Enriching Business Artifacts with Coordination

Matteo Baldoni, Cristina Baroglio, Federico Capuzzimati, Roberto Micalizio

Università degli Studi di Torino — Dipartimento di Informatica
c.so Svizzera 185, I-10149 Torino (Italy)
{firstname.lastname}@unito.it

Abstract. This paper proposes to enrich the artifact-centric approach in two ways. First, relying on the Agent-Oriented Paradigm (AOP), the tasks acting on artifacts are organized in agents, seen as autonomous loci of control, whose execution is goal-driven. Second, the business artifact model is complemented by a normative dimension. Norms are used to represent the data lifecycle in a form that is inspectable and reasoned upon by agents. Agents can therefore create expectations about the behaviors of others and hence, leveraging on the norms, agents can act on an artifact so as to entice, or oblige, others to act themselves. The paper discusses the advantages and consequences of this norm-aware enrichment, and outlines a possible realization based on social commitments.

Keywords: Business Artifacts, Normative MAS, Social Commitments

1 A Normative Approach to Business Artifacts

The *artifact-centric* approach [5,9,8] is recently emerging as a viable solution for specifying and deploying business operations by combining both data and process as first-class citizens. In particular, the notion of *Business Artifact*, initially proposed by Nigam and Caswell [11], opened the way for the development of a data-driven approach to the modeling of business operations. The data-driven approach counterposes a data-centric vision to the activity-centric vision, traditionally used when workflows are explicitly modeled in terms of processes. *Artifacts* are concrete, identifiable, self-describing chunks of information, the basic building blocks by which business models and operations are described. They are business-relevant objects that are created and evolve as they pass through business operations. They include an *information model* of the data, and a *lifecycle model*, that contains the key states through which the data evolve, together with their transitions (triggered by the execution of corresponding tasks). A change to an artifact can trigger changes to other artifacts, possibly of a different type. The lifecycle model is not only used at runtime to track the evolution of artifacts, but also at design time to understand who is responsible of which transitions. In [6], the artifact-centric model is at the basis of the BALSA data-centric declarative model of business operations. The BALSA methodology can be summarized in three steps: 1) identify the relevant business artifacts of the problem at hand and their lifecycles, 2) develop a detailed specification of the services (or tasks) that will cause the evolution of the artifact lifecycles, 3) define a number of ECA-rules (Event-Condition-Action) that create associations between services and artifacts. ECA-rules are the building blocks to define, in

X. Franch, J. Ralyté, R. Matulevičius, C. Salinesi, and R. Wieringa (Eds.):
CAiSE 2017 Forum and Doctoral Consortium Papers, pp. 17-24, 2017.

a declarative way, processes operating on data. A way to coordinate different processes operating on the same artifacts is by means of choreographies. However, BALSA does not specify how choreographies should be used to coordinate declarative processes.

Although the BALSA model is extremely interesting and introduces a novel perspective on the modeling of business processes, we deem the absence of an explicit representation of the coordination structure within the model a significant flaw. In particular, in inherently *destructured* settings, as *cross-organizational interactions*, the involved actors are all peers, each of which has *its own* business goals, and acts in an autonomous way. Each actor does not know and does not care about the possible goals of the others. Nevertheless, actors generally need to interact to achieve goals they would not be able to achieve alone. The interaction is a critical dimension that need to be explicitly modeled to coordinate the usage of shared resources. This poses the question of how to scale the business artifact model to *coordinate* autonomous entities.

We see in the introduction of a *coordination model within business artifacts* the way to achieve this goal, and explain what we mean with a simple example. Let us consider a purchase scenario, involving a merchant and a client. We claim that in order to coordinate the interaction between the two agents, it is necessary to add to the plain message exchange (which standard approaches to business processes envisage as the only means of interaction), one further abstraction that explicitly represents the engagements each player has towards the other. We also claim that business artifacts should trace such engagements and their evolution, in order to enable an effective agent coordination. For example, when offering to sell some goods, the merchant *commits to* the client to ship the items the client will pay for. Such a commitment is stored by the business artifact involved in the interaction between the two. Because of his *awareness* of such a commitment, the client, having paid for the goods, *expects* the shipment to occur. If this does not happen, the commitment progresses into a state of "violation" and this information, stored in the business artifact, provides a proof of the merchant's misbehavior. From a different perspective, a client is enticed to use a business artifact by the merchant's commitment, which makes explicit the course of interaction the merchant binds to, and creates a right on the client that such an expected course of action be respected (i.e., my payment will put an obligation on the merchant to ship the bought items or the merchant will violate the commitment). On the other hand, the merchant uses commitments inside the business artifact to entice interactions with potential clients – indeed, the obligation yielded by a commitment is activated only if a client pays for some goods.

In the example, the commitments that go along with a business artifact make explicit the behavior the agents are expected to stick to. They also have a normative flavour, as diverging behaviors will be considered as violations. This awareness causes agents to take part to an interaction only if they are fine with the commitments. As such, commitments provide a standard to define standards of interaction mediated by business artifacts. To realize this vision, we claim that (1) services should be encapsulated and organized into *goal-oriented containers*; (2) it is necessary to introduce a *normative layer*. For what concerns (1), the Agent-Oriented Paradigm (AOP), briefly introduced in Section 2, is a good candidate. In particular, the Agent and Artifact meta-model (A&A) [12] has already shown how artifacts can be used as environment components that mediate agents' interactions. However, artifacts in the A&A model are radically different from the business artifacts because they *do not come* with an explicit information model for data, and they do not exhibit data lifecycles. Thus, this information cannot be exploited at design time, nor at runtime, to reason about which actions should be taken towards the achievement of an agent's goals. Concerning (2), the normative layer would provide an explicit representation of the business artifacts lifecycles, and of how coordination is expected to occur. Such a representation would allow agents to reason about the use of business artifacts and to create *mutual engagements* for driving their activities. Indeed, we envisage engagements as encoding *causal relations* between the actions of an agent and the goals and actions of another, with a normative power that would allow each agent to have expectations on the behavior of the others. In the purchase example, it is easy to see how the introduction of a norm in form of the commitment *whenever a customer pays, the merchant will ship the goods*, would enhance coordination.The customer now knows that after service *pay*, the merchant will be pushed to consider the service *ship-goods* as one of its next goals, otherwise it will violate the norm and will be sanctioned for that. This provides the customer a guarantee about the achievement of its own goal (or to recoup its losses). An explicit normative layer plays a central role both at the design time, to verify whether all the engagements can converge towards their satisfaction, and at running time to monitor the execution of a system and determine the violation of engagements. In this paper we introduce the notion of normative business artifacts as a means to extend the artifact-centric approach with a normative layer, where engagements and norms are expressed in terms of social commitments [13]. The introduction of a normative layer in the more general setting of business processes is seen as desirable also in [15].

2 Coordination via Normative Business Artifacts

Business artifacts are, by definition, data-aware. They consider data as a first-class primitive that drives the process modeling [5]. Artifacts, however, are not an end in themselves. They are business relevant entities that are created, accessed, and manipulated by different services along a business process. Business artifacts, however, are also resources on whose use interacting parties coordinate. We now show how to introduce a normative layer that supports such coordination.

Introducing Goal-oriented Containers. Destructured business processes call for a modularization of the control flow. AOP [7,17] is conceived exactly for handling multiple and concurrent control flows. Two elements are central in AOP: the *agents* and the *environment*. Agents, as abstractions of processes, possess their own control flow, summarized as the cyclic process in which an agent observes the environment (updating its beliefs), deliberates which intentions to achieve, plans how to achieve them, and finally executes the plan [7]. Beliefs concern the environment. Intentions lead to action [17], meaning that if an agent has an intention, then the expectation is that it will make a reasonable attempt to achieve it. In this sense, intentions play a central role in the selection and the execution of actions, which represent the innate capabilities agents have to modify their environment. Among others, (business) artifacts (see A&A-meta model [12]) are privileged elements of an environment. In particular, in contexts where agents cannot achieve their goals on their own, but need to interact with other agents to do so, artifacts provide shared resources that agents will use to mediate their interactions.

Introducing Norm-awareness. We claim that business artifacts should be *norm-aware* in two ways. First, the *lifecycle* of a business artifact should be made explicit by way of norms that specify the rules by which the data evolve. The agents (i.e., the artifact users), will be able to inspect and reason upon them to decide if and how to operate on an artifact to obtain some result. Second, agents need to coordinate and regulate their interaction in *using the artifacts* to achieve their goals. Given these two bodies of norms, agents will apply reasoning techniques to plan proper *coordination* that, possibly without violating any norm, will lead to goal achievement. This is possible because norms enable the creation of *expectations* and *commitments* among agents. Moreover, given an explicit representation of such elements it will be possible to realize systems of *accountability* to discourage or to detect and explain deviant behavior [4].

Even though data-awareness and norm-awareness are by and large orthogonal to BDI [17] notions, it is natural to think of agents as BDI agents for a seamless integration of all the aspects of deliberation, including the awareness of data and of their lifecycles. For instance, an agent, that is involved in handling orders, may conclude that, since it has to pick up three items in the warehouse, since each such item is to be packed, since all packagings are performed by a same other agent, and since one of its goals is saving energy, it is preferable to pick them up altogether, and deliver them to the other agent only afterwards, instead of picking and delivering one item at a time. Data-awareness here is awareness that three items of a same kind are requested. Norm-awareness that items are picked because each of them is part of some order, whose lifecycle says that after being picked they will be packed. Again data-awareness allows our agent to know that all parcels are to be made by a same other agent.

Relying on AOP is promising also because a the agent-based model allows to naturally tackle the issue of coordination by introducing the concept of *norm* [16]. The deliberative cycle of agents is affected by the norms and by the obligations these norms generate as a consequence of the agents' actions. The limit of current AOP approaches is that they provide no holistic proposal where constitutive norms are used also to specify data operations, and where regulative norms are used to create expectations on the overall evolution of the system (agents behavior and environment evolution).

Environment/Information systems based on normative business artifacts. Figure 1 describes the high-level architecture of the kind of system we imagine: (1) involving business artifacts and agents (with their goals), and (2) holistically norm-aware. Agents interact with each other and with the environment by creating and modifying data which belong to an information system and that are reified by business artifacts. They are goal-driven and capable of coordinating with other agents by creating and exploiting commitments, obligations, permissions, and prohibitions. The conceptual model of the information system is described in terms of the norms that regulate the evolution of data, that is, data lifecycles, capturing how data pass from one state to another as a consequence of actions that are performed by some agent. Moreover, business artifacts will include all those normative elements that regulate the coordination of the agents that interact by way of the artifact. All this information is available to the interacting agents in a form that allows agents to reason on it. The agents are aware of the current state

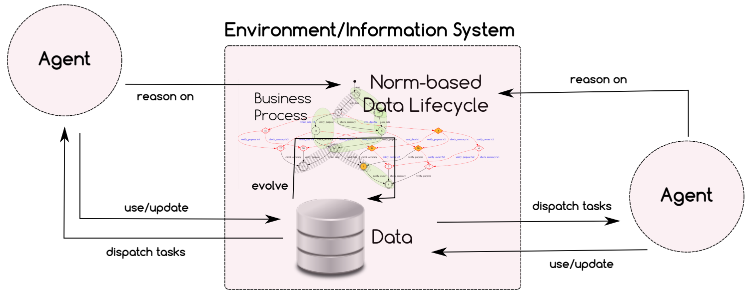

Fig. 1. Environment/Information system based on normative business artifacts.

(of the lifecycle) of the data, as well as of the obligations, prohibitions, commitments, permissions put on them, and thus they are aware of the tasks expected of them and of their parties. At any time it is possible to check the execution, identifying pending tasks and who is responsible of them, while behaviors that violate some norm (e.g. some obligation or some commitment) will be automatically detected, e.g. causing the activation of procedures that are specifically designed to handle the case. As a final remark, at design time, norms would provide a programming interface between agents and their environment, given in terms of those state changes in the environment that are relevant to the agent and that the agent should tackle.

3 An Exemplification with 2COMM

We now show how the above concepts can be implemented by relying on the 2COMM/JaCaMo+ framework [3]. We refer to an implementation where the BDI agents are implemented in the Jason agent programming language, and where agents share artifacts, whose creation and manipulation involves an explicit creation and manipulation of social commitments. Social commitments [13] provide the normative layer and enable the coordination of the goal-driven agents. We show how the norm-driven artifact-based coordination of agents is realized in the purchase scenario. A social commitment $C(x, y, s, u)$ captures that agent x (debtor) commits to agent y (creditor) to bring about the consequent condition u when the antecedent condition s holds (s and u are conjunctions or disjunctions of events). Only the debtor of a commitment can create it. When s is $true$ the commitment is *detached* and turns into an obligation on the debtor. When u is $true$ the commitment is *satisfied*. A detached commitment that is canceled or whose consequent becomes $false$ is *violated*. The business artifact, besides representing the chunk of information at issue, maintains the created commitments, that can be inspected by the agents. Specifically agents will be notified of the changes to the business artifact state which include changes occurred to the commitments. Among other events, they will be aware of the detachment of commitments of which they are debtors, and of the satisfaction (violation) of commitments of which they are creditor.

The example implementation involves a merchant, selling on-line, and a customer agent. The merchant advertises some item and specifies the number of available units.

Its goal is to be *paid* for the sold units. The goal of the customer is to *get the goods* it is interested in. A customer starts an interaction by requesting a quotation for a number of units. When the merchant sends the quotation he also creates a commitment *C(merchant, customerId, accepted(price, quantity, customer), goods(customer))*, meaning that he commits with the customer that if the quotation is accepted, he will have the goods delivered to the customer. The customer is, thus, enticed to accept the quotation because the presence of the commitment, as part of the information provided by the business artifact, yields that this action will create an obligation on the merchant to deliver the goods that will make him achieve his goal. Should delivery not occur, thanks to the explicit representation of the commitment within the business artifact, and because of the violation of the obligation expressed by it, it would be possible to identify the merchant as the liable[1] party. The payloads (quotation, price, quantity) are stored in the business artifact. On the other hand, acceptance of a quotation binds the customer with the merchant to pay, by the creation of the commitment *C(customer, merchant, goods(customer), paid)*, leading the merchant to satisfy his goal. After payment, the merchant is expected to send a receipt. This expected interaction involves the execution of operations that are exposed by the business artifact (*quote*, *ship* and *emitReceipt* for the merchant; *request*, *accept*, *reject* and *sendEPO* for the customer).

```
+requestedQuote(Quantity, Customer_Id)
    <- quote(1000, Quantity, Customer_Id).
+cc(My_Role_Id, Customer_Role_Id, accept(Price, Quantity, Customer_Role_Id),
    Goods,"DETACHED") : enactment_id(My_Role_Id)
    <- ship(Customer_Role_Id, Quantity).
+cc(My_Role_Id, Customer_Role_Id, paid(Customer_Role_Id), Receipt,"DETACHED")
    : enactment_id(My_Role_Id)
    <- emitReceipt(Customer_Role_Id).
```

Above, an excerpt of the merchant agent program. The merchant is solicited to act by the reception of a *requestedQuote* event, that comes from a customer through the business artifact. The execution of *quote* sends a quotation to the customer and causes the creation of the merchant's commitment to send the goods if the quotation is accepted. The detachment of such a commitment (due to an event raised by the customer) is notified to the merchant by the artifact alongside with the relevant information. The merchant will now try to satisfy the commitment by executing *ship*. The other commitment is detached when the customer pays, causing the merchant to emit a receipt. It is, thus, possible to observe how coordination is regulated by the commitments. Since commitments are created by their debtors, it is natural to assume that such debtors will have the code for tackling their detachment.

4 Conclusions

The presented work is strictly related to the problem of interaction in multiagent systems. In these systems, interaction is mainly focused on the modeling of communication patterns (*protocols*), which are concerned with the sequence of messages that can be exchanged between two communicating agents, but disregard the information conveyed by these messages. Recent approaches such as HAPN [18] and BSPL [14]

[1] The merchant himself committed to have the goods delivered in case of payment.

have started to consider also the information dimension. HAPN is formally based on automata where nodes represent states of the interaction and transitions between nodes represent the messages that can be exchanged. Transitions have a complex structure since for each message it is possible to define a guard condition on message sending. A similar approach is BSPL where the information flow is decomposed in a number of "simple protocols", each defining the schema of the messages that can be exchanged together with their parameters. Parameter are decorated as *in* or *out* (meaning it is received or emitted). BSPL provides a formal framework in which it is possible to verify properties such as liveliness and safety of a protocol. Both HAPN and BSPL, however, show some weaknesses in properly handling information. In HAPN, for instance, guards, that enable message sending, may refer to information which is not carried by the message itself, but rather maintained in an external information system, which is not an integral part of the HAPN proposal, and hence the complete verification of an interaction is not actually achievable. BSPL, on the other hand, assumes a distributed view of information. Each participant has its own knowledge base, and the progression of the interaction makes the local knowledge bases evolve. The problem, in this case, is that each participant has just a local view of the information lifecycle. Thus, an agent cannot create expectations about the behaviors of other participants as a consequence of the messages it sends. The approach we propose overcomes these limitations. Business artifacts abstract an information system, and provide the environment in which the agents, which are autonomous loci of control, interact. Both business artifacts and agents are first-class components. The autonomy and flexibility of the agents are preserved and supported; moreover, it is possible to reason both on the evolution of the business artifacts and on the interaction. This work can be extended along three main lines of research. First of all, an explicit normative layer paves the way to formal verification techniques for cross-organizational business processes. In this respect, the notion of *accountability* is rapidly gaining importance since, when more organizations come into play, it is even more important to trace back who is responsible for what. First steps can be found in [4]. Another promising extension is to understand how agents could plan the use of business artifacts for reaching their goals. An initial attempt to use social commitments in planning has been discussed in [2], but business artifacts are yet to be considered. Finally, the standardized lifecycle of commitments can be the key for developing an agent programming methodology, similar to the one discussed in [1]. The idea is to program agents so that they can properly tackle part of the events that are generated in the business artifacts of their interest; specifically, the state transitions that occur to commitments in which they are involved. To conclude, we mention RAW-SYS [10], which enriches the prescriptive process model with data-awareness. Although RAW-SYS looks similar to a (normative) business artifact, the objectives of the two models are quite different. RAW-SYS is essentially a framework for verifying business processes taking into account both the control- and the data-flows. A normative business artifact, instead, aims at coordinating autonomous agents.

Acknowledgements This work was partially supported by the *Accountable Trustworthy Organizations and Systems (AThOS)* project, funded by Università degli Studi di Torino and Compagnia di San Paolo (CSP 2014).

References

1. Matteo Baldoni, Cristina Baroglio, Federico Capuzzimati, and Roberto Micalizio. Empowering agent coordination with social engagement. In *AI*IA 2015, Advances in Artificial Intelligence, LNCS 9336*, pages 89–101, 2015.
2. Matteo Baldoni, Cristina Baroglio, Federico Capuzzimati, and Roberto Micalizio. Exploiting social commitments in programming agent interaction. In *PRIMA 2015: Principles and Practice of Multi-Agent Systems*, pages 566–574, 2015.
3. Matteo Baldoni, Cristina Baroglio, Federico Capuzzimati, and Roberto Micalizio. Commitment-based agent interaction in jacamo+. *Fundamenta Informaticae (to appear)*, 2017.
4. Matteo Baldoni, Cristina Baroglio, Katherine M. May, Roberto Micalizio, and Stefano Tedeschi. Computational accountability. In *Proceedings of the AI*IA Workshop on Deep Understanding and Reasoning*, pages 56–62, 2016.
5. K. Bhattacharya, N. S. Caswell, S. Kumaran, A. Nigam, and F. Y. Wu. Artifact-centered operational modeling: Lessons from customer engagements. *IBM Syst. J.*, 46(4):703–721, 2007.
6. K. Bhattacharya, R. Hull, and J. Su. *A data-centric design methodology for businessprocesses*, pages 503–531. Handbook of Research on BusinessProcess Modeling. IGI Publishing, 2009.
7. Michael E. Bratman. What is intention? In P. Cohen, J. Morgan, and M. Pollack, editors, *Intensions in Communication*, pages 15–31. MIT Press, Cambridge, MA, 1990.
8. Diego Calvanese, Giuseppe De Giacomo, and Marco Montali. Foundations of data-aware process analysis: a database theory perspective. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS*, pages 1–12. ACM, 2013.
9. David Cohn and Hull Richard. Business Artifacts: A Data-centric Approach to Modeling Business Operations and Processes. *IEEE Data Eng. Bull.*, 32(3):3–9, 2009.
10. Riccardo De Masellis, Chiara Di Francescomarino, Chiara Ghidini, Marco Montali, and Sergio Tessaris. Add data into business process verification: Bridging the gap between theory and practice. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 1091–1099, 2017.
11. A. Nigam and N.S. Caswell. Business artifacts: An approach to operational specification. *IBM Systems Journal*, 42(3):428 – 445, 2003.
12. Andrea Omicini, Alessandro Ricci, and Mirko Viroli. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17(3):432–456, December 2008. Special Issue on Foundations, Advanced Topics and Industrial Perspectives of Multi-Agent Systems.
13. Munindar P. Singh. An ontology for commitments in multiagent systems. *Artif. Intell. Law*, 7(1):97–113, 1999.
14. Munindar P. Singh. Information-driven interaction-oriented programming: BSPL, the blindingly simple protocol language. In *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 491–498, 2011.
15. Munindar P. Singh. NoBPM: Supporting Interaction-Oriented Automation via Normative Specifications of Processes, 2015. Invited talk, BPM.
16. Göran Therborn. Back to norms! on the scope and dynamics of norms and normative action. *Current Sociology*, 50:863–880, 2002.
17. Michael J. Wooldridge. *Introduction to multiagent systems, 2nd edition*. Wiley, 2009.
18. Nitin Yadav, Lin Padgham, and Michael Winikoff. A tool for defining agent protocols in HAPN: (demonstration). In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pages 1935–1936, 2015.