Programming with Commitments and Goals in JaCaMo+ (Extended Abstract)

Matteo Baldoni, Cristina Baroglio, Federico Capuzzimati, Roberto Micalizio
Università di Torino, Italy

Categories and Subject Descriptors: I.2.11 [Distributed Artificial Intelligence]: Coherence & co-ordination; multiagent systems.

General Terms: Design.

Keywords: Social Computing, Agent Programming, Commitments and Goals, Agents & Artifacts, JaCaMo.

1. INTRODUCTION

Multiagent systems (MASs) provide a natural way to realize applications where a group of autonomous actors coordinate their interactions. Nevertheless, a growing feeling [6, 5] is that there is the need of equipping agents with the means for developing a social behavior. Social aspects are not limited to message exchanges nor to the mere realization of teamwork. An agent who perceives another agent will try to predict its actions and intents, or it may try to establish forms of cooperation in the achievement of some goals. Crediting Singh [9], for capturing the social computation, it is necessary to maintain the *social state* of the system; i.e., the state of such a computation. Starting from 2COMM4JADE [2], we developed *JaCaMo+*, an agent-based framework that builds upon JaCaMo [4]. JaCaMo+ provides social abstractions specifically devised to capture the state of social computations. The proposal shows the importance of Social Computing also from a Software Engineering perspective, inasmuch it increases the decoupling of the interacting parties (keeping at the same time their views of the social state automatically aligned), and hence preserving their autonomy. This fosters the modularity of the system, improving its maintenance. Last, when an agent uses a JaCaMo+ artifact it accepts to satisfy the engagements it will be involved in along the interaction.

2. JACAMO+

Most of the current multiagent frameworks and platforms spread the interaction logic across the agents' implementations. The drawback is that in order to change the coordination among the agents there is the need to modify the behaviors of the agents themselves. JaCaMo+ approaches the coordination problem by relying on an explicit representation of the *social state*, defined in terms of *social relationships* among the agents, and in terms of the rules that cause the social state to evolve. Both agents and social relationships are first-class entities that interact in a bi-directional manner. Social relationships are created by the execution of *interaction protocols*, and *provide expectations* on the agents' behaviors; existing social relationships affect the decisions of the agents they involve. It is, thus, necessary to provide the agents with the means to create and manipulate social relationships, and to observe, reason and deliberate on these relationships and on how they affect (and are affected by) the *goals* each agent has.

We focus on social relationships that can be represented as *commitments* [8]. A commitment $\mathsf{C}(x, y, s, u)$ captures that agent x (debtor) commits to agent y (creditor) to bring about the consequent condition u when the antecedent condition s holds. Antecedent and consequent conditions are conjunctions or disjunctions of events and commitments. Commitments are manipulated by means of the standard operations *create*, *cancel*, *release*, *discharge*, *assign*, *delegate*. They have a *normative value*, providing social expectations on the agents' behaviors. They also satisfy the requirement that social relationships cannot but concern the observable behavior of the agents themselves. As a consequence, commitments can be used by agents in their *practical reasoning* together with beliefs, intentions, goals.

The *relation between commitments and goals* is studied in [10] and formalized in terms of *practical* rules, which capture patterns of pragmatic reasoning: (1) rules from goals to commitments capture how commitments evolve when the state of some goals change; and (2) rules from commitments to goals capture how a goal evolves when the corresponding commitment changes in the social state.

In [10] an agent goal G has its own lifecycle, and is modeled as $\mathsf{G}(x,\ p,\ r,\ q,\ s,\ f)$, where x is the agent pursuing G, p is a precondition that must be satisfied before G can become Active, r is an invariant condition that is true when G becomes Active and holds until the achievement of G, q is a post-condition (effect) that becomes true when G is successfully achieved, and finally, s and f are the success and failure conditions, respectively. Goals and commitments are closely related. Intuitively, when agent x wants to achieve goal G (i.e. it wants to reach condition s) but it cannot do it autonomously, it can convince another agent y to bring about s by offering a service u; this is naturally represented by the commitment $\mathsf{C}(x, y, s, u)$.

JaCaMo+ extends the JaCaMo framework [4]. As [3], JaCaMo+ reifies commitments as resources by extending CArtAgO artifacts and allowing a seamless integration of Jason

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

BDI agents with social commitments. Moreover, JaCaMo+ agents have also access to the social state of the artifact they are focusing on. The social state is mapped onto a portion of the belief base each agent has; the underling JaCaMo platform guarantees that any change occurred in the artifact's social state is instantaneously propagated to the belief bases of all the focusing agents. Agents are thereby constantly aligned with the social state.

From an organizational perspective, a JaCaMo+ artifact encodes a commitment protocol, that is structured into a set of roles, and that represents different ways of manipulating the social state. By enacting a role, an agent receives social powers by the artifact, whose execution has public social consequences, expressed in terms of commitments.

We extend the *Jason* component of JaCaMo by allowing the specification of plans whose triggering events involve social relationships; i.e., commitments. In JaCaMo+, a commitment is represented as a term *cc*(*debtor*, *creditor*, *antecedent*, *consequent*, *status*) where *debtor* and *creditor* are the identities of the involved agents, while *antecedent* and *consequent* are the commitment conditions: the *debtor* is responsible towards the *creditor* agent for the satisfaction of commitment. *Status* is the commitment state (created, satisfied, violated, conditional, detached, expired, pending, terminated as in [7]). Commitment operations are realized as *internal operations* of the new class of artifacts we add to CArtAgO. They cannot be invoked directly by agents, but the protocol actions use them to modify the social state. Commitments can be used inside a *plan context* or *body.* Differently than beliefs, commitment assertion/deletion can only occur through the artifact, as a consequence of a change of the social state. Agents also have goals of their own, for the sake of discussion, we abstract the notation $\mathsf{G}(x, p, r, q, s, f)$ in [10] in terms of a simple label G.

2.1 Programming in JaCaMo+

Our approach positively impacts on agent programming in two ways: (1) the dependencies encoded by way of social relationships allow agents to include in their practical reasoning also aspects that concern other agents; (2) social relationships facilitate the decoupling among the agents because the coordination resides inside protocol artifacts instead of residing in the agents themselves. We, now, show how easy it is to implement in JaCaMo+ practical rules that encompass both agent goals and social relationships (commitments). We take inspiration from the rules in [10].

Entice. This rule captures the case where agent x can achieve G only with the help of agent y: x creates an offer to agent y such that, if y brings about s (success condition of G), then x will engage into achieving a condition u of interest for y. Such offer is naturally modeled as the commitment $C(x, y, s, u)$. The corresponding JaCaMo+ rule template is:

```
+!G : p
  <-?r;
    social_action;
    ?cc(x, y, s, u, CONDITIONAL).
```

The post-condition corresponds to the existence of a commitment, while G success condition s is the antecedent condition that will be achieved by y. The body consists of a protocol action offered by the artifact x focused on, and whose meaning is the creation of commitment C.

Deliver. If commitment C becomes detached, then debtor x activates a goal $G_1 = \mathsf{G}(x, p, r, q, u, f)$ to bring about the consequent. In JaCaMo+:

```
+cc(x, y, s, u, DETACHED) : context
  <- !G1; ?cc(x, y, s, u, SATISFIED).
```

The test goal at the end of the rulet allows x to verify after G_1 is achieved, its corresponding commitment is satisfied.

Detach. When a conditional commitment $C_1(y, x, s', t)$, appears in the social state, the creditor x activates a goal $G_2 = \mathsf{G}(x, p'.r', q', s', f')$ to bring about the commitment antecedent. The corresponding JaCaMo+ rule template is:

```
+cc(y, x, s', t, CONDITIONAL) : context
  <- !G2; ?cc(y, x, s', t, DETACHED).
```

As in the previous case, x can verify that, after the satisfaction of G_2, the corresponding commitment is detached.

One of the strongest points of the proposal is the *decoupling* between the design of the agents and the design of the interaction, that builds on the decoupling between computation and coordination done by coordination models like tuple spaces. Agent behavior is defined based on the existing social relationships and not on the process by which they are created. For instance, in CNP the initiator becomes active when the commitments which bind it to accept or reject the proposals are detached. It is not necessary to specify nor to manage, inside the agent, such things as deadlines or counting the received proposals. The decoupling allows changing the definition of the artifact without the need of changing the agents' implementation. We plan to leverage this characteristic and the typing system in [1], to support the realization of open multiagent systems.

REFERENCES

[1] M. Baldoni, C. Baroglio, and F. Capuzzimati. Typing Multi-Agent Systems via Commitments. In *Post-Proc. of EMAS 2014*, *LNCS* 8758, pages 338–405. Springer, 2014.

[2] M. Baldoni, C. Baroglio, and F. Capuzzimati. A Commitment-based Infrastructure for Programming Socio-Technical Systems. *ACM Transactions on Internet Technology*, 14(4):23:1–23:23, December 2014.

[3] M. Baldoni, . Baroglio, and F. Capuzzimati. Social Computing in JaCaMo. In *Proc. of the ECAI*, pages 959–960. IOS Press, 2014.

[4] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, and A. Santi. Multi-agent oriented programming with JaCaMo. *Sci. of Comp. Prog.*, 78(6):747–761, 2013.

[5] F. Dignum, R. Prada, and G. J. Hofstede. From autistic to social agents. In *Proc. of AAMAS*, pages 1161–1164. IFAAMAS, 2014.

[6] G. A. Kaminka. Curing robot autism: a challenge. In *Proc. of AAMAS*, pages 801–804. IFAAMAS, 2013.

[7] F. Meneguzzi, P. R. Telang, and M. P. Singh. A first-order formalization of commitments and goals for planning. In *Prof. of AAAI.* AAAI Press, 2013.

[8] M. P. Singh. An ontology for commitments in multiagent systems. *A. I. Law*, 7(1):97–113, 1999.

[9] M. P. Singh. Social computing: Principles, methods, and technologies, 2014. Invited talk at the 1st Int. WS on Multiag. Found. of Social Computing.

[10] P. R. Telang, N. Yorke-Smith, and M. P. Singh. Relating Goal and Commitment Semantics. In *Proc. of ProMAS 2011*, *LNCS* 7212, pages 22–37. Springer, 2012.