CLAM: Cross-Layer Adaptation Management in Service-Based Systems*

Asli Zengin
Supervised by: Marco Pistore

Fondazione Bruno Kessler – IRST, Trento, Italy
{zengin,pistore}@fbk.eu

Abstract. Service-based systems (SBS) have a complex layered structure where the service-based application (SBA) is implemented through a composition of services, which run on top of infrastructures. Adaptation is not straightforward when we take into account the heterogeneous and dynamic execution context of such complex systems. While several state-of-the-art approaches, unaware from each other, target different problems at specific parts of the system, the isolated enactment of those adaptations results in ignoring the overall impact on the whole SBS. In this ongoing PHD work, we propose a complete, flexible and extensible solution for the cross-layer adaptation of SBSs. Our proposed solution integrates and coordinates existing analysis and adaptation tools to assess the impact of an adaptation at different system levels. Moreover, throughout the impact analysis, starting from an initial adaptation trigger it incrementally constructs adaptation strategies, consistent with the overall system.

1 Introduction

Service-based systems must provide their functionality with the required/agreed qualities of services, cope with the unreliability of the infrastructure on which they operate, and also deal with the changes in the context in which they are executed, or in the partner services with which they interact. This means that all these problems must be discovered as soon as they appear, and these systems must be able to adapt their behavior to handle them.

Adapting the behavior may mean changing the actual composition of services, or selecting different partner services, but the satisfaction of the service level agreements in place may also impose changes in the way services are offered. For example the supervision system may re-negotiate some quality parameters with the providers of the partner services, change the configuration of the engine that runs the composition (e.g. BPEL process), and even adjust the infrastructure (resources) used by the application and its partners.

Many existing solutions [1,2,3,4,5] have addressed adaptation in a "local" way by only considering one concern at one layer. However, if we considered the whole SBS stack, some adaptations may trigger others, or they may influence some quality parameters, or even the operation, or parts of the system. Adaptation is thus a cross- and

* The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

G. Pallis et al. (Eds.): ICSOC 2011, LNCS 7221, pp. 213–219, 2012.

multi-layer problem and we must take into account the entire system consistency rather than only local problems and their specific solutions. To address this issue, this PHD work introduces a novel approach that comprises a comprehensive cross-layer system representation, and an extensible platform that integrates and coordinates existing analysis and adaptation tools in a holistic way. Our approach assesses the impact of an adaptation at different system levels, and in case of detecting an adaptation's incompatibility, it proposes new adaptations and incrementally constructs adaptation strategies (a set of adaptations), consistent with the whole system.

2 Problem Statement

We present a scenario to illustrate the cross-layer adaptation problem. We define the SBS layers for the scenario as follows: *(i) Application layer:* "Call & Pay Taxi" composite service (CPTS), implemented as a BPEL process. *(ii) Service layer:* The partner services of our application, namely a short messaging service (SMS), a location service (LS) and a payment service (PS) provided by the telecom company, and the taxi service (TS) provided by the taxi company. *(iii) Infrastructure layer:* The underlying platforms on top of which CPTS, SMS, LS, PS and TS run.

In CPTS, the client requests a taxi by sending a text message (SMS) to the application. Then, his/her location is identified and the taxi company is contacted to organize the real taxi service. After transporting the user to the destination, the process terminates with a successful payment.

Let us consider an adaptation case on our scenario: The CPTS provider decides to switch to a cheaper telecom provider. This means replacing SMS, LS and PS in the BPEL process. However, there is a problem with the new LS service's output message format. It provides client's location in geographical coordinates instead of full address while in our application design we use full address as the input message to the TS. To remove this new data mismatch, the service composition is adapted by adding a data mediator service in the workflow. Yet it triggers a new problem: We notice that the new service for data mediation is too costly and in fact increases the overall cost of the process in an unforeseen way, eventually conflicts with the initial adaptation goal, which is cost reduction.

Problem. With the existing approaches when an adaptation is performed, it targets a particular problem occurring at a specific aspect of a specific SBS layer. Thus, they tend to propose local solutions to local problems in a way that is isolated from the overall application context. To avoid this, we must understand the impact of a change across different system layers, which have their own characteristics and constraints. In our example while we are trying to improve the cost, yet we do not know the consequences of replacing the services for the whole system.

We need a holistic approach with the following research challenges in hand [6]: (i) Address adaptation problem globally for the SBS. (ii) Provide a neat and comprehensive reference model for system aspects and layers and the adaptations in the SBS. (iii) Integrate existing and forthcoming solutions that are in isolation into a consistent and coherent solution. (iv) Understand the impact and consequences of possible adaptations for all the system elements. (v) Release an innovative and extensible cross-layer

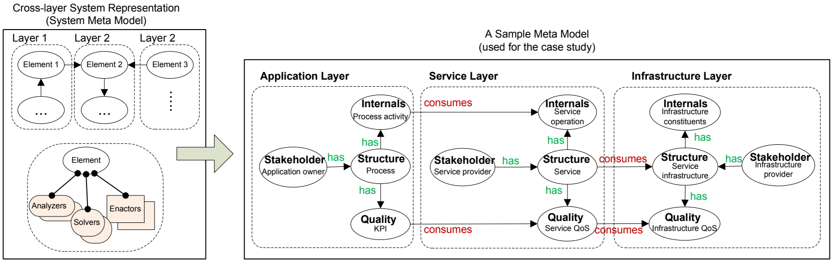

Fig. 1. System Modeling for CLAM

adaptation solution to devise efficient adaptation plans consistent with the overall SBS. (vi) Evaluate the proposed solution on various application domains.

3 Proposed Solution

Our approach exploits two key facts about SBSs to solve the cross-layer adaptation problem: First, SBSs already contain the dependencies among different layer elements implicitly while those dependencies are not trivial among adaptations in different layers. Hence, we would like to benefit from the layer dependencies known for the system, and model the SBS in a cross-layer manner so that the dependencies get explicit to be easily used by our technique. Second, there are several state-of-the-art approaches for various types of SBS analysis and adaptation, and they already run in the system in an uncoordinated way. In our approach we would like to plug those mechanisms in a cros-layer adaptation management (CLAM) platform so that we can reuse them to analyze the impact of an adaptation trigger and to extend it if required. Thus, CLAM aims to coordinate the current approaches to prevent conflicting adaptations and to produce a final, validated adaptation strategy aligned with the overall SBS. Taking into account these facts, our solution has three bases:

1) Cross-layer System Representation. We solve cross-layer adaptation problem on the meta model of the system. It is basically a directed graph where nodes represent the system elements from different layers and the edges represent the relations between these elements. Apart from system elements and their relationships, we have the system artifacts associated to the elements. They are the analysis and adaptation mechanisms that are available for the SBS: *(i) Analyzers* check the compatibility of a new adaptation for a system element that they are associated to. *(ii) Solvers* get an incompatibility problem triggered by an analyzer and try to propose an adaptation to handle the problem. *(iii) Enactors* implement final adaptation strategies validated by CLAM.

The meta model for our case study could be found in Figure 1. While this model is created for the application scenario in this work, meta models for other application domains could involve different system elements and relationships. Our approach is able to work with various application domains, as long as the application system can be be modeled by its elements and their dependencies to each other.

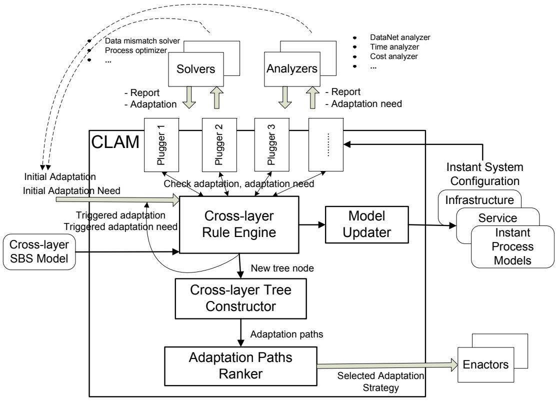

Fig. 2. CLAM: Cross-layer Adaptation Manager

2) Methodology. We perform the entire adaptation analysis as a continuous execution of predefined *rules*. These rules determine how to navigate the SBS meta model to identify the system elements affected by an adaptation, and subsequently how to decide which analyzers and solvers to invoke, and finally how to gradually construct a cross-layer adaptation tree upon receiving results from those external tools. Cross-layer adaptation tree is basically the output of the impact analysis where the branches correspond to alternative adaptation strategies that can overall address the negative impacts of an initial adaptation. Figure 3 shows a sample tree produced by our approach for the analysis of the adaptation case study presented in Section 2.

3) Supporting Tool. The architecture for CLAM platform is given in Figure 2. In order to perform a comprehensive impact analysis for an adaptation, CLAM executes the rules defined in the methodology. Moreover, through its pluggers it provides an integration platform where one can plug in new analyzers and solvers, and improve the analysis by including more artifacts in the system.

We have implemented a first version of CLAM where we have a time and cost analyzer associated to the KPI node in the meta model, then DataNet analyzer associated to the process activity node, finally process optimizer and data mismatch solver associated to the process node. All of them are real tools from the state-of-the-art.

In this first implementation, let us see how CLAM platform works to produce the tree in Figure 3 for our case study: CLAM makes use of instantiations (instant model, e.g. M_0, M_1...) of the meta model with concrete value assignments of system elements to keep track of alternative adaptation proposals during an analysis. When the new adaptation trigger "replace telecom provider X by Y" comes to the rule engine, the engine identifies which nodes in the instant model are changed (service provider, service, service operation, service QoS) and which nodes might be affected due to those changes (process activity, KPI). For the changed and affected nodes it identifies the associated artifacts (datanet, cost and time analyzers) to be invoked for the impact analysis. Then, it contacts the model updater to update the instant model with the changes. On the new instant model it initiates the analysis by calling the analyzers one by one. If they report

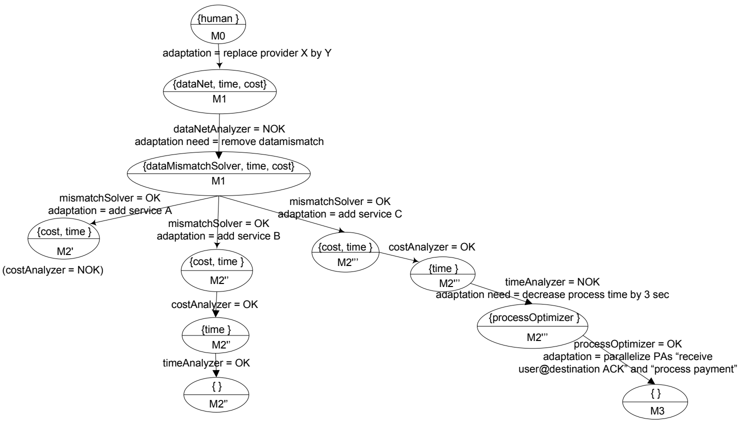

Fig. 3. CLAM Tree for Alternative Adaptation Paths

back some problems (adaptation needs), then it determines and contacts a proper solver to solve the problem newly occured. In this way, it continues the analysis until all the necessary artifacts are called.

The outcome is the cross-layer adaptation tree with some alternative adaptation paths (strategies). We take a tree branch as an alternative strategy only if all the problems on the path have been solved properly by the artifacts. Finally based on some criteria (e.g. shortest path, minimum cost path etc) we can select one of the paths, and through enactors we can deploy the adaptations present on the path.

Preliminary Results.The construction of the tree in Figure 3 took around 3 seconds on the 2 GHz Intel Pentium M Processor Windows XP laptop with 1 GB of RAM. Our experiences from the first implementation show that it is worth investigating further the on-line usage of our solution considering the time performance. Moreover, our approach is light-weight and extensible to plug in various artifacts in CLAM.

4 Related Work

Our expected research impact with respect to the state-of-the-art is in two ways: First, existing adaptation work mostly focuses on specific aspects of the SBA where adaptation problem is solved merely in this narrow scope without taking into account its consequences for the whole SBS stack. The approaches in BPM adaptation [1], self adaptation and self healing systems [2], QoS-awareness [3], mediator design for service interactions [4] and finally context-awareness [5] are the prevalent ones in this category. Adaptations proposed by such approaches may be conflicting with each other, even they may be harmful for the application. Our solution aims to align and coordinate them effectively to prevent such cases.

Second, there are few approaches in literature that use cross-relations for adaptation. However, compared to those works our approach brings the novelty of extensibility and genericness, i.e., not being domain specific. In this category, we have [7] that analyzes the dependencies of KPI violations on quality metrics from different layers of an SBS. Then, [8] proposes a technique where the designer prepares the taxonomies of adaptation mismatches, and later designs the adaptation templates that define generic solutions to tackle mismatches. Finally, [9,10,11] have a cross-layer representation of the system. While [10,11] target limited number of adaptation cases such as service replacement, [9] uses the cross-layer model for monitoring and analysis rather than adaptation.

5 Conclusions and Future Work

We have presented a holistic approach to solve the cross-layer adaptation problem. In the remaining six months of this PHD work, we will work on the formalization of both the research problem and the proposed solution. This will directly contribute to the enhancement of the implementation. We will use more adaptation case studies and more application domains where we have different system meta models. In this way, we are planning to consolidate the platform regarding its extensibility and genericness. Meanwhile we will also improve the algorithm especially considering the tree construction to avoid infinite trigger of adaptations, i.e., to ensure the termination of the algorithm. Eventually, we would like to evaluate our approach in two ways: (i) investigating the usability of our solution at run-time, (ii) comparing two cases where in one case we run the adaptation and analysis tools coordinated by our approach, and in the other case we run them in an uncoordinated manner.

References

1. Brogi, A., Popescu, R.: Automated Generation of BPEL Adapters. In: Dan, A., Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 27–39. Springer, Heidelberg (2006)
2. Charfi, A., Dinkelaker, T., Mezini, M.: A plug-in architecture for self-adaptive web service compositions. In: ICWS 2009: Proceedings of the 2009 IEEE International Conference on Web Services, pp. 35–42. IEEE Computer Society, Washington, DC (2009)
3. Christos, K., Vassilakis, C., Rouvas, E., Georgiadis, P.: Qos-driven adaptation of bpel scenario execution. In: ICWS 2009: Proceedings of the 2009 IEEE International Conference on Web Services, pp. 271–278. IEEE Computer Society, Washington, DC (2009)
4. Kongdenfha, W., Motahari-Nezhad, H., Benatallah, B., Casati, F., Saint-Paul, R.: Mismatch patterns and adaptation aspects: A foundation for rapid development of web service adapters. IEEE Transactions on Services Computing, 94–107 (2009)
5. Ardagna, D., Comuzzi, M., Mussi, E., Pernici, B., Plebani, P.: Paws: A framework for executing adaptive web-service processes. IEEE Softw. 24, 39–46 (2007)
6. Zengin, A., Kazhamiakin, R., Pistore, M.: Clam: Cross-layer management of adaptation decisions for service-based applications. In: 2011 IEEE International Conference on Web Services, pp. 698–699. IEEE (2011)
7. Kazhamiakin, R., Wetzstein, B., Karastoyanova, D., Pistore, M., Leymann, F.: Adaptation of Service-Based Applications Based on Process Quality Factor Analysis. In: Dan, A., Gittler, F., Toumani, F. (eds.) ICSOC/ServiceWave 2009. LNCS, vol. 6275, pp. 395–404. Springer, Heidelberg (2010)

8. Popescu, R., Staikopoulos, A., Liu, P., Brogi, A., Clarke, S.: Taxonomy-driven adaptation of multi-layer applications using templates. In: SASO 2010, pp. 213–222 (2010)
9. Baresi, L., Caporuscio, M., Ghezzi, C., Guinea, S.: Model-Driven Management of Services. In: ECOWS 2010, pp. 147–154 (2010)
10. Tripathi, U., Hinkelmann, K., Feldkamp, D.: Life cycle for change management in business processes using semantic technologies. Journal of Computers 3, 24 (2008)
11. Burgstaller, B., Dhungana, D., Franch, X., Grunbacher, P., López, L., Marco, J., Oriol, M.: Stockhammer: Monitoring and Adaptation of Service-oriented Systems with Goal and Variability Models. Technical report, Universitat Politècnica de Catalunya (2008)