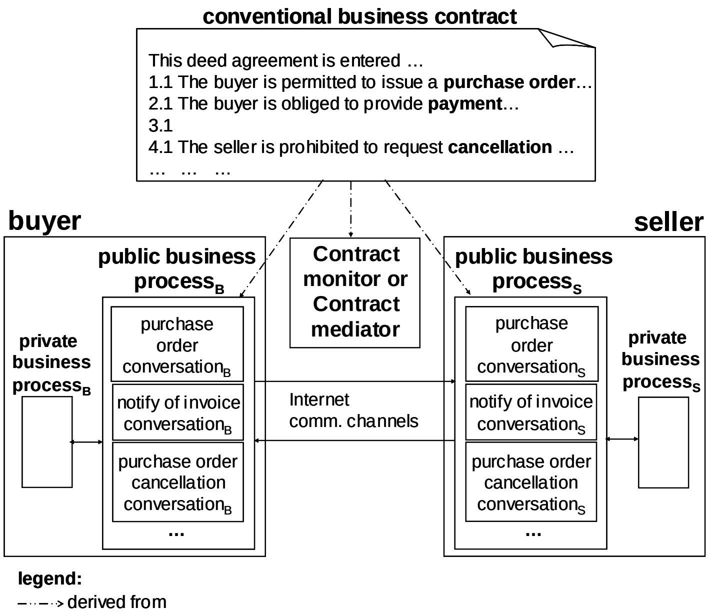
On Contract Compliant Business Process Coordination

Carlos Molina-Jimenez and Santosh Shrivastava
School of Computing Science
Newcastle University
Newcastle upon Tyne, NE1, 7RU

{Carlos.Molina, Santosh.Shrivastava}@ncl.ac.uk

1. Introduction

We assume that business-to-business (B2B) interactions between trading partners are being regulated by electronic contracts. The business scenario that motivated our research is depicted in the figure below. The buyer and the seller represent two autonomous organizations (trading partners) who have chosen to conduct business by means of exchanging messages over Internet communication channels. To interact, the buyer and the seller need to expose to each other, part of their business interfaces, in the figure, these interfaces are represented by the public business $process_B$ and public business $process_S$, respectively. To preserve their autonomy, the buyer and the seller conceal behind their public business processes those aspects of their business that they do not wish to disclose; in the figure, this is represented by the private business $process_B$ and private business $process_S$.



The B2B interactions between the buyer and the seller can be regarded as the execution of a global (cross-organizational) business process composed out of public business $process_B$ and public business $process_S$. Typically, a cross organizational business process can be constructed out of a set of small conversations (joint activities or dialogs) such as *issue a purchase order, send invoice, cancel purchaser order*, etc. We highlight some technical issues involved in ensuring that an execution of a cross-organizational business process is compliant with the corresponding electronic contract.

2. Contract Representation

The electronic representation of terms and conditions of the contract should be such that it can be utilized at design time by partners for validating their public processes and at run time for monitoring/mediating business interactions between trading partners, ensuring that these indeed correspond to meeting the rights and obligations that each interacting entity has promised to honour.

It is not well understood yet what formal language or languages should be used for contract description. To start with, such a language should be expressive enough to capture the permissions, obligations and prohibitions together with their time and order of execution constrains stipulated in the contract. Secondly, the language should produce a model that is simple enough to reason (perhaps by means of existing model-checking tools) about the logical consistency of the interaction; this requirement suggests that the formal notation should abstract away irrelevant implementation details. Finally and in direct conflict with the previous requirement, the language should produce a notation that can be mapped by the programmer into existing middleware technology and e-business standards such as RosettaNet, ebXML, CORBA, J2EE, Web services etc. From this observation it seems very unlikely that a single contract description language is enough; several equivalent descriptions of the contract might be needed with different level of abstractions. Considering our personal research experience we are planning to focus our efforts on the lowest two, perhaps three, levels [1]. The interest in implementation-independent contract notations is that they become declarative as their level of abstraction increases; this is a feature that we would like to explore as it is relevant in adaptable contractual business interactions. We believe that in practice, long term business interactions are likely to experience changes to adapt to new market conditions. On this basis, we speculate that the permissions, obligations and prohibitions stipulated in the original conventional contract should be split into two sets: those that stipulate the core business operations and are not expected to change and those that stipulate the complimentary business operations and are very likely to experience changes (within or outside the scope of the contract) during the contract life time. We suspect that the static set should be described in an implementation oriented notation and hard coded in an imperative language; on the other hand, the changeable set should be described in a declarative notation, so that it can be read, understood and changed by non-technical people such as business managers, as needed at run-time. In the literature, this approach is known as policy-driven or rule-driven and is a promising research avenue that needs exploring.

3. Process coordination

Naturally, business process executions at each partner must be coordinated at run-time to ensure that partners are performing mutually consistent actions. Distributed coordination mechanisms are attractive in B2B settings where all the partners are autonomous entities. A primitive B2B conversation typically involves exchange of one or more electronic business documents (e.g., a purchase order, shipping notification, invoice notification, etc.) and has various QoS constraints (timing, security, message validation, etc.). RosettaNet Partner Interface Processes (PIPs) are a good example of such conversations. The QoS constraints are expected to be met despite encountering software and hardware related problems (e.g., clock skews, unpredictable transmission delays, message loss, corrupted messages, node crashes, etc.). A failure to deliver a valid message within its time constraint could cause mutually conflicting views of an interaction (one party regarding it as timely whilst the other party regarding it as untimely). A conflict can also arise if a sent message is delivered but not taken up for processing due to some message validity condition not being met at the receiver (the sender assumes that the message is being processed whereas the receiver has rejected it).

In a loosely coupled system, it could take a long time before such inconsistencies are detected. Subsequent recovery actions - frequently requiring compensation - may turn out to be quite costly. It is best to deal with this problem at the source rather than at business process level by ensuring that interacting parties get a mutually consistent view on how a given interaction completed. The sender needs a timely assurance that the sent document will be processed by the receiver, and the receiver needs the assurance that if it accepts the document for processing, the sender is informed of the acceptance in a timely manner; in all other cases, the interaction returns failure exceptions to both the parties. Whereas existing middleware solutions have concentrated on providing generic client-centric communication primitives (such as RPCs), in the world of loosely coupled peer-to peer entities, we speculate that we need messaging abstractions with bi-lateral (multi-lateral) consistency guarantees. This is a challenging problem to solve, since we need to provide consistency without unduly compromising loose coupling. Encapsulating a business conversation within a single atomic transaction would not be considered practical as it would introduce tight coupling. Instead we need some form of a state synchronisation protocol that can be executed to reach an agreed outcome. Some ideas on how this could be achieved are presented in [2,3] where distributed as well as centralised synchronisation approaches are discussed.

We believe that the issue here is all about the execution of business processes with state notification to guarantee the observance of safety properties. The problem has practical relevance and can be reduced to the task of finding a set of synchronisation constraints that guarantee that a given list of safety properties in never violated. A more challenging problem is to present the business partners with a minimal set (which is not necessarily unique) of synchronisation constraints that satisfy the requirements. The problem can be also presented as follows: given the specification of a business process whose execution results in the violation of one or more safety properties, find the minimal synchronisation constraints to be introduced into the specification to remedy the problem.

Acknowledgements

This work is funded in part by UK Engineering and Physical Sciences Research Council (EPSRC), Platform Grant No. EP/D037743/1, “Networked Computing in Inter-organisation Settings”

References

[1] Carlos Molina-Jimenez, Santosh Shrivastava and John Warne, “A Method for Specifying Contract Mediated Interactions”, In Proc. of the IEEE International Enterprise Distributed Object Computing Conference (EDOC 2005), Sep. 2005, Enschede, The Netherlands, pp. 106-115.

[2] Carlos Molina-Jimenez, Santosh Shrivastava, “Maintaining Consistency Between Loosely Coupled Services in the Presence of Timing Constraints and Validation Errors”, Proc. the 4th IEEE European Conf. on Web Services (ECOWS’06), Dec. 2006, Zurich, pp. 148-157.

[3] Carlos Molina-Jimenez, Santosh Shrivastava and Nick Cook, “Implementing Business Conversations with Consistency Guarantees Using Message-oriented Middleware”, To appear in Proc. of the IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), Oct. 2007, Annapolis, Maryland U.S.