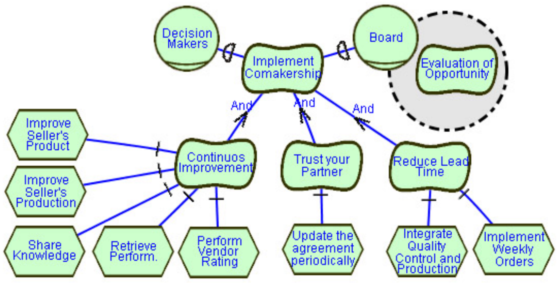


Figure 11 – Intentional view of a comakership pattern (operative integration).

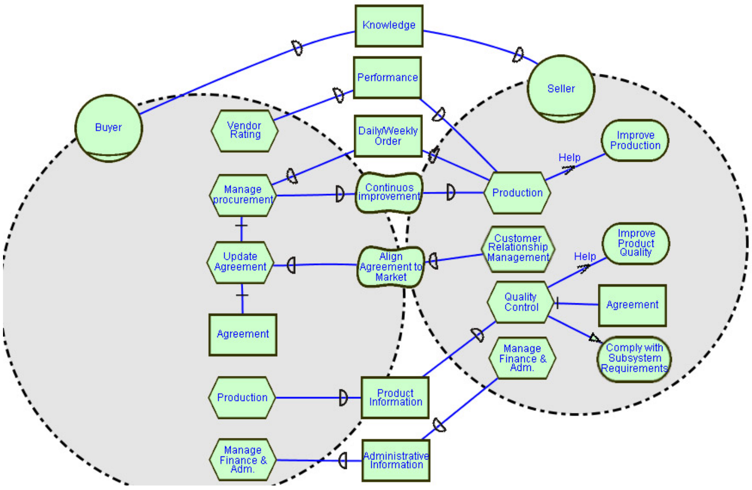
5.3.3 Social View

The Sofaland – Matera Cinghie scenario is a particular instance of the social view provided in Figure 12. As discussed Sect. 5.3.1, a distinctive characteristic of this relationship is the strong integration between "*Production*" and "*Quality Control*" tasks. Moreover, during the physical delivery of sofa backbones assembled with springs and straps, the "*Quality Control*" task communicates with Sofaland providing information useful to set its production process (e.g. "*Order Type & Comfort Level*" resource). We also note that the "*Quality Control*" task uses the "*Agreement*" resource to check that the transformation process complies with initial requirements.

From a strategic perspective, Sofaland delegates to Matera Cinghie the improvement of product quality and the downsizing of prices in order to pursue a continuous improvement of key performance indicators over time. Within Matera Cinghie, price downsizing is obtained by fulfilling the *"Improve Production"* goal since an improvement of error percentages reduces scraps and, as a consequence, total costs. Moreover, quality control activities help Matera Cinghie to improve its final output thus fulfilling the *"Improve Product Quality"* softgoal.

On the contrary, the Matera Cinghie's *"Customer Relationship Management"* task delegates the *"Update Agreement according to the Market"* goal to Sofaland. This delegation highlights the relationship of trust between buyer and supplier since Sofaland has the responsibility to monitor average prices on the market in order to pay Matera Cinghie within the average plus an half of control costs [25].

Finally, the *"Production"* task provides to Sofaland information around the performance of the overall production process. Accordingly, Sofaland's *"Vendor Rating"* task has enough information to compare the average trend of the market with its relationship with Matera Cinghie. As a consequence, if performances are under expectations, Sofaland shares with Matera Cinghie its know-how in order to improve its partnership.



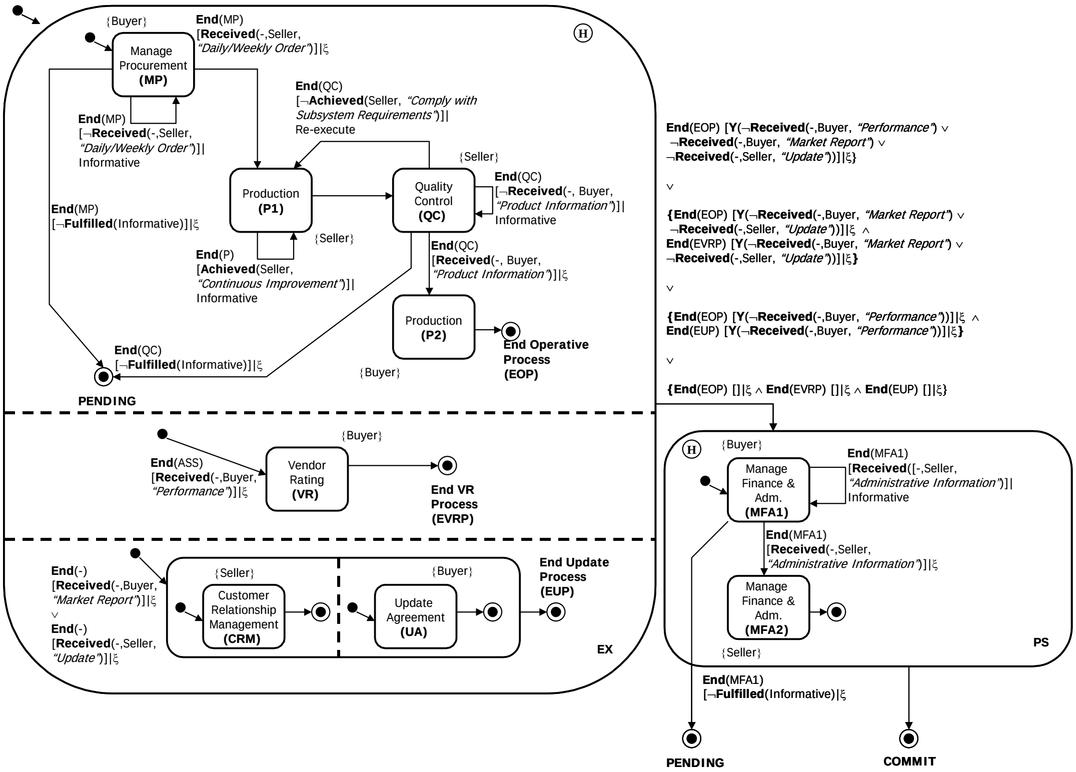
*Roles and intentional elements are the parameters of this pattern
Figure 12 – Social view of a comakership pattern (operative integration)

5.3.4 Process View

The Sofaland-Matera Cinghie scenario is a particular instance of the process view shown in Figure 13. In particular, interactions are organized into *execution* and *post-settlement* as well as vertical quasi-integration. However, comakership is based on trust and, as a consequence, embeds a weaker control over the seller. Accordingly, this coordination paradigm focuses on communicating the fulfillment of goals and softgoals as opposite to the communication of violations typical of market and vertical quasi-integration.

Execution involves three parallel standard flows: *an integrated operative flow*, *a vendor rating* flow and a flow of activities aimed at *updating prices* according to market trends. We note that both the vendor rating and price updating flows are enacted periodically.

The integrated operative flow begins when the *"Management Procurement"* task sends an order to Sofaland's *"Production"* task. Moreover, at the end of production, improvements around quality of production and price downsizing are notified to Sofaland. Then, the *"Quality Control"* task receives backbones with springs and straps and checks that specifications *"Comply with Requirements"*. If violations occur, Matera Cinghie re-executes the overall production task on the same backbones. We note that Sofaland is not aware of violations since control is performed by Matera Cinghie. Finally, since Sofaland adopts a free-pass policy, backbones with springs and straps are delivered directly to Sofaland's assembly line without any other control. During physical delivery, Matera Cinghie sends to Sofaland *"Process Information"* around sofas comfort level useful to set the assemble sofa process.



*State and transition labels are parameters of this pattern
Figure 13 – Process view of a comakership pattern (*operative integration*).

In the following a set of rules complementing this specification are outlined.

- **Refinement.** Tasks cannot be refined since the process view already provides the maximum view on the seller's internal business process. Goal, softgoal and resources can be instead refined further. Moreover, a *"Performance"* resource can embed information around total costs, process capability, quality insurance, technological maturity, human resources, management, improvements for the future.
- **Management of residual rights of control.** When a comakership conversation reaches a pending state, the buyer has the responsibility to take control of the conversation and to execute a recovery procedure. We note that even if the buyer retains the decisional power and delegates the execution of operating activities, it does not supervise its seller since control is weak and task refinement is not allowed.
- **Management of abort.** Abort is not allowed unless another supplier providing the same strategic service is available on the market. However, on pending, the buyer usually performs recovery actions in order to bring the transaction into commit.
- **Management of time-outs.** A time-out violation during execution brings the whole conversation into pending.

6. Conclusion and Future Work

This paper discusses the implications of different patterns for organizational cooperation on control and coordination mechanisms. The study of cooperation patterns is the first step towards improving the quality of cross-organizational conversations and overcoming the limits of traditional intra-organizational workflow design. The concept of control and coordination has been introduced and defined in compliance with previous literature and its implications on cross-organizational control and coordination has been analyzed as a means to formalize organizational structures. Three organizational patterns have been exemplified, discussing how control and coordination mechanisms can mix organizational hierarchy and market to build complex cooperation patterns suitable for real business cases. Each pattern have been provided using three different concurrent views: an *intentional view*, a *strategic view* and a *process view*. Each view addresses a specific set of concerns of interest to different stakeholders in the system and, as a consequence, it has its own particular notation, rationale and constraints. Moreover, the present work can provide a formal basis for orchestrating e-applications involving several organization, providing a coordination and control infrastructure consistent with cross-organizational structures.

Future research directions include the study of additional views supporting architectural design. In particular, we are currently formalizing a component view which discusses how a business conversation is described as composition of multiple e-services. Moreover, a deployment view modeling the service-oriented infrastructure necessary to support control and coordination according to different organizational styles is under development. Finally, a *"pattern factory"* supporting pattern identification, documentation, choice and composition will be developed in order to support modelers during the requirement engineering process.

7. References

[1] Antony, R.N. (1965). Planning and Control Systems: A Framework for Analysis, Boston (MA), Harvard University School Press.

[2] Arrow, K. J. (1974). The limits of organization. Norton & Company.

[3] Blois, K. (1972). Vertical Quasi Integration, Journal of Industrial Economics, 20

[4] Casati, F., Ceri, S., Paraboschi, S. and G. Pozzi (1999), Specification and Implementation of Exceptions in Workflow Management Systems. ACM Trans. Database Syst. 24(3): 405-451.

[5] Castro J., Kolp M., Mylopoulos J. (2002). Towards Requirement-Driven Information Systems Enginnering: The Tropos Project. Inf. Syst. 27(6): 365-389

[6] E. Colombo, C. Francalanci and B. Pernici (2002). Modeling Coordination and Control in Cross-Organizational Workflows, COOPIS'02, Irvine, USA.

[7] Colombo, E., Francalanci, C. and B. Pernici (2004), Modeling cooperation in virtual districts: a methodology for e-service design, International Journal of Cooperating Information Systems, Special Issue on Service Oriented Modeling, 13(4).

[8] Colombo, E. and C. Francalanci (2004), Modeling Business Conversation based on Non-functional Requirements, SoRE'04 co-located with RE'04, Kyoto, Japan.

[9] Colombo, E. and J. Mylopoulos (2004). Organizational Patterns for Virtual Enterprises, Technical Report, Politecnico di Milano

[10] Coase, R. H. (1960). The problem of social cost. Law and Economics, 3, 1-44.

[11] Darimont, R. and A. van Lamsweerde (1996) Formal Refinement Patterns for Goal-Driven Requirements Elaboration. SIGSOFT FSE, 179-190

[12] De Antonellis, V., Melchiori, M., Pernici, B. and P. Plebani, (2003). A methodology for e-Service substitutability in a Virtual District environment, Proceedings of CAiSE 03, 552-567.

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

[14] Fuxman, A., Liu, L., Mylopoulos, J., Roveri, M. and P. Traverso (2004) Specifying and analyzing early requirements in Tropos. Requir. Eng. 9(2): 132-150

[15] Galbraith, J. R. (1995). Designing Organizations, An executive briefing on strategy, structure and process, Jossey-Bass Publishers, San Francisco (CA).

[16] Gamma, E.. Helm, R., Johnson, R., and J. Vlissides (1995). Design Patterns: Elements of Reusable Object-oriented Software, Addison-Wesley.

[17] Harel, D. and Naamad A. (1996). The STATEMATE Semantics of Statecharts, ACM Trans. on Softw. Eng. and Meth., 5(4), 293-333.

[18] Kolp, M., Giorgini, P. and J. Mylopoulos (2003), Organizational Patterns for Early Requirements Analysis, Proceedings of CAiSE 2003.

[19] Konrad S. and B. Cheng (2002), Requirements Patterns for Embedded Systems, Proceeding of RE'02, Essen.

[20] Krutchen, P. (1995), The 4+1 View Model of Architecture. IEEE Software 12(6).

[21] Maiden, N. and B.P. Bright (1996), Recurrent Communication Patterns in Requirement Engineering Meetings, Proceedings of WET ICE '96

[22] Malone, T.W. and K., Crowston (1990). What is coordination theory and how can it help design cooperative work systems. In Proceedings of CSCW. ACM Press, New York, 357-370.

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

[24] Mecella, M. and B. Pernici (2001), Designing wrapper components for e-services in integrating heterogeneous systems. VLDB Journal 10(1): 2-15

[25] Merli, G. and M. Luoni (1997), Comakership, Isedi.

[26] Mintzberg H. (1992), Structure in fives: designing effective organizations, Englewood Cliffs, N.J., Prentice-Hall.

[27] Papazouglou M., Delis A., Bouguettaya A., Haghjoo M. (1997). Class Library for Workflow Environments and Applications. IEEE Trans. on Comp., 46 (6), 673-686.

[28] Teece, D. (1986). Profiting from Technological Innovation: Implication for Integration, Collaboration, Licensing and Public Policy, Research Policy, 15

[29] Williamson, O. E. (1996). The mechanisms of governance. Oxford University Press.

[30] Workflow Management Coalition, WfMC Handbook, 2001

[31] Yang, J., Mike P. Papazoglou: Web Component: A Substrate for Web Service Reuse and Composition. CAiSE 2002: 21-36

[32] Yu, E. and J. Mylopoulos (1996). Using goal, rules and methods to support reasoning in business process reengineering. Intelligent Systems and Accounting, Finance and Management, 5(1),1-13