A flexible high-performance service-oriented production system for Triple Play telecommunication products

Timo Steffens[1], Robert Mertens[1], Stefan Andres[2]

[1]Fraunhofer IAIS, Schloss Birlinghoven, 53754 Sankt Augustin

[2]T-Home, Am Wissenschaftspark 31+33, D-54296 Trier

{timo.steffens,robert.mertens}@iais.fraunhofer.de, stefan.andres@telekom.de

Abstract: This paper introduces concepts for a service-oriented IT-landscape supporting flexible and high-performance provisioning of telecommunication products. On the one hand these concepts are designed to be generic in order to facilitate a short time-to-market for future products. On the other hand the concepts guarantee a scalable and high-performance operation of the system. The concepts have been developed in cooperation with and successfully employed in a major German telecommunications company.

1 Introduction

Nowadays, telecommunication products have an increasingly short life-cycle. This is due to the fact that new products are launched whenever there are technological advances. This situation places high demands on the flexibility of IT-systems supporting the provisioning and operation of the products. On the one hand, the IT-landscape has to be generic and flexible in order to be open for future product launches and to guarantee a short time-to-market. On the other hand, typical IT-systems for provisioning have to guarantee high performance in order to handle several thousands of processes per hour. This is especially important for technically complex products like Triple Play services (Voice-over-IP, TV over IP, and internet access).

Usually, there is a trade-off between genericity and performance. This paper describes how this problem has been been tackled for the provisioning and operation of Triple Play Services. Particularly, we introduce the ”FlexProd”(Flexible Production) system, which controls the production of T-Home Triple Play products, including prequalification, provisioning, storno etc.. FlexProd employs a special service-oriented architecture that has been developed particularly for the requirements of Triple Play products. The architecture design was driven by the fact that the products have to be independent from the underlying technic (e.g. special hardware versions for DSLAMs or Label Edge Routers), from protocols (such as VDSL or ADSL). It should also be modular in order to facilitate the flexible definition of product bundles for marketing purposes.

The remainder of this paper is organized as follows. The next section briefly describes the process of provisioning Triple Play products on an abstract level. Section 3 introduces the general FlexProd architecture that is custom tailored to meet the needs of this process. In section 4 it is described how performance is optimized by using short-term processes only. Consequently, longer-lived processes are handled using production plans which are outlined in section 5. Finally, the last section concludes the paper and outlines future work.

2 Provisioning Processes for Triple Play Products

Processes for provisioning a Triple Play telecommunications product usually involve many different IT systems and are distributed over a number of business units. In general such a process is organized as follows: (1) In a so-called prequalification test, the customer's line (identified by the customer's adress) is checked against a representation of an inventory containing several millions of ports, pins, cables and Second Mile equipment to determine whether the network supports the product on this line. (2) If this check was successfull, a provisioning process is started. This process includes reconfiguration of port profiles in various Second Mile elements and access management in the AAA (Authentication, Authorization and Accounting) domain as well as manual network reconfiguration in the First Mile and installing access devices at the customer's home. To these ends, a multitude of IT systems have to be coordinated that in turn configure network elements, interface to the AAA domain or handle the disposition of work-force.

3 FlexProd Architecture

FlexProd serves as the central control unit for the provisioning and handling of Triple Play services. It realizes some of the tasks involved itself, such as managing the inventory or running prequalification test. For other tasks it serves as a form of control unit that interacts with the IT systems actually performing the tasks (see section 2). This section outlines the general aspects of the FlexProd architecture to illustrate its most important concepts.

Business transactions and services:

On the technical level, the FlexProd architecture is a service-oriented architecture [Glo07, HWSD07], conforming to the J2EE standards for message-driven services. As will be detailled later, all services are required to be stateless for performance reasons.

On an abstract level, FlexProd is designed according to business transactions. Any operation initiated in FlexProd is part of 14 customer-to-customer business transactions. That is, FlexProd does not offer single services to other systems, but only business transactions. Each of these 14 cases is implemented in a special process modelling language called LAMPS [HSZ06]. This allows to adapt to business needs on the business logic level without interfering at the source-code level (cf. [Sch98]).

Data model:

An important part of FlexProd is a database modelling the first-mile-network of more than 750 cities. The topology and material properties of the copper cables are essential for checking whether a certain DSL-bandwidth can be guaranteed for a customer. Additionally, the devices of the second mile (fibre-part of the network) are modelled in the database. Since the hardware, technics and protocols are prone to change with technological advances, the data model has to be flexible in order to cope with any changes. To this end, a meta-data-model has been developed, that provides much of the flexibility of object-oriented data-models while retaining the high performance of relational data-bases.

All business objects are categorized into so-called aspects. These aspects are *technics* (e. g. ports, pins, cables, switches), *production plans* (see next section and section 2), *product information*, and *contracts*. In each aspect, types can be defined with their components, attributes and relations without modifying the physical table-definitions or adding new tables. The representations of concrete instances are then stored in existing table-configurations that are specified by the type-records.

4 Processes Model

FlexProd has to deal with two different kinds of processes: short-term and long-term processes. We define long-term processes as any process with a duration of more than 30 seconds. In telecommunication, the largest part of long-lived processes is made up of waiting (e. g. for the begin of the ordered operation period or for receipts from the work-force). Since long-lived processes may have a duration up to several weeks, keeping the calling threads alive is practically impossible as they bind ressources and are sensitive to system reboots, system failures etc.. Thus, a crucial design decision was to allow only short-term processes in FlexProd. Therefore, even processes that take several weeks in total are emulated by short-term processes that each take up to 30 seconds maximum. The states of the emulated long-lived processes are stored in a database. In order to establish the overall structure of the complete processes, each business transaction is made up of two phases:

In the near-real-time phase, the short-term processes are executed. One of these short-term processes configures a so-called production plan for the customer. This production plan comprises a description of all the steps that have to be taken to fulfill a customer's request (such as ordering a product or cancelling a subscription) and the order in which these steps have to be taken. The production plan has to be customer-specific since there exist several thousand different configurations for providing a household with DSL. In the second phase, the production plan is executed.

If a customer orders a Triple Play product for usage starting at a *customer specified date*, the phases might look like the following (simplified for ease of illustration).

Near-real-time phase:

1. Prequalification: Using its inventory, FlexProd checks whether a customer (as identified by an address or phone number) can be technically served with a certain DSL-access. An optimized route through the cable network is identified. Furthermore,

FlexProd communicates with the work-force-management system in order to check whether technician personnel is available for the *customer specified date*.

2. Reservation: The ressources in the network (ports, pins, wires, VLANs) are reserved for the customer and the necessary activities are persisted as a production-plan. If, however, within several minutes no booking takes place, the reservations and the production-plan are discarded.
3. Booking: After the customer has signed the contract, the production-plan is started. Note that at this time the provisioning process is already planned down to every detail and will not be modiefied unless problems occur during its execution.

Production-plan-phase:

1. Work-Force: Few days before the *customer specified date* the work-force system retrieves technical data from FlexProd which is required for the technician (e. g. port numbers, location of the different hardware multiplexers). The disposition of the work-force is then managed by a special work-force system.
2. Network-Activation-System (NAS): Simultaneously, FlexProd transmits the product profile data to the NAS so that the ports are activated.
3. AAA-Domain: After the NMS signals that the configuration has been finished, FlexProd provides authentification data to the RADIUS server [RRSW00] in the AAA domain.

It is obvious that splitting complex processes into shorter ones requires means of linking processes together. Particularly, when a synchronous process is divided into asynchronous tasks, the reply of the second process has to be associated to the original cause. This is done by using correlation-IDs that are generated during the near-real-time phase and are persisted into the activity steps of the production plan.

Using production plans has several advantages. First, since they are persisted in the database, they are robust against system-failures and -reboots after maintenance. Second, they do not use up ressources when they are inactive (waiting for their activation). And third, they are a convenient way to document the process execution by holding status information whether a step is still waiting, has been finished or has failed. The front-end to the customer can be provided always with the exact status of the business transaction.

As mentioned earlier, the business transactions are modelled in a Petri-net-based language called LAMPS [HSZ06]. The models are executed by a workflow-engine that has originally been developed for agent-based simulation [SH08]. This allows to simulate the whole process model before deployment. Furthermore, the business transactions benefit from the inherent properties of Petri Nets, such as concurrency and efficiency [Voj97].

Experiences with high-performance systems reveal that database-transactions bind ressources and produce long intervals of waiting [BCK98, GMS87]. Therefore, FlexProd does not use transactions that have scopes beyond one service. Mitigation strategies for failed actions are implemented and modelled manually using a newly developed tool-kit. The higher implementation costs have been accepted in favour of performance.

5 Production plans

This section provides more detailled insights into production plans that were already discussed in the previous section.

Production plans as used in FlexProd have their roots in the parts lists employed in production processes (such as manufacturing cars or furniture). Each component on the list has to be processed and checked afterwards. Although an old idea in manufacturing, this idea is new for telecommunication products. Instead of components, the list consists of provisioning steps. Furthermore, the list is not restricted to being processed sequentially. Instead, each step can be linked to several predecessors and successors. A step is only executed if a) all its predecessors have been executed b) all associated FlexProd-watch-agents signal their agreement. FlexProd-watch-agents are modifiers that control the execution of steps in the production plan. There are the following types:

- Time-agent: A step may not be processed before a certain time. This is useful if the customer ordered his product for a time several weeks in the future. Since the actual provisioning only takes few days, the time-handler can be used to postpone provisioning.
- Alarm-agent: Contrary to the time-agent, this agent checks whether a specific time has passed without completion of the production plan step. Usually the alarm-agent will become active to signal disturbances.
- Ready-agent: A step may not be processed before a receipt from an external system has been received. This is for example used to guarantee that the NMS has configured the ports correctly, before access is granted in the AAA servers.

Each step in the production plan calls a business transaction. In this sense, a production plan is a persistant ordered graph of process-calls. The basic concept behind production plans is to reify processes. This way, active provisioning processes can be adapted to modifications in the provisioning cycle. If for example new hardware or new protocols are used, the persisted process data in the database can be updated during normal operations. Doing the same with running processes that are not reified would be much more challenging and would probably require an offline maintenance phase.

6 Conclusion and future work

The FlexProd system incorporates a number of features that are custom tailored to the needs of a telecommunications provider offering Triple Play products. The architecture offers flexibility for future technological advances while guaranteeing high performance. Thanks to the introduction of production plans, the system relies on short-term processes.

The FlexProd system has been successfully launched in the second half of 2006. Since that time, a number of changes on several aspects of the company ranging from technics

over protocols to new products had to be met. The object-oriented data modelling technique used has proven flexible enough to allow for a fast implementation of the neccessary adjustments. The concept to use manually implemented transaction handling has proven to be an important concept in order to guarantee high performance.

In the future, FlexProd will gain more interfaces to technical and service personnel. One approach to facilitate working with complex SOA systems is to specify the communication between services on a semantic level [RFPK07]. Future work on the FlexProd architecture will include the adoption of this approach.

Literatur

[BCK98] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. Addison-Wesley Longman Publishing, Boston, 1998.

[Glo07] Herbert Gloeckle. IT-Integration und Migration - Konzepte und Vorgehensweisen. *Praxis der Wirtschaftsinformatik*, 257(1):7–19, 2007.

[GMS87] H. Garcia-Molina and K. Salem. Sagas. *SIGMOD Rec.*, 16(3):249–259, 1987.

[HSZ06] Philipp Hügelmeyer, Timo Steffens, and Thomas Zöller. Agent-based Modelling of Processes and Scenarios with high-level Petri Nets. In Alexandre Nketsa, editor, *Proceedings of ESM 2006*, pages 36–40, 2006.

[HWSD07] Oliver Hoess, Anette Weissbecker, Thomas Specht, and Jens Drawehn. Migration zu service-orientierten Architekturen - top-down oder bottom-up? *Praxis der Wirtschaftsinformatik*, 257(1):39–46, 2007.

[RFPK07] Michael Rebstock, Janina Fengel, Heiko Paulheim, and Matthias Knoll. Semantische Synchronisationsdienste zur dynamischen Informationsintegration im elektronischen Geschaeftsverkehr. *Praxis der Wirtschaftsinformatik*, 257(1):28–38, 2007.

[RRSW00] C. Rigney, A. Rubens, W. Simpson, and S. Willens. Remote Authentication Dial In User Service (RFC 2865), June 2000.

[Sch98] M. Schwermer. *Modellierungsvorgehen zur Planung von Geschäftsprozessen*. PhD thesis, Fraunhofer IPK, Berlin, 1998.

[SH08] Timo Steffens and Philipp Huegelmeyer. Real-time prediction in a stochastic domain via similarity-based data-mining. In B. Biller, M.-H. Hsieh, and R. R. Barton, editors, *Proceedings of the Winter Simulation Conference 2007*, pages 1430–1435, 2008.

[Voj97] Tomas Vojnar. Various Kinds of Petri Nets in Simulation and Modelling. In J. Stefan, editor, *Proceedings of 31st Spring International Conference on Modelling and Simulation of Systems MOSIS '97*, volume 1, pages 227–232, 1997.