The Evolution of Interoperability

Amit K. Chopra[1] and Munindar P. Singh[2]

[1] University of Trento, Italy
chopra@disi.unitn.it
[2] North Carolina State University, USA
singh@ncsu.edu

1 History

This note is a retrospective review of our 2006 paper [1] on the properties of protocols, especially interoperability. A bit of history is in order. By 2006, the importance of a social semantics for protocols was well-established in the multiagent systems community. Further, commitments had emerged as the preeminent abstraction for capturing the semantics. The big advantage was that specifying the meaning of protocol messages in terms of the commitments among agents enabled the agents to act flexibly. Informally, the notion of flexibility derives from reasoning about the legal executions from a global perspective: if the set of legal executions of a protocol is a subset of those of another, then the latter is more flexible. For example, all other things being equal, a protocol that enables merchants and customers to exchange goods and payment in any order is more flexible than one that only supports payment before goods. Specifying protocols in terms of commitments promotes flexibility because compliance with a protocol amounts to fulfilling one's commitments. This in principle frees a protocol designer from the necessity of specifying the order of messages.

Flexibility is great! It ties in well with qualities that are commonly ascribed to agents—proactivity, opportunism, intelligent exception-handling, and so on. Flexibility is good from the business perspective. The greater the flexibility one can act with the greater are the opportunities for engaging others in business. For example, by adopting a protocol that enables payment and goods to be exchanged in any mutual order, a merchant can also engage customers who are unwilling to make their payments before the delivery of goods.

2 Distributed Enactment

Let us revisit the above assumption that increased flexibility offers expanded possibilities for engaging others. Note that protocols are enacted by agents in distributed settings. In such settings, it is difficult to ensure that the agents operate in lockstep with one another. Specifically, the agents may send and receive messages as they please without being made to block for another agent. This phenomenon is commonly referred to as *asynchrony*. About the only constraint we can rely upon is that the receipt of a message is causally later than its sending. Asynchrony makes interoperation challenging. And without interoperability, there can be no meaningful engagement. As a result, increased flexibility might not improve the opportunities for engagement in practice.

For concreteness, let us assume reliable, noncreative, order-preserving, point-to-point messaging. Even so, unexpected things could happen during enactment. Consider two agents who have adopted roles in some protocol. Even in acting according to the protocol, their messages to each other could cross in transit. For example, consider that a merchant's cancellation of an offer it had made earlier to some customer crosses in transit the customer's payment for the offered items. We naturally ask: what is the state of this interaction? Should the customer's payment entitle it to the items offered or should the payment be considered as having being made too late? Answering this question is important to enabling meaningful engagement. In scenarios involving more than two agents, the problem is exacerbated by the fact that an agent would in general lack knowledge of the messages exchanged by agents other than itself.

Perhaps the most commonly adopted solution today is to assume synchronous communication, which means that when an agent sends a message, it waits to take its next step until it receives an acknowledgment for the message it has sent. Synchronous communication is ill-suited to distributed settings since it creates delays corresponding to message roundtrips, and effectively makes one party dependent upon progress by another. Another common solution is to specify protocols so that if agents were to follow them, then the scenario of their messages crossing would not even arise. This is achieved quite simply by specifying protocols in terms of a progression of interaction state and ensuring that in each state, only one agent can send a message (the recipient would eventually block pending receipt of the message). In essence, the agents would enact protocols in close to lockstep synchrony.

Returning to our example, the problem scenario of the cancellation and the payment crossing could be handled as follows. The merchant would not be allowed to cancel offers. If it could, it could do so only in some limited period of time (a *make up your mind* period) before the sending of payment is enabled. Some protocols may be more flexible and allow cancellations at any time. However, in that case, when payment and cancellation cross, the situation for all practical purposes must be handled offline from the system that the protocol represents. (Assigning priorities to the agents can help decide which of two or more conflicting messages "wins". However, such priorities alone cannot synchronize the agents because the agents may since have progressed further.) So, for all practical purposes, it would seem we are stuck with lockstep synchrony in protocols. That does not leave the agents much flexibility. Much research in distributed systems arises from relaxing synchronous enactment in one way or another.

It was the thinking leading up to the paper being reprinted that made the tension between flexibility and interoperability clear to us. In the paper, we defined the notion of conformance with protocols based on commitments. This notion afforded designers much flexibility in building agent implementations. We defined interoperability following the more traditional manner of distributed systems research (based on the absence of *deadlocks*). We noted that a conformant agent may be noninteroperable with other agents. (In subsequent work not dealing with commitments [2], we revised conformance to be an interoperability preserving relation.) Winikoff [3] also noted the challenge of interoperability with commitment protocols. His assumption though of what amounts essentially to a monotonically growing knowledge base, rules out the possibility of even the discharge of a commitment.

Recall that the motivation behind commitment protocols was as much flexibility in enactment as demands the notion of an agent's autonomy. *An agent can do or communicate anything anytime; it is compliant as long as it fulfills its commitments.* However, without reconciling flexibility with interoperability, none of the flexibility could be realized in practice. Since 2006, reconciling them has become an important research direction for us.

3 Commitment-Level Interoperability

We have since made substantial progress on this issue. The key for us was to think in terms of interoperability conceptually. Interoperability is about the assumptions that agents make of each other. When the assumptions are compatible, the agents are interoperable. Traditionally, the assumptions have taken an operational form: they specify the order in which each agent expects to observe messages. A protocol is a global specification of assumptions: we can derive from it the assumptions relevant to any single participant. Traditionally, protocols for distributed systems have been specified operationally (for instance, as a finite state machine).

What made the difference in our work was the realization that for commitment protocols, the assumptions among the agents are the commitments themselves, not the order of the messages they exchange. Reconstructing the above problem scenario in terms of commitments, the problem is that the customer would infer that the merchant is, on account of the payment, committed to sending the items offered, whereas the merchant, on account of the cancellation, would infer that it is not. In other words, they would have an incompatible view of the state of the interaction, and their engagement would break down. This motivates a definition of interoperability in terms of commitments (from [4]). In the definition, the local perspective of an agent refers to the sequence of messages it has observed, both sent and received; a system execution is essentially a snapshot in time: it refers to a set of local perspectives, one for each agent.

Definition 1. *A multiagent system is aligned (interoperable) if and only if, for every relevant system execution, for every pair of agents* (x, y) *in the system:*

- *if the local perspective of* y *entails a commitment of the form* $\mathsf{C}(x, y, r, u)$*, then the local perspective of* x *entails it too.*

(The meaning of "relevant" in the above informal definition is beyond the scope of this paper.) Essentially what Definition 1 states is an invariant on (relevant) system executions. The interesting thing from the perspective of accommodating flexibility is that it does not refer to the sending or receiving of messages by any agent whatsoever. That represents the beginning of freedom from synchronous executions, which facilitates the reconciliation of flexibility with interoperability. In [5], we proposed that commitment protocol specifications specify only meanings of messages, not the order in which agents should exchange messages—another step in our argument. This was in contrast to earlier work, including our own, that relied on ordering constructs to some extent. In [4], we proposed the computational rules that ensure that the invariant in Definition 1 actually holds—another step. In [6], we presented an architecture in which the computational rules constitute a middleware—logically speaking, the final step of our argument.

In reality, the actual thinking and research evolved far more haphazardly than the steps above might indicate. The main point though is this: we wanted highly flexible protocols. So we specified protocols in terms of message meanings. However, meaning-based specifications make interoperability challenging. So we formulated a set of computational rules that guarantee interoperability. The rules form the basis of a middleware that the agents run upon. From the application (agent) perspective, the middleware offers the guarantee of interoperability; its implementation is, however, transparent to agents—just as reliable message queues offer guarantees about message delivery, but are transparent to applications that use them.

4 Conclusions

The knowledge flow between distributed systems research and multiagent systems research has largely been in one direction: toward multiagent systems research. Distributed open systems have informed multiagent systems research since the very beginning [7–9]. The flow need not be one way though. We, as a community of researchers in multiagent systems, place a high value on accommodating the autonomy of agents. Therefore, we value flexibility in protocol enactment. We value social abstractions. These criteria are not central to traditional distributed systems research, but are clearly central to practical distributed systems applications. If we formulate problems keeping our own values in sight, there is a significant potential for influencing the building of large distributed systems that are comprised of multiple autonomous parties [10]. Dealing with multiple autonomous parties is the need of the moment in areas such as health care, e-governance, and interorganizational business processes. A recent collection of manifestos [11] lays out interesting research directions in protocols and multiagent systems.

Acknowledgments. Amit Chopra's contribution was partially supported by a Marie Curie Trentino Cofund award. Munindar Singh was partially supported by National Science Foundation Grant #0910868.

References

1. Chopra, A.K., Singh, M.P.: Producing compliant interactions: Conformance, coverage, and interoperability. In: Declarative Agent Languages and Technologies IV: Selected, Revised, and Invited Papers. Volume 4327 of LNCS., Springer (2006) 1–15
2. Baldoni, M., Baroglio, C., Chopra, A.K., Desai, N., Patti, V., Singh, M.P.: Choice, interoperability, and conformance in interaction protocols and service choreographies. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems. (2009) 843–850
3. Winikoff, M.: Implementing commitment-based interactions. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems. (2007) 1–8
4. Chopra, A.K., Singh, M.P.: Multiagent commitment alignment. In: Proceedings of the Eighth International Conference on Autonomous Agents and MultiAgent Systems. (2009) 937–944

5. Chopra, A.K., Singh, M.P.: Constitutive interoperability. In: Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems. (2008) 797–804
6. Chopra, A.K., Singh, M.P.: Elements of a business-level architecture for multiagent systems. In: Proceedings of the 7th International Workshop on Programming Multi-Agent Systems. Volume 5919 of LNCS., Springer (2010) 15–30
7. Huhns, M.N., ed.: Distributed Artificial Intelligence. Pitman/Morgan Kaufmann, London (1987)
8. Gasser, L., Huhns, M.N., eds.: Distributed Artificial Intelligence, Volume II. Pitman/Morgan Kaufmann, London (1989)
9. Hewitt, C.: Open information systems semantics for distributed artificial intelligence. Artificial Intelligence **47**(1–3) (1991) 79–106
10. Chopra, A.K.: Social computing: Principles, platforms, and applications. In: Proceedings of the 1st Workshop on Requirements Engineering for Social Computing. (2011) to appear.
11. Chopra, A.K., Artikis, A., Bentahar, J., Colombetti, M., Dignum, F., Fornara, N., Jones, A.J.I., Singh, M.P., Yolum, P.: Research directions in agent commmunication. ACM Transactions on Intelligent Systems and Technologies (2011) To appear.