Evolving Commitments for Self-Adaptive Socio-Technical Systems

Xin Peng*†, Yi Xie*†, Yijun Yu‡, John Mylopoulos§, Wenyun Zhao*†
*School of Computer Science, Fudan University, Shanghai, China
†Shanghai Key Laboratory of Data Science, Fudan University, China
‡Department of Computing, The Open University, UK
§Department of Information Engineering and Computer Science, University of Trento, Italy
Email: {pengxin, 11210240033, wyzhao}@fudan.edu.cn, y.yu@open.ac.uk, jm@disi.unitn.it

***Abstract*—Socio-technical systems (STSs) consist of human, hardware and software agents that work in tandem to fulfill stakeholder requirements. A specification for an STS consists of a set of (social) commitments among participating agents that serve as a contract among them. However, by their very nature, STSs are open, dynamic and continuously evolving along with their environments. To ensure that such systems continue to satisfy their requirements, the agents that comprise an STS must continuously adapt their behaviors to take into account risks and opportunities that arise at runtime. This paper presents a decision-theoretic self-adaptation framework that proposes candidate adaptation strategies for participating agents. These strategies are implementable by reconfiguration of plans or even changes in contractual commitments among agents. The adaptation procedure involves negotiating commitment changes and possible compensations to/from creditor agents. To evaluate the proposal, the paper also presents the results of an experimental study with a simulated STS, which confirms that success rates for achieving stakeholder goals and overall tradeoff utility can be improved significantly by incorporating evolving commitments to support STS adaptation.**

***Keywords*-Socio-Technical Systems, Open Systems, Agent Commitments, Software Adaptation**

I. Introduction

Socio-technical systems (STSs) consist of human, hardware and software elements as well as processes that work in tandem to fulfill stakeholder requirements [1], [2]. Insurance claim processing, supply chain management and stock market systems are all examples of STSs, characterized by heterogeneity and partial autonomy of their elements, complexity of the inter-dependencies therein, and uncertainty with respect to the environments wherein these system operate. What are the concepts in terms of which we can conceive such systems? How do we design them? And how do we ensure that they can cope with heterogeneity, autonomy, complexity and uncertainty?

An excellent starting point for answering such questions treats STSs as protocols consisting of a collection of commitments over a given set of roles that satisfy stakeholder requirements. Then a running STS can be regarded as a collection of agents who have assumed various roles in the STS's protocol and take responsibility for all the commitments that their respective roles entail. In such a system, each agent intends to best achieve its own goals relying on its own capabilities, but also on other agents through contractual commitments and social interactions.

It has been argued that STSs need self-adaptation capabilities in order to cope with uncertainty (e.g., [1], [2]), including self-optimization and self-repair. However, self-adaptation of STSs is complicated by their distributed, multi-agent nature where each agent can be a locus of adaptation, and adaptation typically involves interactions with other agents [3]. For example, an agent can delegate all/part of its commitments to an alternative agent. To analyze and reason about adaptation of STSs, one can use role-based goal models and commitment protocols that model agents and their social interactions.

We adopt the agent-based conceptual framework for adaptive STSs proposed earlier in [3], [4]. There, a commitment C is a tuple (Debtor, Creditor, Antecedent, Consequent) where the Debtor agent is committed to fulfill the Consequent condition for the Creditor agent if the Antecedent condition holds. The requirements for an STS are modelled as goals, and a specification for fulfilling these requirements (also known as *protocol*) is modelled as a collection of commitments [3]. The adaptation decision of an agent for a particular goal consists of selecting an alternative protocol for fulfilling that goal.

STSs by their very nature, are open systems. Existing approaches for open system adaptation (e.g. [3], [5]) assume that protocols are static, i.e., commitments between two agents remain unchanged during the operation of an STS. With this assumption, agents may be unable to benefit from adaptation when risks and opportunities arise at runtime. For example, a commitment between a restaurant (Debtor) and a customer (Creditor) about serving ordered food within 40 minutes may be impossible to fulfill due to a traffic jam. Or, a particular customer may request the food he ordered be delivered 10 minutes early. In both cases, renegotiating a new commitment (with possible compensation) instead of insisting on existing ones may be beneficial to participating agents by alleviating risks and/or optimizing business performance.

In summary, flexible adaptation mechanisms for an STS require the ability for each agent to reconsider and renegotiate its commitments with others, i.e., support evolving commitments. Thus adaptation involves not only changes of plans (i.e., how does an agent fulfill its existing commitments), but also changes of commitments. The main objective of

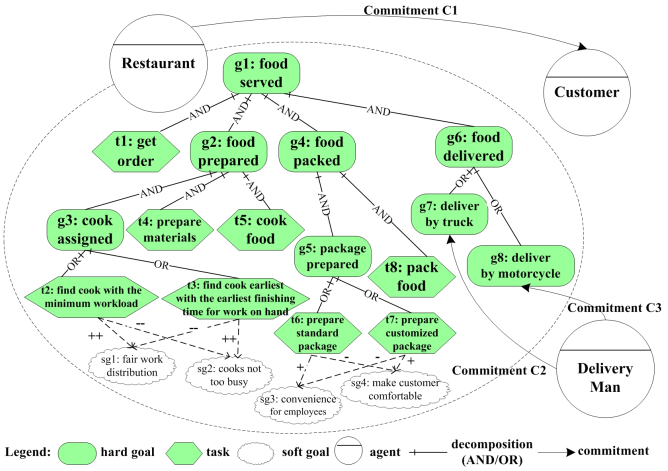

Fig. 1. Food Ordering Goal Model

this paper is to propose a decision-theoretic self-adaptation framework, that (1) defines adaptation strategies for a debtor agent involving both reconfiguration of plans and changes of commitments; (2) estimates and ranks costs and benefits of candidate adaptation strategies based on probabilistic utility functions; and (3) provides a procedure for a debtor agent to negotiate commitment changes and possible compensations with other agents. With this support, each debtor agent can rank and select the best adaptation strategy and make decisions and renegotiations for evolving commitments.

To evaluate the effectiveness of our framework, we conduct an experimental study with a simulated STS developed using AnyLogic [6]. In the study, we simulate adaptation scenarios caused by environmental and intentional changes and compare the fulfillment of stakeholder goals with and without evolving commitments. The study shows that the success rate of stakeholder goals and overall utility can be significantly improved by incorporating evolving commitments in STS adaptation.

The rest of the paper is structured as follows. Section 2 illustrates the idea of evolving commitments with a small real-life example. Section 3 presents our self-adaptation framework for STSs. Section 4 evaluates the proposed framework and discusses related issues. Section 5 introduces a number of existing proposals and compares them with ours, and finally section 6 draws conclusions.

II. A Motivating Example

Fig. 1 describes the goal model for a food ordering system involving three roles, Restaurant, Customer and Delivery Man. In the goal model, the goals of an agent are refined into subgoals through AND/OR refinements until reaching leaf-level tasks or goals. Leaf-level tasks can be accomplished by the agent itself with its own capabilities. Leaf-level goals can be delegated to other agents through social interactions and collaborations. Thus, a goal specification for the root goal of an agent (e.g., “g1: food served”) can be described as a sequence of tasks that can be accomplished by the agent itself (e.g., “t6: prepare standard package”) and goals that can be assigned to other agents (e.g., “g7: deliver by truck”).

Typically, a Customer (human agent) orders food from a Restaurant (organizational agent) via an online ordering system, then the Restaurant prepares the food and assigns a Delivery Man (human agent) to deliver the food to the Customer. When the Restaurant accepts a food order from a Customer, a commitment is established between them, for example, with antecedent condition “bill paid” and consequent condition “food served within 40 minutes”. Likewise, when the Delivery Man (a part-time employee of Restaurant) accepts a food delivery task, a commitment is established with the Restaurant with antecedent condition “job paid” and consequent condition “food delivered within 30 minutes”.

Assume that a customer Bob orders two set courses for his friend and himself from restaurant RES and pays 30 dollars online. RES promises to serve the food in 40 minutes. After 10 minutes, RES has the food prepared, packed and assigned to Jack for delivery by truck. Jack gets 1 dollar for this job and promises to deliver within 30 minutes. Consider now the following two cases.

Case 1: Jack is on his way to deliver the food by truck (goal “deliver by truck”) but gets stuck on a traffic jam. Upon reporting the problem, RES estimates that the current plan can’t be completed on time. According to state-of-the-art adaptation approaches, RES can switch to an alternative goal “deliver by motorcycle”, i.e., sending another delivery man to deliver the food by a motorcycle (assuming that a motorcycle is much faster during a traffic jam). This alternative, however, may not be the best choice when (1) it is still cheaper to use the current truck to deliver the food while compensating the customer for the delay; or (2) the risk of the food delivery being delayed is even higher since now the food needs to be delivered sooner than 30 minutes.

Case 2: Jack is on his way to deliver the food by truck. But Bob’s friend arrives 10 minutes early, so Bob calls RES to ask whether the food can be delivered 10 minutes early, adding that he can pay a 10 dollars compensation for the change. According to the current commitment between Jack and RES, he only needs to deliver within 30 minutes and gets 1 dollar. However, he is happy to hurry up and deliver in 20 minutes if he gets 2 dollars. The change of commitments here is clearly beneficial to all parties.

These two cases underscore the importance of considering commitment evolution during the operation of an STS, where:

- Initially established commitments may be renegotiated or cancelled during adaptation, given that compensation, delay, or cancellation are acceptable;
- Possible adaptation alternatives should be considered and evaluated together with possible commitment changes or cancellation by each participating agent;
- At all times, each agent selects the best adaptation strategy based on a cost-benefit analysis and makes decisions accordingly.

III. Adaptation Framework

To support evolving commitments, an agent shall incorporate commitment changes and cancellation into candidate strategies when making adaptation decisions. The cost-benefit analysis of a candidate strategy includes two parts: estimated value of achieving the target goals; possible compensation paid for commitment changes or cancellation. The value of a strategy is estimated with probabilistic utility functions by investigating the risks and expected utility of the strategy. Based on the utility estimation, the agent can rank and select candidate strategies and renegotiate commitments with other agents if required.

A. Adaptation Strategy

Following other conceptual models for open systems [3], [7], we model a participant in an STS as a goal-driven agent who has the capability to achieve certain goals and depends on other agents for the achievement of other goals. Such agent dependencies are modelled by commitments that are explicitly established and constitute publicly verifiable forms of social interaction. An agent acts on a target goal it wants to achieve and adapts when the current strategy for achieving the goal is inadequate or improvable [3]. During adaptation, a new strategy for the target goal may be activated.

According to Chopra et al. [3], an adaptation strategy consists of a set of goals, supported either by the agent's capabilities or commitments of other agents, that need to be achieved in order to achieve a root goal. Such a strategy is inherently static. During adaptation, however, existing commitments can be renegotiated or even cancelled through possible compensation. Therefore, considering possible commitment changes, an adaptation strategy in our framework consists of a goal specification, a set of commitments with other agents (changed or unchanged), and possible compensations. These concepts are defined more precisely below.

Definition 1. (Goal Specification) A *goal specification* is a collection of tasks (including goals assigned to other agents) an agent chooses from a goal model to achieve the root goal.

Definition 2. (Compensation) Upon cancellation or other changes to the initial commitment, a *compensation* consists of the resources returned or provided to an affected agent such that the commitment is sustainable within the open system.

Definition 3. (Adaptation Strategy) During adaptation decision-making with evolving commitments, an *adaptation strategy* consists of a goal specification, a proposal for commitment changes and corresponding compensations.

With evolving commitments, a debtor agent can consider not only different goal specifications for the target goal but also renegotiation of commitments with other agents. A debtor agent can try to keep current commitments by adopting better goal specifications. It can choose to renegotiate the commitments with other agents if it is too costly to adopt alternative specifications to satisfy its current commitments. It can also combine specification switching and commitment renegotiation.

For example, in the first case discussed in Section II, the initial commitment between RES and Bob, which can be represented by (RES, Bob, bill paid, food served within 40 minutes), is at risk due to an unexpected environmental change

TABLE I
POSSIBLE ADAPTATION STRATEGIES FOR RES RESTAURANT

Strategy	Specification	Commitment	Compensation
St1	{g8}	unchanged	0
St2	unchanged	time prolonged to 1 hour	to be negotiated
St3	{g8}	time prolonged to 50 minutes	to be negotiated
St4	{t7, t8, g8}	time prolonged to 55 minutes	to be negotiated
St5	N/A (cancelled)	N/A (cancelled)	to be negotiated

(i.e., traffic jam). Assume that the current goal specification for the root goal "food served" is $\{t1, t2, t4, t5, t6, t8, g7\}$ and a subset of the tasks $\{t1, t2, t4, t5, t6, t8\}$ have been accomplished successfully. Considering evolving commitments, there are several adaptation strategies for RES restaurant (see Table I). Apart from the strategy of ensuring the initial commitment by switching to "deliver by motorcycle" (St1), RES can also try to persuade Bob to accept a delay of 20 minutes with the current delivery arrangement (St2), to reduce the extent of predicted delay by switching to "deliver by motorcycle" (St3), to repack the food with customized package and redeliver the food by motorcycle (St4), or even to cancel the order (St5). For simplicity, other alternatives involving delivery men (Jack and others) are not included in the discussion.

The goal specification involved in an adaptation strategy can reuse the outcome of some completed tasks in the current attempt to achieve the target goal and cancel others. For example, both St1 and St3 involve the same goal specification, which reuses the outcome of a series of tasks ($\{t1, t2, t4, t5, t6, t8\}$) and cancels the outcomes of the other tasks ($\{g7\}$) that have been completed (i.e., making the delivery man come back to the restaurant). Another strategy St4 cancels more tasks ($\{t6, t8, g7\}$) to make the food repacked with customized package and redelivered by motorcycle.

Considering different proposals for commitment changes (e.g., different extension of food serving time), more concrete strategies can be derived. For commitment change or cancellation, proper compensation shall be negotiated with the affected agent (e.g., Bob for the strategies St2-St5).

B. Utility Estimation

The estimated utility of each adaptation strategy depends on the following three aspects.

- **Cost**: the cost of achieving the target goal, including money, needed resources, time, etc..
- **Probability**: the probability of satisfying the target goal and related commitments with the new goal specification, considering the uncertainties with respect to the environment and agent capabilities.
- **Softgoals and Preferences**: the overall satisfaction of related quality goals considering stakeholder preferences.

Therefore, our utility estimation combines probabilistic goals [8] and preference goals [9], [10] to capture the adaptation cost, the risk of goal failure, and the benefit gained from the satisfaction of stakeholder goals. To integrate utility estimation with compensation in commitment renegotiation (see Section III-C), both utility and compensation are defined under the same value system, for example, by currency.

For an adaptation strategy switching from goal specification S_0 to S_1 (S_0 and S_1 can be the same when no specification switch is involved) to achieve a target Goal G, the method for calculating cost, probability, benefit as the overall utility is given below. We assume that there is a maximum compensation allowed ($maxCom$) for different types of commitment changes (including cancellation). In other words, the compensation charged by a creditor or a debtor cannot exceed $maxCom$.

1) Cost: The tasks involved in a goal specification may consume various resources and human effort. The cost of an adaptation strategy consists of both the cost spent on those completed tasks in the latest attempt of achieving the target goal G with the current specification S_0, and the estimated cost for achieving G with the planned specification S_1. Apart from these two parts, there is possibly an additional cost for switching from S_0 to S_1. For example, for the first case in Section II, if the restaurant decides to try another alternative for food delivery (i.e., "deliver by motorcycle"), Jack needs to first drive back to the restaurant before attempting a redelivery by motorcycle.

The cost for completing or cancelling a task can usually be estimated from historical data and experience. Therefore, we assume that the cost for completing a set of tasks S and the cost for cancelling a set of tasks S can be calculated by $Cost_{task}(S)$ and $Cost_{task-cancel}(S)$, respectively. Suppose $Completed(S_0)$ represents the subset of tasks in S_0 that has been completed in the latest attempt, the cost of completed tasks and the cost of planned tasks can be computed by $Cost_{task}(Completed(S_0))$ and $Cost_{task}(S_1)$, respectively. For the switching cost, one can treat it as the cost of cancelling those completed tasks in S_0 that are replaced by their alternatives in S_1. The tasks to be cancelled ($Cancel(S_0, S_1)$) can be defined as the following set, where $Ancestor(t)$ represents the ancestor goal set of a task t, $decomposed(g, sg, OR)$ predicates that a subgoal or task sg is OR-decomposed from goal g.

$$\begin{aligned} Cancel(S_0, S_1) = \{t \mid t \in Completed(S_0) \bigwedge \exists t', g_0, g_1, g \ s.t. \\ t' \in S_1 \bigwedge g_0 \in Ancestor(t) \bigwedge \\ g_1 \in Ancestor(t') \bigwedge g_0 \neq g_1 \\ \bigwedge decomposed(g, g_0, OR) \\ \bigwedge decomposed(g, g_1, OR)\} \end{aligned}$$

Therefore, for an adaptation strategy involving a specification switch from S_0 to S_1, $Cost(S_0, S_1)$ can be estimated as follows.

$$\begin{aligned} Cost(S_0, S_1) = Cost_{task}(Completed(S_0)) + Cost_{task}(S_1) \\ + Cost_{task-cancel}(Cancel(S_0, S_1)) \end{aligned}$$

2) Probability: The probability estimation in our framework is based on the probabilistic way of specifying and reasoning about partial goal satisfaction proposed in [8]. We use quality variables and objective functions to specify the probability of achieving the target goals. For example, for the cases discussed in Section II, food serving time $time$ is a critical quality variable and $time \leq 40\ minutes$ is an objective function for the root goal "food served".

For a target goal G, we denote its quality variable as Qv_G and its lower limit and upper limit in the objective function as c_1 and c_2, respectively, then its probability of satisfaction is defined as:

$$P(G) = \int_{c_1 \leq x \leq c_2} pdf_{Qv_G}(x)\, dx$$

In the equation, pdf_{Qv_G} is the probability density function (pdf) of Qv_G.

If the target goal G is achieved through multiple tasks, we can follow the refinement patterns in [8] to aggregate the probability density functions from tasks to high-level goals. For example, for two tasks $t1$, $t2$ conforming to the milestone-driven refinement pattern, their probability density functions for the quality variable Qv_G can be aggregated as follows.

$$\begin{aligned} Qv_G &= Qv_{t1} + Qv_{t2} \\ pdf_{Qv_G}(x) &= \int_0^x pdf_{Qv_{t1}}(i) \times pdf_{Qv_{t2}}(x - i)\, di \end{aligned}$$

For an adaptation strategy for the target goal G with a switch from goal specification S_0 to S_1, involved tasks consist of completed tasks in S_0 that are reused, planned tasks in S_1, and completed tasks in S_0 to be cancelled. As the quality variables of completed tasks are fixed (constants), we have:

$$\begin{aligned} Qv_G &= Qv_{Completed(S_0)} + Qv_{S_1} + Qv_{Cancel(S_0,S_1)} \\ pdf_{Qv_G}(x) &= \int_0^{x - Qv_{Completed(S_0)}} pdf_{Qv_{S_1}}(i) \times \\ & \quad pdf_{Qv_{Cancel(S_0,S_1)}}(x - Qv_{Completed(S_0)} - i)\, di \end{aligned}$$

3) Benefit: The target goal G, if successfully completed, can bring benefit to the debtors who contributed to the completion. This benefit usually results from the debtor's commitments. For example, for the two cases in Section II, the benefit that the restaurant obtains from Bob can be estimated by the money Bob pays for his order (30 dollars). Besides this direct benefit obtained from the target goal G, there is also indirect benefit obtained from the satisfaction of preference "nice-to-have" goals [9]. Softgoals are typical preference goals with different priorities for stakeholders, e.g., "sg2: cooks not too busy" and "sg3: convenience for employees" in Fig. 1.

To conduct benefit estimation, we use direct benefit as basic estimation and augment it with an algorithm similar to the *calConfigScore* algorithm proposed in [10]. This algorithm calculates a score that indicates how positively a goal specification contributes to softgoals. The score is calculated as the sum of a goal specification's contributions to each softgoal. It provides a relative utility estimation for various indirect benefits obtained from the satisfaction of stakeholder quality requirements. The score calculation propagates satisfaction labels from tasks in the given goal specification to hard goals and softgoals on the basis of a set of goal reasoning axioms [11]. A softgoal

sg thus can have one or more satisfaction labels (denoted by $labels(sg)$) out of a series of satisfaction levels, e.g., FS (fully satisfied), FD (fully denied), PS (partially satisfied), and PD (partially denied). For each satisfaction label l, we assign a numeric value $satisf(l)$, e.g., 2, 1.5, 0.5, 0, to FS, PS, PD, and FD respectively. On the other hand, we assign a priority value $priority(sg)$ to sg, e.g., 1, 2, or 3, to reflect stakeholder priority for that goal. A given goal specification's contributions to sg ($Contribution(sg)$) thus can be calculated as follows.

$$Contribution(sg) = \sum_{l \in labels(sg)} satisf(l) \times priority(sg)$$

Therefore, for an adaptation strategy switching from goal specification S_0 to S_1 for target goal G, we estimate the potential benefit $Benefit(S_0, S_1)$ based on the direct benefit $Benefit_{direct}(G)$ and a score function $Score(S)$ where S is a goal specification. Here $(S_0 \cup S_1) \setminus (Cancel(S_0, S_1))$ represents a complete goal specification including both tasks in S_1 and completed tasks in S_0 that are kept. The fraction part of the expression below represents the utility of the current goal specification relative to the best goal specification of all possible goal specifications ($Spec(G)$) for achieving G. Minimum/maximum score specifications can be calculated efficiently using a minimum cost SAT solver, as in [12].

$$Benefit(S_0, S_1) = Benefit_{direct}(G) \times \frac{Score((S_0 \cup S_1) \setminus (Cancel(S_0, S_1)))}{\max\limits_{S \in Spec(G)} Score(S)}$$

4) Overall Utility: The overall utility $Utility(G, S_0, S_1)$ combines three factors, i.e., the benefits gained from the satisfaction of target goals, the losses for possible failures, and the cost. If G is achieved with a probability $P(G)$, a benefit $Benefit(S_1)$ can be gained with the goal specification S_1. If G fails with a probability $1 - P(G)$, we assume a loss equivalent to a cancellation, i.e., the debtor compensates the creditor for the cancellation of their commitments. For a commitment cancellation, the creditor may request a compensation between 0 and a maximum compensation $maxCom$. Therefore, the loss can be estimated to $\frac{maxCom}{2}$ on average. Because cost is independent of the probability of success, the utility can be estimated as follows:

$$Utility(G, S_0, S_1) = P(G) \times Benefit(S_0, S_1) - (1 - P(G)) \times \frac{maxCom}{2} - Cost(S_0, S_1)$$

As a special case of adaptation, commitment cancellation means that no new goal specification (i.e., S_1) will be activated for G, so there is no anticipated benefit or loss for goal satisfaction. In this case, the utility can be estimated with the following equation.

$$Utility(G, S_0, null) = -Cost(S_0, null)$$

C. Commitment Renegotiation

To facilitate commitment renegotiation, a set of candidate adaptation strategies are generated for a debtor agent based on its goal model. An adaptation strategy consists of a goal specification, a proposal for commitment changes and corresponding compensations (see Definition 3 in Section III-A). As the compensation part of an adaptation strategy is to be negotiated in the renegotiation process (see Algorithm 1), a candidate adaptation strategy includes only a goal specification and a proposal for commitment changes. The goal specifications for the root goal of an agent can be enumerated based on the goal model. The commitment changes can be enumerated in a certain range and with a predefined pace. For example, for a commitment of "food served within 40 minutes", a series of commitment changes can be derived with the time prolonged to 45, 50, 55, 60 minutes respectively, assuming that the maximum time is 60 minutes and the pace is 5 minutes. The set of candidate adaptation strategies is derived from all possible goal specifications for the root goal by combining different proposals of relaxed or improved commitments. In addition, a special adaptation strategy with null goal specification (i.e., cancelling the established commitment) is also included in the set of candidate adaptation strategies.

On the basis of the utility estimation for each candidate strategy, the debtor can follow a procedure (see Algorithm 1) to renegotiate a commitment with its creditor. The renegotiation process can be initiated by the debtor or the creditor. If it is the former, it is usually the case that the commitment is at risk and the debtor is trying to persuade the creditor to accept a relaxed commitment with proper compensation (see the first case in Section II). In this case, the debtor can start by offering zero compensation and increase the offer gradually. If it is the latter, it is usually the case that the creditor expects the debtor to improve the initial commitment with proper compensation (see the second case in Section II). In this case, the debtor can start with a high compensation and decrease the compensation gradually. The principle behind the procedure is to reach an agreement on commitment changes with the creditor that can maximize the cost-benefit of the debtor. In the renegotiation process, not all the candidate strategies need to be considered by the debtor for negotiation with the creditor. Rather, the negotiation can just focus on high utility strategies.

The selection algorithm takes as input a series of candidate strategies $Stra$, an initiator flag $initiator$, a maximal compensation $maxCom$, and a compensation negotiation pace parameter $pace$. Each strategy in $Stra$ is represented by a 4-tuple ($commit$, $prob$, $utility$, $compen$), where $commit$ is the involved commitment, $prob$ is the estimated probability of success, $utility$ is the estimated utility, and $compen$ is the current compensation in the renegotiation process. The algorithm returns an adaptation strategy that involves an agreed commitment change with possible compensation or keeps the current commitment.

The algorithm first initializes the compensation of each candidate strategy to 0 or $maxCom$ according to whether the renegotiation process is initiated by the debtor or the creditor (Lines 2-8). Then it performs an iterative process to find a best strategy (Lines 9-34). In each iteration, it first finds a strategy with the highest value including the estimated utility and the current compensation (Lines 10-21). Note that the compen-

TABLE II
CANDIDATE ADAPTATION STRATEGIES

Strategy	Specification	Commitment	Utility
St1	unchanged	time prolonged to 55 minutes	3
St2	{g8}	unchanged	2
St3	{g8}	time prolonged to 55 minutes	6
St4	N/A (cancelled)	N/A (cancelled)	-13

TABLE III
RENEGOTIATION PROCESS

Iteration	Strategy	Utility	Compensation	Value
1	St3	6	0	6
2	St3	6	-2	4
3	St1	3	0	3
4	St2	2	0	2

sation paid by the creditor to the debtor is conditional, i.e., the debtor can only obtain the compensation when the goal is successfully achieved with the new commitment (Lines 14-15). If the best strategy returned retains the current commitment, then no further negotiation is needed (Lines 22-23); if the compensation of the best strategy has exceeded $maxCom$, the negotiation process also ends with the current strategy (Lines 24-26); otherwise, the debtor proposes the current strategy to the creditor (Line 28). If the creditor accepts the proposal, then the current strategy is selected (Lines 29-30). If not, the compensation of the current strategy is decreased with a given pace (Lines 31-32). For a renegotiation process initialized by the debtor, such a decrease actually increases the compensation paid to the creditor.

Let us illustrate the renegotiation process with an example. In the first case discussed in Section II, the restaurant (RES) estimates that the current plan with the goal specification $\{t1, t2, t4, t5, t6, t8, g7\}$ cannot be completed on time. Assume that RES considers the adaptation strategies presented in Table II. For simplicity not all the candidate strategies are shown here. The utility estimation is based on the following parameters: the benefit for a successful food serving is 30 dollars; the maximal compensation (i.e., $maxCom$) is 100 dollars; the probabilities of St1, St2 and St3 are 0.85, 0.9, 0.95, respectively; the costs of St1, St2, St3, and St4 are 15, 20, 20, 13 dollars respectively. The utilities of St1, St2, St3 and St4 thus can be estimated respectively as: $0.85 \times 30 - 0.15 \times 50 - 15 = 3$, $0.9 \times 30 - 0.1 \times 50 - 20 = 2$, $0.95 \times 30 - 0.05 \times 50 - 20 = 6$, -13.

Based on these candidate strategies, an iterative renegotiation process can be conducted as shown in Table III, in which each row denotes the best candidate strategy selected in that iteration. As the renegotiation process is initiated by RES (the debtor), the compensation of each candidate strategy is initialized to 0. In the first iteration, RES proposes St3 to the customer Bob with no compensation and the value 6 is the highest. As Bob refuses the proposal, the compensation of St3 is decreased to -2 assume that the *pace* is 2. Thus in the second iteration, RES proposes St3 to Bob again with the updated compensation as its value 4 is still the highest. Bob still refuses the proposal and the compensation of St3 is decreased to -4. In the third iteration, RES proposes St1 to Bob with no compensation, which yields the highest value at

Algorithm 1 Commitment Renegotiation

```
1: procedure RENEGO(Stra[ ], initiator, maxCom, pace)
2:     for (i = 0; i < Stra.length; i++) do
3:         if initiator == Debtor then
4:             Stra[i].compen = 0
5:         else
6:             Stra[i].compen = maxCom
7:         end if
8:     end for
9:     while true do
10:        maxValue = −∞
11:        for (i = 0; i < Stra.length; i++) do
12:            if Stra[i].compen ≤ 0 then
13:                value = Stra[i].utility + Stra[i].compen
14:            else
15:                value = Stra[i].utility + Stra[i].prob × Stra[i].compen
16:            end if
17:            if value > maxValue then
18:                best = i
19:                maxValue = value
20:            end if
21:        end for
22:        if Stra[best].commit == unchanged then
23:            return Stra[best]
24:        else if Stra[best].compen ≤ −maxCom then
25:            Stra[best].compen = −maxCom
26:            return Stra[best]
27:        end if
28:        propose Stra[best] to the creditor
29:        if Stra[best] is accepted by the creditor then
30:            return Stra[best]
31:        else
32:            Stra[best].compen− = pace
33:        end if
34:    end while
35: end procedure
```

3. Bob still refuses the new proposal and the compensation of St1 is dropped to -2. In the fourth iteration, RES proposes St2 to Bob with no compensation and the highest value is 2. As St2 does not change the current commitment to Bob, the renegotiation process ends with St2 with no compensation as the selected strategy. Notice that in each of the above iterations the renegotiation process can end with the proposed strategy if Bob accepts it.

IV. EXPERIMENTAL STUDY

Our experimental study is based on a simulated socio-technical system developed using AnyLogic [6]. In the experiment, we simulate two kinds of adaptation scenarios with various probabilistic uncertainties. Based on the results, we evaluate the effectiveness of our framework and discuss some related issues.

A. Subject System

The subject system used in our experimental study is the food ordering system introduced in Section II. Three kinds of agents, i.e., Restaurant, Customer and Delivery Man, are involved in this system.

In the normal case, a restaurant makes a food serving commitment to a customer who orders food from the restaurant, then gets the food prepared and packed, and delegates a delivery man to deliver the food by truck or motorcycle. However, various uncertainties with environmental and intentional factors may place the initial commitment at risk or require the restaurant to provide better service. For example, traffic conditions vary at different time of the day or the delivery man may lose his way. These uncertainties may prevent on-time delivery of the food. On the other hand, the customer may ask the restaurant to serve the food earlier than agreed upon for personal reasons. In this case, the restaurant may insist on the initial commitments. However, if the restaurant chooses to change current arrangements, it can increase its benefits and also deliver better customer service. In both cases, the commitment between the restaurant and the customer needs to be renegotiated.

In commitment renegotiation, different customers may adopt quite different attitude to the compensation obtained from or paid to the restaurant. When the restaurant proposes to prolong the food serving time, some customers are more tolerant to this commitment change, while others are less tolerant and may charge a high compensation. On the other hand, when the restaurant is asked to speed up delivery, some customers appreciate the extra effort.

For a successful delivery, the restaurant can earn the price of the transaction. For an unsuccessful delivery, the restaurant earns nothing, and may actually have to compensate the customer for reneging on its commitments.

B. Adaptation Scenarios

The behaviors of the three kinds of agents in our experiment are modelled by statecharts in AnyLogic. Related agents are connected and interact by exchanging messages. Our adaptation mechanisms including utility estimation and commitment renegotiation are integrated with the behavioral models of different agents using AnyLogic APIs.

In the experiment, we simulate two kinds of adaptation scenarios introduced in Section II, i.e., risk of failure and opportunity of improvement. The first is triggered by estimated delays of delivery caused by traffic conditions, and, while the second is triggered by customer requests for serving food earlier than anticipated. The following uncertainties with environmental and intentional factors are accounted for in the simulation:

1) Traffic Condition and Speed: Traffic conditions in our experiment are represented by a random number between 0 and 1 (0 for the most congested and 1 for no traffic). We assume that each customer session has its own traffic condition variable. The influence of the traffic condition on speed of delivery is reflected in Fig. 2. It can be seen that both truck and motorcycle decrease speed when there is congestion. Trucks are faster than motorcycles, except when there is congestion.

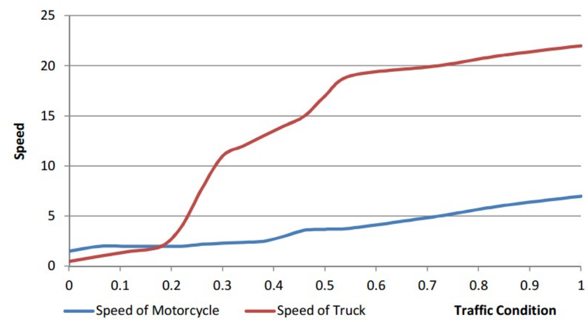

Fig. 2. Change of Speed with Traffic Condition

Since the speed of the delivery depends on traffic conditions, we calculate the mean value of the time for traveling through a road segment. Further we assume that the actual time for a delivery to pass a road segment follows a normal distribution, which is simulated by AnyLogic's random number generator for the normal distribution. To avoid the time being less than or equal to zero, we use a fairly small standard deviation, which ensures that the probability of the time being less than or equal to zero is quite small (less than 0.01%). And when a value less than or equal to zero is generated for the time, our implementation will reset it to a minimum time t_{min} ($t_{min} > 0$). With this setting, about half of the orders can be delivered in time without any adaptation, while the remaining orders require proper adaptation actions otherwise delivery time will be delayed.

According to the behavior models of the agents, a delivery man shall pass several road segments before reaching the customer. After passing each road segment, the delivery man evaluates the probability of on time delivery and report an adaptation request to the restaurant if the risk of being late is above 50%.

2) Customers' Requests for Serving Food Earlier: In our experiment, 40% customers ask the restaurant to serve the food earlier than on the appointed time. A delivery man checks possible requests from the customer on the way and reports an adaptation request to the restaurant upon customer request.

3) Customer Attitudes to Compensation: There are four levels of customer tolerance to delay with an expected compensation of 3, 2, 1, 0.5 dollars per minute respectively. Similarly, there are four levels of customer appreciation for the restaurant's efforts to serve food earlier with an expected compensation of 2, 1, 0.5, 0.3 dollars per minute respectively. We thus simulate 16 different categories of customers with different levels of tolerance to delay or appreciation for faster service. In the experiment, customers categories have equal probability.

C. Results

The simulations are run on a computer with Intel Core i3 Duo M370 and 4G memory. In the experiment, the simulated scenarios are executed under three different conditions, i.e., without adaptation, with primitive adaptation (static commitments), and with our adaptation framework (evolving commitments), respectively. Note that primitive adaptation can be seen as a special case in our framework that considers only adaptation strategies without commitment changes. Under each condition, the scenarios are executed three times and in each execution 1000 customer instances are created. Therefore, we have a total of 3000 customer sessions executed under each

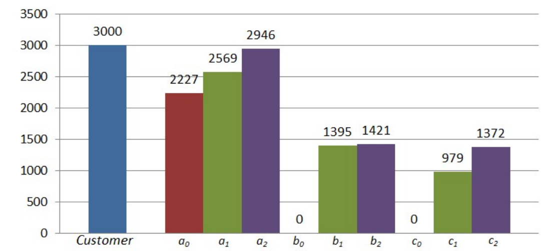

Fig. 3. Experiment Results: Successful Orders and Successful Adaptations

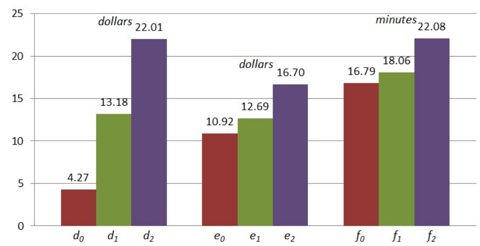

Fig. 4. Experiment Results: Cost-Benefit Analysis and Time

TABLE IV
ADAPTATION CASES

No.	Traffic	Appr	Best Strategy				Compen	Result
			S	T	P	U		
1	0.8	0	-	-	-	-	-	S
2	0.8	1	N	-	100%	31	-	S
3	0.8	2	N	-4	98.7%	29.8	7	S
4	0.3	0	-	-	-	-	-	F
5	0.3	1	Y	-	73.0%	-5.3	-	F
6	0.3	2	N	22	100%	31	-12	S
7	0.3	0	-	-	-	-	-	F
8	0.3	1	Y	-	73.0%	-5.3	-	S
9	0.3	2	Y	4	98.8%	18.6	-9	S
10	0.24	0	-	-	-	-	-	F
11	0.24	1	Y	-	19.0%	-53.8	-	F
12	0.24	2	Y	6	98.0%	17.5	-12	S
13	0.07	0	-	-	-	-	-	F
14	0.07	1	N	-	0	-9	-	F
15	0.07	2	C	-	0	-9	-50	F

condition and in each customer session one order is placed with the restaurant.

All the customer sessions are executed with exactly the same configurations of related parameters: the distance from the restaurant to customer's home is 80; the probability of choosing truck (motorcycle) initially is 50%; the price of each order is 40; the max compensation is 100; and the food serving time in initial commitments is 30.

To evaluate the effectiveness of our framework, we collect and compare the following six key performance indicators under three different conditions: **a)** the number of customers who get their food delivered on time; **b)** the number of order sessions where some kind of adaptation occurs; **c)** the number of order sessions where adaptation occurs and the target goal is satisfied; **d)** the average value (with both cost and benefit considered) that the restaurant earns from each customer (including both successful and failed orders); **e)** the average cost that the restaurant spends on each successful order, including compensation paid to the customer; **f)** the average food service time for each successful order.

The experiment results of these measurements are presented in Fig. 3 and 4, where the label under each bar denotes one of the six key performance indicators listed above. The subscripts of each label indicate execution conditions: "0" denotes execution without any adaptation; "1" denotes execution with primitive adaptation; and "2" denotes execution with our framework.

From Fig. 3, it can be seen that the success rate of customer orders (a/3000) is significantly improved by adaptation (from 74.2% to 85.6%) and further improved by adaptation with evolving commitments (from 85.6% to 98.2%). Moreover, the success rate of adaptation (c/b) rises from 70.2% to 96.6% by incorporating evolving commitments in adaptation.

From the cost-benefit analysis in Fig. 4, it can be seen that the restaurant earns much more from each order by adaptation with evolving commitments (22.01 dollars) compared to primitive adaptation (13.18 dollars) and no adaptation (4.27 dollars). We can also see that the cost (16.70 dollars compared with 12.69 dollars and 10.92 dollars) and time spent on each successful order (22.08 minutes compared with 18.06 minutes and 16.79 minutes) rise significantly after incorporating evolving commitments into adaptation. The rise in cost and time is caused by additional cost and time spent for adaptation and compensation paid to customers. These additional cost and time are rewarded by more successful orders and also compensation obtained from customers for serving food earlier.

To further analyze how evolving commitments are incorporated in adaptation decisions, we investigate some typical adaptation cases taken from the execution logs. Table IV provides five groups of cases and each group includes three cases that are executed with the three different kinds of approaches under the same situations (e.g., traffic condition). For each case, we describe the traffic condition, the approach used (0 for no adaptation, 1 for primitive adaptation, and 2 for adaptation with evolving commitments), the best strategy selected, the compensation (positive/negative value means compensation earned from/paid to the customer), and the final goal satisfaction result ("S" for success, "F" for failure). And for each best strategy, we provide the specification ("N" for no change, "Y" for switching to a new one, and "C" for cancellation), the commitment change (positive/nagative value means the time prolonged/shortened), the estimated probability of success and utility.

From the first group (Cases 1-3), it can be seen that when traffic conditions are good, delivery can be achieved on time with little need for adaptation. In cases where the customer asks for early delivery, evolving commitment strategies improve substantially restaurant benefits.

The second group (Cases 4-6) presents three cases where the risk of being late is quite high due to bad traffic conditions. For the case of primitive adaptation (Case 5), the restaurant chooses to switch to an alternative specification but the order still fails eventually. Considering evolving commitments (Case 6), the restaurant decides to propose a new commitment with a prolonged food serving time (22 more minutes). After negotiation, the customer accepts the new commitment with a compensation of 12 dollars paid by the restaurant.

In Case 9 of the third group (Cases 7-9), which is executed with the same situation, the customer requests a rather high compensation for the same commitment change (22 more minutes), so the restaurant chooses another strategy with a much shorter time extension and a lower compensation.

In the fourth group (Cases 10-12), the traffic gets worse and merely switching specifications cannot ensure success. Similar to Case 9, in Case 12 the restaurant chooses an adaptation strategy involving both specification switching and commitment changes.

When the traffic gets even worse in the fifth group (Cases 13-15), the probability of success and utility are quite low even after adopting the best adaptation strategy, so the restaurant chooses to cancel the order with proper compensation (Case 15). In this case, other adaptation strategies are even less cost-beneficial due to additional cost and low probability of success.

From all these cases, it can be seen that by incorporating evolving commitments into adaptation the restaurant can always adopt better adaptation strategies by leveraging customer tolerance to moderate delay and appreciation for the restaurant's efforts towards a better service.

D. Discussion

Existing self-adaptation approaches focus on seeking "better" specifications that can alleviate the risk of failing to satisfy committed requirements or improve the quality of provided services. Unlike these approaches, our framework incorporates evolving commitments into adaptation strategies with cost-benefit analysis and commitment renegotiation so as to: (1) avoid vain attempts to satisfy existing commitments by trying different alternatives; (2) choose the most cost-beneficial strategy with acceptable commitment relaxation and compensation; and (3) respond actively to creditor requests for improved commitments by fully tapping available opportunities.

Our experimental study is based on a simulated STS with a series of parameter configurations. The simulated scenarios and related parameters may not completely reflect realistic settings for an STS. For example, the range and frequency of changes to traffic conditions are key parameters that influence the effectiveness of self-adaptation approaches. If traffic conditions change little or slowly, few adaptations are triggered; but if the traffic condition changes too much and too fast, selected adaptation strategies are more likely to fail due to poor estimation of the probability of success and utility. However, by incorporating evolving commitments, our approach can achieve higher success rate and improved overall utility than primitive adaptation approaches.

There are also limitations within our approach: First, apart from direct compensation paid to or earned from the creditor, commitment changes may also implicitly influence the satisfaction of the creditor and the reputation of the debtor. For example, if the restaurant cancels orders for the same customer several times, it is likely that the customer will move on, even though he was compensated for each cancellation. This kind of outcome cannot be captured by our current framework. For this problem, we need a proper mechanism for representing reputation and trust and for quantifying the influences of commitment changes on these.

Second, cascading commitment changes and renegotiation are not well supported in our proposal. Cascading commitment changes and renegotiation occur when a debtor agent is also a creditor, i.e., depending on other agents for some goals. For example, when the restaurant makes the promise of early delivery to the customer, a commitment renegotiation may also be required between the restaurant and the delivery man. The renegotiation procedure for cascading commitment changes can be much more complex, because different commitments may depend on each other. To cascade commitment changes, a more comprehensive renegotiation procedure with a top-down or a bottom-up change propagation process is required.

Third, as a utility-based approach, our framework only considers stakeholder preferences for different high-level goals. This may lead to a high-risk strategy being selected for its higher utility. More comprehensive preference representation and reasoning mechanisms that support the tradeoff between utility and risk tolerance should be incorporated to allow, for example, a stakeholder to choose a strategy that is less cost-beneficial but has higher probability of success.

V. Related Work

Requirements goal models capture stakeholder goals, alternatives for achieving high-level goals, and their different respective contributions to softgoals. Goal modeling techniques such as i* [13] and Tropos [14] further support agent-oriented modeling of social dependencies. These techniques have been widely used to support requirements reasoning [9], [8], [11], monitoring [15], diagnosing [16], and decision-making for purposes of self-repair [10], [17] and self-optimization [18].

Some researchers further explore the adaptation of STSs involving social interactions with the support of goal-oriented requirements modeling and reasoning. Dalpiaz et al. [17] proposed a requirements-based conceptual architecture for runtime self-reconfiguration, which is structured as a set of interacting components connected through a Monitor-Diagnose-Compensate (MDC) cycle. Their architecture supports requirements-based adaptation for STSs by using Tropos [14] to model social dependencies between stakeholders. Moreover, they enriched Tropos models with context-dependent goal decompositions, goal commitments with time limits, and domain assumptions. Our previous work [2] proposed a distributed, stateful, goal-based monitoring and self-repairing framework for STSs. The framework creates and maintains live instances of goal state machines for each agent and manages interactions between different goal state machines. The isomorphic nature of goal state machines and event-driven interactions facilitate decentralized requirements monitoring and repairing that are demanded by STSs.

STSs are typical open systems, characterized by interactions among autonomous and heterogeneous participants [3], [5]. As an explicit and verifiable abstraction for social interactions, commitments are thought to be better suited to open systems than dependencies. To support reasoning about the suitability of interaction protocols and agent specifications for particular agent goals, Chopra et al. [4] formalized the semantic relationship between an agent specification (in terms of goals) and protocols (in terms of commitments). Based on the work, Dalpiaz et al. [3] further formalized a participant's strategy for particular goals (in terms of goals, plans and commitments) and proposed a conceptual model and framework for adaptive agents in open systems. The adaptation framework provides generic criteria for strategy selection and the optimization of commitments and capabilities to actual agents and plans. Their approach incorporates social interactions into adaptation strategies of adaptive agents, but considers neither cost-benefit estimation nor commitment changes.

Self-adaptation and agent negotiation have also been investigated in the area of multi-agent systems. Weyns et al. [19] showed how multi-agent systems and agent-oriented software engineering can help master the complexity of self-adaptation for modern distributed software systems such as Web-based e-commerce systems. Multi-agent systems help alleviate challenges such as decentralized coordination of self-adaptation in a distributed setting. To reach agreements, different agents can conduct automated negotiation with each other through an iterative process of making offers [20]. The works on this line focus on supporting automated negotiation and decision making for self-interested agents or cooperative agents with techniques like game-theory, economics based techniques, and logical based mechanisms for argumentations [21]. Some works also focus on the tradeoff strategy of multiple decision variables [20]. These works do not associate agent negotiation with requirements goals, social commitments, or commitment changes in adaptation.

VI. Conclusions and Future Work

Evolving commitments allow an STS to reconsider the protocols through which it fulfills its requirements. In this paper, we propose a decision-theoretic self-adaptation framework supporting evolving commitments. The effectiveness of the framework has been evaluated with an experimental study on a simulated STS. The contributions of our proposal lie in the following aspects. First, we propose the concept of evolving commitments for self-adaptation in STSs, which allows agents to exploit both plan reconfiguration and commitment changes in their adaptation strategies. Second, we provide an approach for estimating the cost-benefit of candidate adaptation strategies based on probabilistic utility functions. Third, we provide a procedure for a debtor agent to negotiate commitment changes and possible compensation with other agents. Finally, we have designed, implemented and used a simulation-based evaluation infrastructure for STSs. Such infrastructures are essential for the engineering of STSs because of the complexity and unpredictability of social settings.

Of course, there are limitations to our proposal. Our models only capture and can cope with certain kinds of uncertainty and they could be more elaborate. The same can be said about the statechart models of agents used in the simulation as well as our renegotiation process. Nevertheless, our results suggest that even with such simple models there are benefits in including evolving commitments in the adaptation mechanism of an STS. Our future work will integrate our proposal with a multi-agent development framework such as JADE [22] and will conduct case studies on real-life socio-technical systems.

Acknowledgment

This work is supported by National Natural Science Foundation of China under Grant No. 61361120097, National High Technology Development 863 Program of China under Grant No. 2013AA01A605.

References

[1] I. Sommerville, D. Cliff, R. Calinescu, J. Keen, T. Kelly, M. Z. Kwiatkowska, J. A. McDermid, and R. F. Paige, "Large-scale complex IT systems," *Commun. ACM*, vol. 55, no. 7, pp. 71–77, 2012.

[2] L. Fu, X. Peng, Y. Yu, J. Mylopoulos, and W. Zhao, "Stateful requirements monitoring for self-repairing socio-technical systems," in *RE*, 2012, pp. 121–130.

[3] F. Dalpiaz, A. K. Chopra, P. Giorgini, and J. Mylopoulos, "Adaptation in open systems: Giving interaction its rightful place," in *ER*, 2010, pp. 31–45.

[4] A. K. Chopra, F. Dalpiaz, P. Giorgini, and J. Mylopoulos, "Reasoning about agents and protocols via goals and commitments," in *AAMAS*, 2010, pp. 457–464.

[5] N. Desai, A. Mallya, A. Chopra, and M. Singh, "Interaction protocols as design abstractions for business processes," *IEEE Trans. Softw. Eng.*, vol. 31, no. 12, pp. 1015 –1027, 2005.

[6] X. Technologies, "Anylogic help," http://www.xjtek.com/anylogic/help/.

[7] A. K. Chopra, F. Dalpiaz, P. Giorgini, and J. Mylopoulos, "Modeling and reasoning about service-oriented applications via goals and commitments," in *CAiSE*, 2010, pp. 113–128.

[8] E. Letier and A. van Lamsweerde, "Reasoning about partial goal satisfaction for requirements and design engineering," in *SIGSOFT FSE*, 2004, pp. 53–62.

[9] S. Liaskos, S. McIlraith, S. Sohrabi, and J. Mylopoulos, "Representing and reasoning about preferences in requirements engineering," *Requir. Eng.*, vol. 16, no. 3, pp. 227–249, 2011.

[10] Y. Wang and J. Mylopoulos, "Self-repair through reconfiguration: A requirements engineering approach," in *ASE*, 2009, pp. 257–268.

[11] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani, "Reasoning with goal models," in *ER*, 2002, pp. 167–181.

[12] R. Sebastiani, P. Giorgini, and J. Mylopoulos, "Simple and minimum-cost satisfiability for goal models," in *CAiSE*, 2004, pp. 20–35.

[13] E. S. K. Yu, "Towards modeling and reasoning support for early-phase requirements engineering," in *RE*, 1997, pp. 226–235.

[14] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An agent-oriented software development methodology," *Auton. Agent Multi Agent Syst.*, vol. 8, no. 3, pp. 203–236, 2004.

[15] A. R. Dingwall-Smith, "Run-time monitoring of goal-oriented requirements specifications," Ph.D. dissertation, University of London, 2006.

[16] Y. Wang, S. A. McIlraith, Y. Yu, and J. Mylopoulos, "An automated approach to monitoring and diagnosing requirements," in *ASE*, 2007, pp. 293–302.

[17] F. Dalpiaz, P. Giorgini, and J. Mylopoulos, "An architecture for requirements-driven self-reconfiguration," in *CAiSE*, 2009, pp. 246–260.

[18] X. Peng, B. Chen, Y. Yu, and W. Zhao, "Self-tuning of software systems through dynamic quality tradeoff and value-based feedback control loop," *J. Syst. Softw.*, vol. 85, no. 12, pp. 2707–2719, 2012.

[19] D. Weyns and M. P. Georgeff, "Self-adaptation using multiagent systems," *IEEE Softw.*, vol. 27, no. 1, pp. 86–91, 2010.

[20] P. Faratin, C. Sierra, and N. R. Jennings, "Using similarity criteria to make issue trade-offs in automated negotiations," *Artif. Intell.*, vol. 142, no. 2, pp. 205–237, 2002.

[21] S. Kraus, "Automated negotiation and decision making in multiagent environments," in *EASSS*, 2001, pp. 150–172.

[22] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa, "JADE: A software framework for developing multi-agent applications. lessons learned," *Inform. Softw. Tech.*, vol. 50, no. 1-2, pp. 10–21, 2008.