

# On the Concurrency of Inter-Organizational Business Processes

Diogo R. Ferreira

Organizational Engineering Center, INESC and  
IST - Technical University of Lisbon  
Avenida Prof. Dr. Cavaco Silva, 2780-990 Porto Salvo, Portugal  
diogo.ferreira@ist.utl.pt

**Abstract.** Within organizations, workflow systems can automate business processes by centrally coordinating activity sequences. But outside their borders, organizations are autonomous entities that cannot be subject to centralized process control. Their internal processes are autonomously defined and controlled, and what they need is to synchronize those concurrent processes. Just like Petri nets are a valuable tool to model activity sequencing in local business processes,  $\pi$ -calculus becomes a useful tool to model concurrency in inter-organizational processes. After a review of the main developments in cross-organizational workflow management, this paper illustrates the use of  $\pi$ -calculus to model the interactions between business processes running concurrently in different organizations. These interactions range from invoking external services to more complex patterns such as contract negotiation and partner search and selection. The paper concludes with a case study that illustrates the application of the proposed approach in a more realistic business scenario.

## 1 Background

Research on the application of workflow management in inter-organizational environments has produced a wealth of interesting approaches on how to deal with the problem of coordinating processes that span multiple organizations. The available solutions for cross-organizational workflow management have been developed in recent years and they have followed a path towards increasing flexibility, from focusing on “low-level” issues of workflow systems interoperability to more flexible, “higher-level” architectures based on contracts and views. Putting these developments in perspective, several trends can be identified, including:

*Workflow interoperability mechanisms* - after the publication of the Workflow Reference Model [1], which described four models of interoperability between workflow systems, supporting cross-organizational workflows seemed to be a matter of selecting the most appropriate interoperability model for a given scenario. In [2], Anzböck and Dustdar describe the application of these models to medical imaging workflows that resemble cross-organizational workflows.

In [3], van der Aalst formalizes the case-transfer and extended case-transfer models using Petri nets, and compares them as approaches to partitioning cross-organizational workflows over multiple business partners.

*Federating heterogeneous workflow systems* - it was quickly found out that run-time interoperability during process execution required build-time interoperability during process definition as well. The problem of connecting workflow systems then turned into a problem of federating them [4], i.e., to devise architectures in which several different workflow systems appear as a single, integrated one. Different solutions emerged, such as [5], in which Lindert and Deiters propose an approach to defining cross-organizational workflows via the interconnection of “process fragments” specified by different parties, in an early effort to address the problem of autonomy of business partners in inter-organizational settings. Reichert et al [6] realized the same problem, but focused on adaptive features that require the use of a centralized workflow modeling facility.

*Agent- and service-based architectures* - the paradigm of software agents renewed the interest in supporting cross-organizational workflows, especially in connection with service-oriented architectures, by facilitating the integration of local and remote services. Blake [7] developed a middleware architecture based on service-invoking agents which are essentially controlled by a global workflow manager agent that enforced a centralized workflow policy. Kwak et al [8] developed a middleware architecture where the workflow system inside an organization can invoke services registered either in a local service interface repository (LSIR) or in a global service interface repository (GSIR). Stricker et al [9] went even further to propose a trader system that promotes the reuse of workflow data types between organizations and supports bidding of service offerings to submitted service requests. Yan and Wang [10] devised an architecture where local processes are exposed as services that can be invoked in cross-organizational workflows.

*Contract-based approaches* - it had already been realized that the autonomy of business partners means that cross-organizational workflows are subject to agreements, or contracts. The CrossFlow project [11] was a milestone in the development of contract-based approaches, since it developed a consistent framework for establishing contracts and configuring workflow systems based on those contracts. Other authors have since then developed similar approaches, such as [12], in which van den Heuvel and Weigand propose a contract specification language to define cross-organizational workflows. But in [13], Kafeza et al realize that cross-organizational workflow contracts should refer only to partial views of local workflows. This is due to several reasons, including the need to keep some information private, or the fact that an organization may establish contracts with several partners having different requirements.

*View-based models* - the work on contract-based approaches opened up a new way of looking towards cross-organizational workflows. One of the most significant developments is [14], in which van der Aalst and Weske describe their public-to-private approach, where “public” stands for the agreed-upon workflow

and “private” stands for the local workflows running at each end. The private workflow can be obtained via inheritance [15] from the public workflow. Chebbi et al [16] describe how to do the opposite, i.e., how to obtain the cooperative (public) workflow from the local workflow, given the set of publicly advertised activities. In [17], Chiu et al present an XML-based language for describing public workflow views.

Apart from a few exceptions, which include for example the work of Chebbi et al [16] just mentioned, the centralized control of cross-organizational workflows - whether during build-time, run-time, or both - has been a recurrent assumption in cross-organizational workflow management. [4], [5], [6], [7], [9] are some examples of contributions that rely on the ability to centrally coordinate a cross-organizational workflow. [8] and [10] are examples of authors who escaped that problem by basing their approaches on activity outsourcing, hence giving legitimate control of the process to the contractor. Still, even in recent developments, such as process mediation by means of web service choreographies [18], authors often resort to centralized process control in order to coordinate cross-organizational workflows.

But inter-organizational processes require a different focus. Whereas activities and their sequencing are the main issue in workflow management within an organization, inter-organizational environments are dominated by interactions and concurrency. In these scenarios, organizations search for potential business partners, engage in contract negotiations, and establish channels in order to perform sets of interactions. Rather than enforcing an activity sequence, the need is to synchronize business processes running concurrently in different organizations. The purpose of this paper is to draw attention to the concurrent nature of inter-organizational processes and to show how one can resort to different techniques in order to model sequencing in one case, and concurrency in another. In this context, the main contenders for workflow modeling - Petri nets and  $\pi$ -calculus - can actually be used together in order to understand those two features of inter-organizational processes.

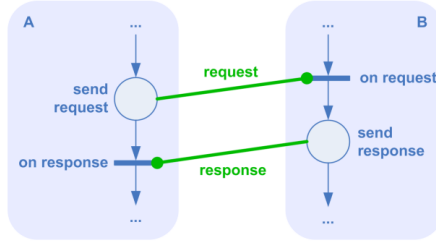
## 2 Modeling concurrency with $\pi$ -calculus

Since the publication of Smith and Fingar’s paper [19], there has been an ongoing debate about the value of  $\pi$ -calculus for workflow management. Most of the controversy has been set around the ability of  $\pi$ -calculus to model certain workflow patterns, defined using Petri nets [20]. Recent work from Puhlmann and Weske [21] suggests that it is indeed possible to describe workflow patterns using  $\pi$ -calculus, although that is far from producing a workflow revolution, as originally proposed in [19]. Meanwhile, there are already some contributions on the application of  $\pi$ -calculus to workflow modeling [22] and to modeling the interactions of different entities in an electronic market [23]. Additional developments will certainly follow.

Apart from the controversy,  $\pi$ -calculus can be extremely useful to model concurrent business processes. In fact,  $\pi$ -calculus was devised having concurrency

in mind, which makes it an obvious choice to represent synchronization points between concurrent processes. On the other hand, we will keep on using Petri nets to represent activity sequencing for the local processes running within organizations.

The basic elements that we will be using to model inter-organizational processes are shown in figure 1. In this figure,  $A$  and  $B$  denote two autonomous organizations. Both of them have internal processes, and they will be interacting with each other while carrying out those local processes. The local processes are described using Petri nets, where each place is associated with an action, and each transition with an event<sup>1</sup>. Each action stands for either a local task or an interaction with the external environment. In this case, only interaction tasks are shown. The interactions are represented using  $\pi$ -calculus links, the dot indicating the receiving end.



**Fig. 1.** Elements for modeling inter-organizational processes

For the sender, the interaction is an action that produces the outgoing message, so it is associated with a place. For the receiver, the interaction is an event that signals the arrival of that message, so it is associated with a transition. In the example shown in figure 1, after  $B$  receives the request from  $A$ , they swap roles; now it is  $B$  who produces an outgoing response, and  $A$  that receives the response as a transition-firing event. This example is illustrative of how an external service invocation - automated at both ends but not between them - could be modeled. In terms of  $\pi$ -calculus, this example could be written as:

$$A(link) = \overline{link} \langle request \rangle . link(response) \quad (1)$$

$$B(link) = link(request) . \overline{link} \langle response \rangle \quad (2)$$

Basically, these two expressions specify what  $A$  and  $B$  are doing concurrently.  $B$  is given a link and waits for a request to be received through that link, then

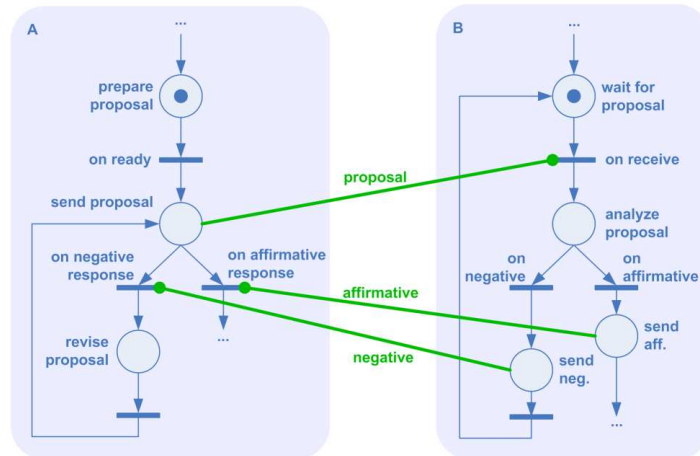
<sup>1</sup> At this point, it should be noted that the adopted approach follows [24], whereas some authors use transitions to represent workflow activities, as proposed in [25]. A discussion of these two approaches is beyond the scope of this article, but can be found in [26].

sends a response. *A* begins by sending the request through the given link, then waits for a response on the same link.

### 3 Modeling contract negotiation

In inter-organizational environments, where autonomous companies interact with each other, it is often difficult to automate the iterative processes that take place between them, as they develop business collaborations. Contract negotiation is one of such processes, since many interactions may be required until both parties reach an agreement. For these processes there is no standard activity sequence, hence the difficulty of workflow systems in modeling this behaviour. However, in terms of concurrency, it all comes down to the set of messages that organizations typically exchange with each other, and this is a well-known set of interactions that can be easily specified.

Let us assume that, while negotiating a contract, organization *A* prepares and sends a contract proposal to organization *B*. *B* considers the proposal and decides whether it should be accepted or not. If *B* does not accept it, then *A* will revise the proposal and send a new one. This behaviour will repeat itself until *A* and *B* reach an agreement. Figure 2 shows the processes running at both ends, as well as the interactions that take place between them. Basically, there is a cycle running at each end: *A* sends and revises proposals until it gets an affirmative response from *B*; on its turn, *B* receives, analyzes and replies to each proposal until an agreement is reached.



**Fig. 2.** Sequencing and concurrency in contract negotiation

In terms of concurrency, the above interactions could be described in  $\pi$ -calculus as:

$$A(\textit{link}) = \overline{\textit{link}} < \textit{proposal} > .(\textit{link}(\textit{yes}) + \textit{link}(\textit{no}).A(\textit{link})) \quad (3)$$

$$B(\textit{link}) = \textit{link}(\textit{proposal}).(\overline{\textit{link}} < \textit{yes} > + \overline{\textit{link}} < \textit{no} > .B(\textit{link})) \quad (4)$$

## 4 Using mobility to model partner search

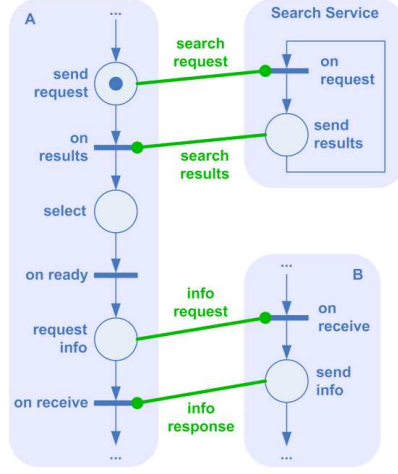
A distinctive feature of  $\pi$ -calculus when compared to earlier process algebras is the concept of *mobility*, which is intended to allow  $\pi$ -calculus to model processes with changing structure [27]. Basically, mobility involves the ability to send links through links. This mechanism turns out to be extremely useful to model partner search, since it provides a way to describe how organizations find information about other organizations at run-time.

Let us assume that, in an electronic market, participants will have access to a partner search service. This service may be implemented as a centralized repository (such as a UDDI registry) or as a fully decentralized service as in a peer-to-peer e-marketplace [28]. Regardless of how the search service is physically implemented, it can be represented as a provider of links to market participants. The search service receives requests, matches them against the products/services available in the market, and replies with one or more possible candidates.

Figure 3 illustrates the interactions between an organization *A* and the partner search service. The service is invoked during the execution of a local process at *A*, which aims at finding a suitable partner for a given business need. The search service replies with a (presumably non-empty) set of matching results, from which *A* will select an interesting candidate, which will be referred to as organization *B*. Then *A* interacts directly with *B* in order to obtain further information about that candidate and its product offers.

In order to formalize this behaviour using  $\pi$ -calculus, we will assume that the following services are available in the market [29]:

- *Trading Partner Search Service (TPSS)* - the partner search service just described.
- *Trading Partner Information Service (TPIS)* - a service link that allows market participants to exchange institutional or product information in order to support partner selection. In general, TPIS will be used to retrieve additional information about the possible candidates provided by TPSS.
- *Trading Partner Agreement Service (TPAS)* - a service link that allows market participants to negotiate contracts, as described in the previous section. In general, TPAS will be used when a candidate partner has already been selected. The communication channel required by TPAS is obtained via TPIS.
- *Trading Partner Execution Service (TPES)* - a one-to-one service link that allows market participants to perform the interactions established in a previously agreed contract. The communication channel required by TPES is obtained via TPAS.



**Fig. 3.** Modeling partner search with mobility (simplified)

The interactions shown in figure 3 can then be expressed using the following expressions, where the subscript is used to denote sub-processes belonging to the same entity:

$$PSS(tpss) = tpss(request).\overline{tpss} \langle result \rangle .PSS(tpss) \quad (5)$$

$$A_{SEARCH}(tpss) = \overline{tpss} \langle req \rangle .tpss(res).A_{SELECT}(res) \quad (6)$$

$$A_{SELECT}(candidates) = A_{INFO}(selection) \quad (7)$$

$$A_{INFO}(tpis) = \overline{tpis} \langle tpis\_request \rangle .tpis(tpis\_response) \quad (8)$$

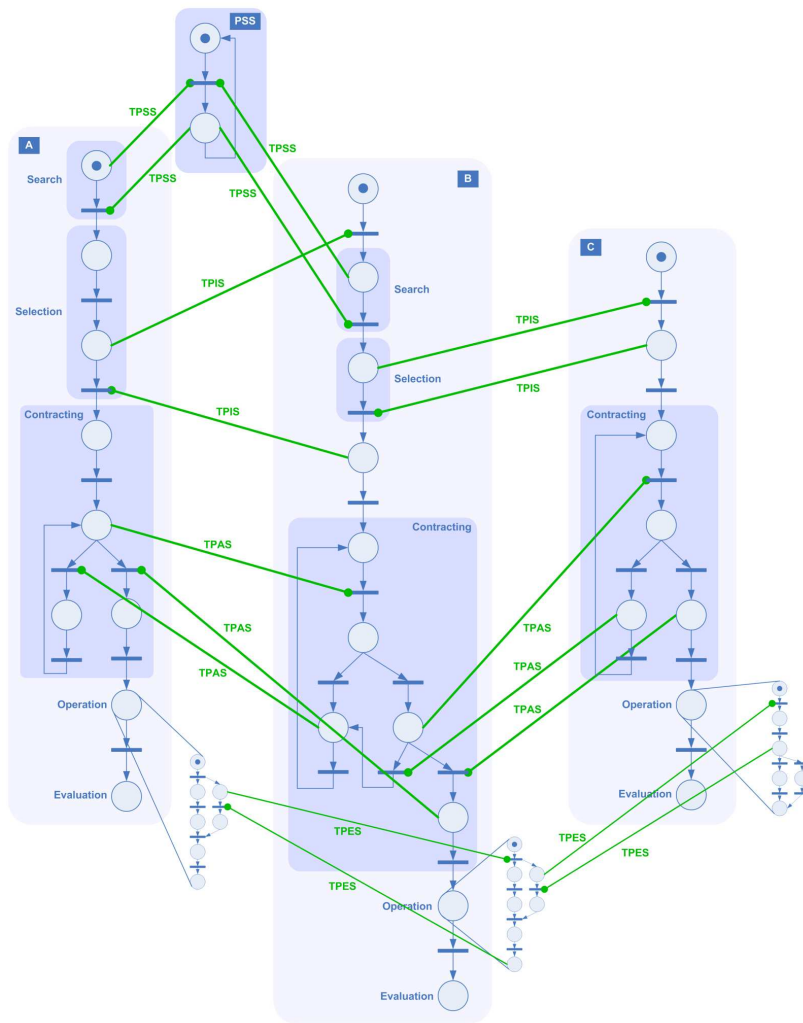
$$B_{INFO}(tpis) = tpis(tpis\_request).\overline{tpis} \langle tpis\_response \rangle \quad (9)$$

## 5 Case study: a semiconductor supply chain

In this section, we will illustrate how the proposed approach can be used to model a real-world business scenario. This scenario is basically equivalent to that presented in [30], and it involves three companies from the semiconductor industry. To avoid working with real names, we will refer to these companies simply as *A*, *B* and *C*. Organization *A* is a manufacturer of electronic subsystems for the automotive industry. Most of these components require application-specific integrated circuits (ASICs), which *A* orders from *B*. In order to produce these and other customized integrated circuits, company *B* needs silicon wafers, which are supplied by *C*. Silicon wafers are standard products, which *C* can supply from its own stock, while driving production for stock replenishment.

For the purpose of our scenario, we will assume that none of these companies know each other *a priori*, so that they will have to search for and develop

business collaborations with one another. First, company *A* will search for a supplier of ASICs. Having identified a list of potential suppliers, *A* will engage in conversations in order to select the best supplier, which eventually will be company *B*. Then *A* and *B* will sign a contract which specifies how the purchase will take place, from initial ordering to final payment. Company *B* will produce the ASICs and send them to *A*, so that *A* can proceed with its own manufacturing process. Finally, *A* will measure the performance of the supplier so that this information can be taken into account in future partner selections.



**Fig. 4.** Concurrent processes in a semiconductor supply chain (simplified)



While *A* develops its collaboration with *B*, *B* develops a collaboration with *C* according to a similar procedure. The timing of these events is such that when *A* searches for a supplier of ASICs, *B* searches for a supplier of wafers. And when *A* signs a contract with *B*, *B* signs a contract with *C* simultaneously. In this case, *B* plays both the service-requester and service-provider roles at the same time, on one side towards *C* and on the other side towards *A*, respectively.

During the operation phase, company *A* initiates production of the electronic system and at the same time sends the order for the ASICs to company *B*. Then *B* initiates its chip-manufacturing process, while at the same time ordering the silicon wafers. Supplier *C* provides the wafers immediately and replenishes its own stock if needed. The wafers arrive in the middle stages of *B*'s manufacturing process which, when completed, will provide the ASICs to company *A*.

## 6 Conclusion

Cross-organizational workflow management has been a long way from the run-time problem of supporting distributed workflow execution, up to the build-time problem of decoupling models of public collaborative workflows from those of private, local business processes. Still, it is clear that inter-organizational processes require solutions other than just mechanisms to enforce activity sequencing. The concurrent nature of inter-organizational processes calls for a solution that supports the interconnection of business processes running under the control of different organizations.

In this context,  $\pi$ -calculus becomes a valuable tool to model concurrency and synchronization in inter-organizational processes, just like Petri nets are useful to model activity sequencing in local business processes. In this paper, we have described mainly in a graphical way how these two formalisms can be combined in order to model inter-organizational systems. In the future, we will study the combination of these notations on formal grounds in order to determine the kind of properties that can be formally verified.

## References

1. D. Hollingsworth, "The Workflow Reference Model", Document Number TC00-1003, WfMC, 1995
2. R. Anzböck, S. Dustdar, "Interorganizational Workflow in the Medical Imaging Domain", ICEIS'03, Angers, France, April 23-26, pp.292-300, Kluwer, 2003
3. W. van der Aalst, "Process-oriented architectures for electronic commerce and interorganizational workflow", Information Systems, 24(9), December 1999
4. A. Geppert, M. Kradolfer, D. Tombros, "Federating heterogeneous workflow systems", Technical Report 05, University of Zurich, 1998
5. F. Lindert, W. Deiters, "Modelling Inter-Organizational Processes with Process Model Fragments", Informatik'99, Paderborn, Germany, October 1999
6. M. Reichert, T. Bauer, P. Dadam, "Enterprise-Wide and Cross-Enterprise Workflow Management: Challenges and Research Issues for Adaptive Workflows", Informatik'99, Paderborn, Germany, October 1999

7. M. Blake, "An Agent-Based Cross-Organizational Workflow Architecture in Support of Web Services", 11th WETICE, IEEE Comp. Soc. Press, June 2002
8. M. Kwak, D. Han, J. Shim, "A Framework Supporting Dynamic Workflow Interoperation and Enterprise Application Integration", 35th HICSS, IEEE, 2002
9. C. Stricker, S. Riboni, M. Kradolfer, J. Taylor, "Market-based Workflow Management for Supply Chains of Services", 33rd HICSS, IEEE, 2000
10. S.-B. Yan, F.-J. Wang, "CA-PLAN, an Inter-Organizational Workflow Model", FTDCS'04, IEEE Computer Society, 2004
11. P. Grefen, K. Aberer, Y. Hoffner, H. Ludwig, "CrossFlow: cross-organizational workflow management in dynamic virtual enterprises", *Comp. Sys. Sci. and Eng.*, 15(5):277-290, 2000
12. W. van den Heuvel, H. Weigand, "Cross-Organizational Workflow Integration with Contracts", *Business Obj. Comp. Workshop (OOPSLA2000)*, Springer, 2001
13. E. Kafeza, D. Chiu, I. Kafeza, "View-based Contracts in an e-Service Cross-Organizational Workflow Environment", *VLDB-TESS, Rome, Italy*, 2001
14. W. van der Aalst, M. Weske, "The P2P approach to Interorganizational Workflows", *CAiSE'01, LNCS 2068*, pp.140-156, Springer, 2001
15. W. van der Aalst, T. Basten, "Inheritance of Workflows: An Approach to Tackling Problems Related to Change", *Theo. Comp. Sci.*, 270(1-2):125-203, 2002
16. I. Chebbi, S. Dustdar, S. Tata, "The view-based approach to dynamic inter-organizational workflow cooperation", *Data and Knowledge Eng.*, 56(2), 2006
17. D. Chiu, S. Cheung, S. Till, K. Karlapalem, Q. Li, E. Kafeza, "Workflow View Driven Cross-Organizational Interoperability in a Web Service Environment", *Information Technology and Management*, 5(3-4), July-October 2004
18. E. Cimpian, A. Mocan, "WSMX Process Mediation Based on Choreographies", 1st Intl. Workshop on Web Service Choreography and Orchestration for Business Process Management", Nancy, France, September 2005
19. H. Smith, P. Fingar, "Workflow is just a Pi process", *BPTrends*, January 2004
20. W. van der Aalst, "Pi Calculus Versus Petri Nets: Let Us Eat Humble Pie Rather Than Further Inflate the Pi Hype", *BPTrends*, 3(5), pp.1-11, May 2005
21. F. Puhlmann, M. Weske, "Using the Pi-Calculus for Formalizing Workflow Patterns", *BPM 2005, LNCS 3649*, pp.153-168, Springer, 2005
22. D. Yang, S. Zhang, "Approach for workflow modeling using pi-calculus", *Journal of Zhejiang University Science*, 4(6):643-650, Nov-Dec 2003
23. J. Padget, J. Bradford, "A pi-calculus Model of a Spanish Fish Market - Preliminary Report", *AMET-98, LNCS 1571*, pp.166-188, 1998
24. D. Ferreira, J. J. Pinto Ferreira, "Developing a Reusable Workflow Engine", *Journal of Systems Architecture*, 50(6):309-324, June 2004
25. W. van der Aalst, "The Application of Petri Nets to Workflow Management", *The Journal of Circuits, Systems and Computers*, 8(1), pp.21-66, 1998
26. D. Ferreira, "Workflow Management Systems Supporting the Engineering of Business Networks", PhD Thesis, University of Porto, February 2004
27. R. Milner, J. Parrow, D. Walker, "A calculus of mobile processes, Part I", *Information and Computation*, 100(1):1-40, September 1992
28. D. Ferreira, J. J. Pinto Ferreira, "Essential Services for P2P e-Marketplaces", *CARS&FOF 2002*, Porto, Portugal, July 3-5, 2002
29. D. Ferreira, J. J. Pinto Ferreira, "Building an e-marketplace on a peer-to-peer infrastructure", *Intl. J. of Comp. Int. Manufact.*, 17(3):254-264, April-May 2004
30. D. Ferreira, J. J. Pinto Ferreira, "Towards a workflow-based integration architecture for business networking", *Business Process Management Journal*, 11(5):517-531, 2005