An Agent Based Approach for the Decentralised Compensation of Business Processes

Tim Lessner

University of the West of Scotland, School of Computing
Paisley
timlessner@lesshome.net

Abstract. An automated execution of business processes, composed of Web Services, also requires a reliable error handling and in the case of failures at least parts of a process need to be recovered. One way to enable a backward oriented recovery is compensation which allows to semantically undo effects even "long" time after the process –a long running transaction– commits. Usually, compensation as for example applied in BPEL (Business Process Execution Language), requires that all compensation steps associated with the business process perform successfully. Also, the conditions for a compensation are not considered and it is not dynamic. In some business scenarios, e.g. a complex production process involving several partners, compensation is a commitment under certain conditions. These conditions in turn can be dynamic and interdependent and often a lot of human effort might be required to perform compensation. Agents in turn, can enable a decentralised compensation in which agents negotiate a commitment to find a proper compensation strategy. Especially in highly dynamic environment this enables on the one hand a dynamic compensation, and on the other it can support human decision finding in case of failure.

Keywords. Compensation, Business Processes, Business Transactions, Transaction Management

1 Introduction

Nowadays, the automated execution of business processes, for instance processes represented in BPEL (Business Process Execution Language), usually relies on a centralised architecture. Beside this, approaches for a decentralised execution of web services exist. In the field of Mobile Ad Hoc Networks (MANETs), for instance, a decentralised execution (orchestration) is required because based on the unreliable nature of mobile networks, and the mobile devices as well, no guarantee for the availability of any instance is given; devices join and leave the network in an unpredictable manner.

This discussion paper focuses on decentralised error handling. Concerning error handling the compensation concept gained much consideration the last years. Compensation allows semantically undoing effects even a "long" time after the

Dagstuhl Seminar Proceedings 10021
Service-Oriented Architecture and (Multi-)Agent SystemsTechnology
http://drops.dagstuhl.de/opus/volltexte/2010/2552

transaction commits which is especially important for long running transactions, like business processes. The assumption is that many business activities allow for a definition of an corresponding activity which semantically undoes the effects. E.g. the cancellation of a hotel compensates the corresponding booking. More details about compensation are provided in section 3. The idea is also about an agent based approach in which agents start a negotiation among themselves using a mobile agent as mediator. The objective of their negotiation is to adjust rules and conditions under which compensation takes place or to form coalitions in a way that the most of them are able to perform their corresponding compensation, of course in the sense of business expectations.

The remainder of the paper is as follows. The next section briefly describes the advantages if agents are used to implement web services. Section three briefly introduces compensation. Based on an example, section 4 introduces the idea in more detail. Section 5 treats about the expected benefits whereas the last section summarises open issues.

2 Agents and Web Services

In general agents behave proactive, they do not only know about themselves, they are meta-level aware, they are able to understand ontologies, agents act autonomously, they handle intentional and they also deliberately cooperate together [1].

It is worth noting that the web service concept describes a mechanism which enables the usage of functionality, i.e. software components, based on internet protocols, whereas agents are a general concept for the implementation of software and they represent autonomous components that ship with specific characteristics like the ones mentioned above.

According to Buhler and Vidal [1] "agents can be viewed as independent applications that provide services to one another through loosely coupled, asynchronous message exchange." Even if they have the ability to learn and to optimise their behaviour, web service technologies like XML are still required in order to exchange data across local boundaries and between heterogeneous systems.

For a thorough discussion concerning web services and agents see [2], for workflow systems and Multiagent Systems (MAS) see [3,4].

3 Compensation

As mentioned, compensation of business processes gained much consideration the last years. Compensation allows semantically undoing effects even a "long" time after the transaction commits which is especially important for long running transactions, like business processes.

Although compensation relaxes isolation, which in turn possibly leads to cascading compensation, a relaxed isolation is often indispensable in loosely or disconnected systems since a too strong isolation leads to resource blocking which

should be avoided, particularly to increase the performance, and thus the overall progress.

An important issue is that compensation is only successful if all required compensation operations, hence the so called Spheres of Joint Compensation (compensation sphere) executes successfully (see [5,6] (chapter 7). So, if only one of them fails a re-execution or an escalation, e.g. notification of the administrator, is required. Whereas the first one is generally an adequate solution, the latter one requires a human intervention which should be reduced due to costs.

4 Idea

Compensation requires that all compensation steps are to be successfully executed. And, as briefly sketched out, services are passive and accordingly to this they are not able to automatically adjust rules and (or) conditions under which they perform their task, i.e. compensation in this case.

Here the idea is that an automatic adjustment of conditions during the compensation's execution could lead to a new situation and partners that heretofore rejected compensation might be inclined to execute the compensation under new, different conditions.

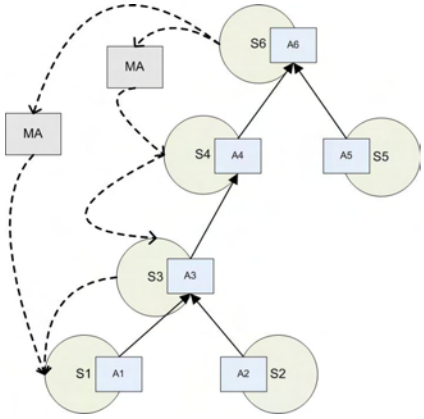

Fig. 1. Compensation steps

The tree in figure 1 presents the required compensation steps for a specific business process which is not of interest here. The five compensation services (operations) $(S1 - S5)$, each implemented by an agent $(A1 - A5)$ represent the compensation steps. Service $S6$, i.e. agent $A6$, is the initiator. The MA rectangle is a Mobile Agent, also referred to as Mediator. Agents may reside at different physical nodes or one node may host several agents.

Once compensation is required, a mobile agent is created and equipped with the rules and conditions under which the compensation should take place. Since

compensation starts in reverse order (bottom-up in this example), the mobile agent MA contacts $A1$ first, and subsequently $A2$. No assumptions are made how MA reaches node 1. Probably the agent follows the path in the tree, but a direct connection is not ruled out. The graph depicts the structure of the compensation and not the physical connections between nodes. Also, it is possible that MA just mediates and has not to move to different nodes. To use a mobile agent increases flexibility and reduces message overhead since the entire state can be transferred.

If $A1$ and $A2$ agree to compensate, MA contacts $A3$. If $A3$ is inclined to compensate, MA contacts agent $A4$, if $A4$ is inclined too, $A5$ is contacted and if $A5$ also agrees compensation is successful.

Obviously, this procedure does not reflect the more important erroneous situation in which one agent is declined to execute the compensation due to its own decision autonomy.

Since agents, like business, follow their own interests first a decision essentially depends on the benefit of taking this concrete decision. But, this also means that it could be possible to convince an agent that a decision is beneficial for him.

If, for example, $A3$ rejects compensation the mobile agent can initiate a negotiation. First of all this requires $A3$ to expose his reasons for the decision. Once $A3$ tells the MA the reasons, MA can verify whether the reasons depend on the outcome of $A1$ or $A2$, and, if so MA can mediate with the aim to find a resolution that fits the expectations of $A3$. If this is not feasible the agents may compromise, or it is also imaginable that $A3$ has more information than $A1$ and $A2$, e.g. the knowledge that compensation makes progress impossible at a later stage. Additionally, a mechanism similar to approaches which require a majority, like for example in atomic commitment protocols can be applied.

The MA can be viewed as an arbitration. He judges under the consideration of facts, rules and interests in order to find the "best" solution.

In summary the following may be the result of negotiation:

1. Eventually, the values which cause the unwilling agent to reject compensation can be adjusted.
2. A compromise is possible, this also means to change the conditions for the unwilling one.
3. The unwilling agent has entitled reasons for his decision and it is more valuable for all participants to accept and follow his decision.
4. A majority conclude that a specific compensation has no longer any effect on the final outcome, thus, the unwilling agent can be ignored.
5. Other agents provide alternative(s) which made the unwilling one obsolete.
6. No solution can be found and escalation, e.g. human intervention, is required.

5 Expected Benefits

– **Increased autonomy:** Service autonomy is a general characteristic in SOA. Agents enable or at least increase autonomy due to their ability to autonomously take a decision based on their interests. Without such a concept

it is hard to achieve this level of autonomy. The consequence however is, that situations may arise in which the ability to autonomously decide worsens the progress in process execution.
- **Detection of compensation spheres:** Usually a compensation procedure, e.g. only a separate activity or a complete process, is defined in advance. To dynamically detect compensation spheres is therefore a worthwhile objective. It is questionable if agents are able to perform this. But, the idea of negotiation increases the options how compensation can be performed more dynamically and autonomously during runtime which seems to be a valuable step.
- **Prevention from recurring failures:** Another benefit is the ability that involved partner services could learn from the misconduct of their partners. As a consequence agents can no longer cooperate with them in future. Thus, an automatic detection of unreliable partners can be achieved.
- **Reduced costs:** If it is possible to reduce the number of escalations which require human intervention, costs usually should decrease.
- **Failure tolerance:** Traditionally, compensation requires atomicity. This approach, however, relaxes atomicity since it is possible that not all steps have to perform successfully.
- **Determination of economic compensation:** In some situations partners will demand a form of economic compensation, e.g. payments to cancel orders. Assumed that specific rules for an economic compensation can be exploited by agents a more (semi)automatic processing might be possible.
- **Simplifies human intervention:** Even if agents fail to find a satisfying solution, they could at least gather more detailed information about the circumstances which lead to the escalation. This might significantly ease the manual work which usually has to be performed in order to resolve the conflict.

6 Conclusions

This paper is a purely informal description of how agents can be used to reduce or even eliminate human effort in order to perform a compensation of business processes. It especially relies on the idea of a dynamic commitment between agents.

References

1. Buhler, P., Vidal, J.M.: Enacting BPEL4WS specified workflows with multiagent systems. In: Proceedings of the Workshop on Web Services and Agent-Based Engineering. (2004)
2. Huhns, M.N.: Agents as web services. IEEE Internet Computing **6** (2002) 93–95
3. Singh, M.P., Huhns, M.N.: Multiagent systems for workflow. In: INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS IN ACCOUNTING, FINANCE AND MANAGEMENT. (1999) 8–105

4. Dang, J., Huang, J., Huhns, M.N.: Workflow coordination for service-oriented multi-agent systems. In Durfee, E.H., Yokoo, M., Huhns, M.N., Shehory, O., eds.: AAMAS, IFAAMAS (2007) 249
5. Leymann, F.: Supporting business transactions via partial backward recovery in workflow management systems. In: BTW. (1995) 51–70
6. Leymann, F., Roller, D.: Production Workflow - Concepts and Techniques. PTR Prentice Hall (2000)