Business Process Interoperability: Extended Abstract

Amit K. Chopra
North Carolina State University
Department of Computer Science
akchopra@ncsu.edu

1. INTRODUCTION

A primary challenge for Computer Science is devising suitable abstractions for a given application domain so that solutions may be assembled and analyzed at the level of the domain, that is, by using primitives of the domain. For example, in personal finance, *deposit* and *withdraw* are primitives, not *add* and *subtract*. Our interest lies in the domain of business processes. In the most general sense, a business process may be thought of as multiple organizations interacting with each other to obtain their respective business goals. Supporting autonomy and compliance checking are key for business processes: each organization in a process is free to act as it pleases within the constraints of the contracts it enters into with others; violation of a contract typically means additional penalties.

What are the primitives for the business process domain? We claim: *agents* and *commitments*. Each organization is modeled as an agent and its interactions with others are given meaning in terms of how they affects its commitments. Commitments represent the business semantics of business processes. 'Commitment' is not just one primitive; it implies a host of additional primitives which correspond to operations on commitments: create, discharge, delegate and so on. In essence, we model a business process as the evolution of the commitments of agents. Our research agenda focuses on devising computational representations of commitments and their applications in engineering business processes.

Commitments provide a basis for reasoning about interoperation: do the commitments of the agents align in a mutually satisfactory manner? Commitments provide a basis for reasoning about protocols: why is paying cash often as satisfactory as paying via credit card? Commitments provide a basis for reasoning about enactment: why is paying $10 in two installments of $5 often as acceptable as a one-shot payment? Further, commitments provide a basis for reuse and composition: why can one substitute one payment protocol for another in a purchase protocol?

Existing approaches for business processes rely on data and control flow abstractions that were developed for structured programming but are ill-suited for business processes for the simple reason that these abstractions do not rise to the level of the domain. It is not that existing approaches don't value the role of commitments; it is just that they don't recognize the centrality of commitments.

In the following, we highlight one major direction of our research: a notion of business interoperability based on commitment alignment.

Cite as: Extended Thesis Abstract for Doctoral Mentoring Program, A. K. Chopra, Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 1730-1731.

2. BUSINESS INTEROPERABILITY

Interoperability is a matter of *manifest agreement*. In other words, the interoperability of two or more principals means not only that there is an agreement among the principals but also that they can act according to the agreement. An *agent* is a computational representation of a "real" business principal, which is the locus of autonomy. Agents interact with each other and their environment. We restrict attention to arms-length interactions in the form of *communications* among agents. These may be naturally realized in the computational infrastructure through messaging, and we refer to the elements of communication as messages. As agents interact, they enter into commitments with one another. We propose a commitment-based theory of interoperability of agents, the premise being that interoperability concerns the ability of agents to enter into and maintain well-aligned commitments to each other.

The theory is inspired from the Parnas' insight that in software architectures connectors should be treated not as control or data flow constructs but as *assumptions* made by each component about the others [5]. In our case, commitments represent the assumptions of agents. Commitments represent an essential level at which to assess and establish interoperability because they yield a notion of compliance eminently suitable for open settings: the principals may act as they please provided it is in accordance with their commitments. Arguably, much of the subsequent work on software architecture and interoperability regressed from Parnas' insight: it has primarily considered connectors and concomitant assumptions at the level of flow, e.g., dealing with message order and occurrence. Our theory contrasts with the prevalent approach wherein interoperability is understood primarily in terms of low-level criteria. Such low-level criteria are largely orthogonal to considerations of business semantics: specifically, what matters at the business level is what commitments exist, not whether a commitment was created or manipulated via a procedure call or a message, and whether a rigid message order was followed (unless the message order has a bearing on the semantics).

Just like checking for ASCII-level interoperability does not obviate the need for checking the interfaces of components for compatibility and vice versa, checking for commitment-level interoperability does not obviate the need for checking messaging-level interoperability of agents and vice versa—they are orthogonal. Commitments represent information about the business semantics of interaction, thus requiring a new formalization of interoperability that takes them into account.

2.1 Commitment-Level Interoperability

With commitments, we address business level interoperability. Commitments are directed from one agent (the *debtor*) to another (the *creditor*), and arise within a particular organizational context.

For the purposes of this paper, a commitment can be thought of as a directed obligation that is reified. When the condition of the commitment is met, the commitment is said to be discharged. In addition, a commitment may be operated upon, for example, by being delegated to a new debtor or assigned to a new creditor.

As stated above, interoperability is concerned with the ability of agents to enter into and maintain well aligned commitments with each other. Technically, this translates into the condition that if an agent's state models a commitment in which the agent is the creditor, then the debtor's state must also model the commitment. For example, let's say a merchant takes a quote message to mean no commitment towards the customer, but the customer takes it to mean that the merchant commits to sending goods if the customer pays first. This is a commitment misalignment: on receiving the message, the customer's state models a commitment in which it is the creditor and the merchant the debtor, but the merchant's state does not reflect this commitment. They are thus noninteroperable.

The reverse condition—if an agent's state models a commitment in which the agent is the debtor, then the creditor's state must also model the commitment—is of no relevance of interoperability. An agent may adopt commitments towards other agents; however, if other agents do not expect it, those commitments are just *gratuitous*.

The definition of interoperability is stated in terms of the observations of each agent and their corresponding states, an agent's observations being the messages it sends and receives. We model communication between agents as being asynchronous and make only fundamental and reasonable assumptions about it: the send of a message precedes its receipt, and message order is preserved.

2.2 Constitutive, Regulative Interoperability

The definition of commitment-level interoperability supports checking two kinds of specifications, constitutive and regulative, which we term as checking for constitutive interoperability and regulative interoperability respectively. We draw this distinction based on Searle's distinction of constitutive and regulative rules [6].

A constitutive rule, in our framework, specifies the meaning of a message. For example, the meaning of a quote message may be specified as a commitment that if the customer pays the quoted price, then the merchant will deliver the goods. Interoperability problems arise when in a customer's constitutive specification, quote means a commitment from the merchant, whereas in the merchant's constitutive specification, quote does not entail any commitment.

A regulative rule, on the other hand, either forces the occurrence of a message or constrains its occurrence, thus *regulating* the flow of messages. For example, a merchant may constrain the quote message to happen only in response to an rfq from the customer. An agent's regulative specification reflects its policies. If agents are constitutively interoperable, then no regulative rules can make them noninteroperable. However, if agents are constitutively noninteroperable, then it might be possible to frame regulative rules that make them interoperable. Technically, adding regulative rules amounts to refinement.

Constitutive interoperability may be thought of as being at the level of interfaces of agents whereas regulative interoperability also takes into account the agents' internal policies. Not agreeing constitutively reflects a deeper problem for agents—a problem at the level of meaning, albeit one which can be tided over by adjusting their regulative specification.

As in earlier work, we use $C+$ for studying interoperability [4]. Both constitutive and regulative rules are specified in $C+$. The decision procedures for determining interoperability proceed by program analysis of $C+$ specifications. In contrast to earlier work which applied $C+$ from the global perspective of a protocol, this work applies $C+$ from the perspective of an individual participant.

3. SUMMARY

Researchers in software components have long addressed the problem of component interoperability. They have approached this problem from the point of view of coordination: the definitions of interoperability are couched in terms of process-algebraic notions of liveness, fairness, choice, and deadlock-freedom of the components [1, 3, 7]. Such formalizations are no doubt relevant *and* essential; however, they do not capture the business semantics of business processes. Our commitment-based approach addresses this shortcoming. It abstracts away from the process-algebraic notions of interoperability, and makes commitment alignment the sole criterion. Our vision is that designers first specify agents in terms of commitments, check for commitment alignment, and then successively refine the specifications in a model-driven manner so as to obtain implementations that also meet the process-algebraic notions of interoperability.

Our expected contributions in this dissertation are threefold:

- A high-level definition of interoperability that takes into account the business meanings of communication.
- Distinction between kinds of interoperability—constitutive and regulative—and decisions procedures for each.
- A decision procedure for determining conformance. Conformance goes hand in hand with interoperability: replacing an agent specification with a conformant one preserves interoperability.

Preliminary results have appeared in [2]. Much works remains to be done. In particular, decision procedures for regulative interoperability and conformance are yet to be devised. In addition, multiparty scenarios need to be adequately addressed.

4. REFERENCES

[1] M. Baldoni, C. Baroglio, A. Martelli, and V. Patti. A priori conformance verification for guaranteeing interoperability in open environments. In *Proceedings of the 4th International Conference on Service-Oriented Computing*, pages 339–351, December 2006.

[2] A. K. Chopra and M. P. Singh. Constitutive interoperability. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, 2008. To appear.

[3] C. Fournet, C. A. R. Hoare, S. K. Rajamani, and J. Rehof. Stuck-free conformance. In *Proceedings of the 16th International Conference on Computer Aided Verification*, pages 242–254, July 2004.

[4] E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, and H. Turner. Nonmonotonic causal theories. *Artificial Intelligence*, 153(1-2):49–104, 2004.

[5] D. L. Parnas. Information distribution aspects of design methodology. In *Proceedings of the International Federation for Information Processing Congress*, volume TA-3, pages 26–30, Amsterdam, 1971. North Holland.

[6] J. R. Searle. *The Construction of Social Reality*. Free Press, New York, 1995.

[7] D. M. Yellin and R. E. Strom. Protocol specifications and component adaptors. *ACM Transactions on Programming Languages and Systems*, 19(2):292–333, 1997.