Process Support in eHome Systems: Empowering Providers to Handle a Future Mass Market

Ibrahim Armac and Michael Kirchhof

Department of Computer Science III, Aachen University of Technology,
Ahornstr. 55, 52074 Aachen, Germany,
{armac|kirchhof}@i3.informatik.rwth-aachen.de

Abstract. This paper focuses on the sales and distribution phase in eHome systems. The huge market potential for affordable solutions leads to a multiplicity in terms of millions of installations in parallel. This situation can not be handled manually. So, tool support for the business process in this phase is crucial. We discuss the prerequisites of the intended tool support, starting with the overall eHome process which describes the complete life cycle of eHome services. Then, we will further analyze the sales and distribution phase and formalize the encapsulated business process. This process specification is executed by an established workflow management system. Summarizing, we present a solution based on automated configuration and deployment techniques, which enables the efficient advertising, migration, and retirement of components in the particularly specific domain of eHome systems.

1 Introduction

In this paper we will take a look at the inside of connected homes, which build up complex IT systems. The building blocks of such systems are electronic devices, networks, and *services*, which empower the user to interact with his environment. Objects in the environment can be the room itself, the building, other persons, and information from outside. Talking about services, we mean any piece of software, which is executed in a *networked* environment, making usage and administration of pervasive appliances easier.

The structure of eHome systems is illustrated in Figure 1: The connected home on the right-hand side of the drawing is equipped with a *residential gateway*, a hardware device, which provides access to connecting infrastructure (e.g., X.10, EHS, Lon, Jini) and acts as the runtime environment for the *service gateway*. The service gateway manages and runs certain software components. These services realize *eHome Services*, which are these visible to the users. In our work, we focus on the domains of Security, Consumption, and Infotainment. The services in these domains are based on certain equipment, as some examples are shown in the figure: an alarm system depends on cameras, motion detectors, and lamps or sirens. Monitoring and optimization of energy consumption can be realized by the use of ammeters, photo sensors, thermometers, the heating systems and so on. Elements of the infotainment domain can be incorporated for audio-based and video-based interaction. The internal and external communication can be handled equivalently using an an IP-based platform as communication backbone.

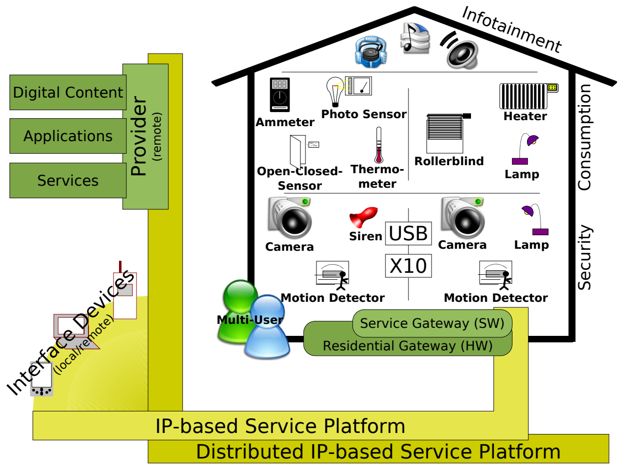


Fig. 1. Structure of eHome Systems

Beside direct interaction with devices in the house, interaction with the eHome system based on personal computers, PDAs, and mobile phones is realized. Service providers are connected to the systems via the IP-based platform to provide digital content, applications, and services. This stands in contrast to many other solutions: Though they provide network access, the supported functionality is most often restricted to remote controlling single devices.

One application of these technologies is a modular alarm system which consists of multiple cameras, multiple sensors, an outbound connection for alarm messages (e.g., email facility, SMS), and some power switches. All components are integrated and coordinated by the residential gateway. The procedure is the following: when some of the sensors, for example motion detectors, detect something worth mentioning, a predefined subset of available switches are activated and selected cameras should take pictures of the location and store them. The house-owner is informed about what is currently happening in his house in order to take adequate measures based on the kind of the event and the pictures obtained from within his house. Possible actions would be to call the police in emergency cases or to discard the event, in situations where the cat raised the alarm accidentally. The provider conserves evidence, which is crucial in the case of the local recording device is stolen, too. The system is extensible by additional functionality. For example, the alarm system setup can be tweaked by the user in terms of which rooms should be monitored. Or, the way to alarm the owner should be made flexible.

Current situation is, that there is no solution for the distribution und deployment of eHome services in such a way, that one provider is able to handle thousands of installations in parallel. The proposed solution in current systems, if even discussed, is that an expert representative of the provider gathers the required data and installs the system and services manually on the customer's site. This task is time-consuming and expensive. Thus, it prevents affordable eHome solutions.

In this paper, we will discuss how the infrastructure and the developed tool support for the development process can be combined to reduce the efforts to deploy eHome services in a mass market. We will describe the roles and their tasks with respect to a clear competence-driven separation of concerns and discuss the process specification which can be executed by a workflow management system. This automation reduces the efforts on the provider's side and leverages the contentment of the customer.

At Department of Computer Science III at Aachen University of Technology, tools for supporting development processes have been built for software engineering, mechanical engineering, chemical engineering, process control, telecommunication systems, and authoring support. This fundament is used for the research of process support in eHome systems. The support is based on ongoing research of processes and infrastructures in the domain of eHome systems at our department (eHome-Group).

The paper is structured as follows: First, we will describe the organizational perspective on eHome systems, namely the eHome process in Section 2. The last phase of this process – the sales and distribution phase – is then discussed in detail in Section 3. Here, the identified activities of the business process will be formally specified using a workflow language. This specification is based on tools supporting the immanent development process in the distribution phase *and* can be automated and monitored. After a discussion of related work in Section 4, we conclude of paper with a summary and an outlook to future work in Section 5.

2 eHome Process

For successful eHome solutions, the multiplicity is a key factor. Today, eHome solutions are products made on specific request and, thus, according to specific requirements. But, the in-materiality of software imposes no reproduction costs if used several times. The cost driver is the customizing and deployment in specific eHomes. If this can be changed, affordable low-cost eHome solutions can be achieved.

Our approach settles on the idea of a clear separation of product-specific tasks, tools, and data on the one hand and eHome-specific tasks, tools, and data on the other hand. Figure 2 depicts the *eHome process* containing these both parts. At the beginning, a new product (an eHome service) is being developed. This should be done under the premise of (re-)usability in different contexts. While this is impossible in general, it is possible in the domain of eHome systems and is supported by our layered software architecture PowerArchitecture [1]. After the development of the product, the business process starts consisting of sales and distribution activities. The three building blocks are (1) *Configuration & Deployment*, (2) *Execution & Billing*, and (3) *Uninstallation*. The backward-oriented loop represents maintenance tasks during runtime of the long-running eHome installation. These blocks will be detailed in the following.

In the first step (1: Configuration & Deployment), the acquisition of data and the matching between environment and software components is being executed. The results of this computation is a formalized description of the components to be installed on their initial settings. This specification is then automatically deployed in the cus-

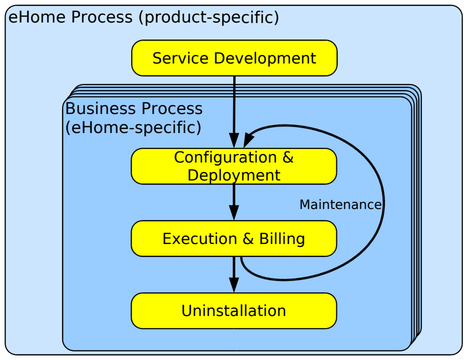

eHome Process (product-specific)
Service Development
Business Process (eHome-specific)
Configuration & Deployment
Maintenance
Execution & Billing
Uninstallation

Fig. 2. The Coarse-Grained eHome Process

tomer's eHome [2,3]. In the second step (2: Execution & Billing), the customer uses the installed eHome services. We encourage the idea of a service subscription with recurrent subscription fees, instead of a single selling activity. The subscription reflects the relationship between provider and customer in an appropriate way and enables ongoing support during whole life time of the eHome. The third step (3: Uninstallation) is triggered by the cancellation of the subscription. This part of the process has be to dealt with, too. This is caused by the long running eHome systems, which have to be in consistent state, whether new services are installed or existing ones are removed. Due to interdependencies between eHome services, this is a difficult task. To tackle this problem, we again utilize the configuration and deployment mechanism which has been used in the first step.

The business process is being executed in close collaboration between provider and customer. Here, data about the current situation of the customer's home, his wishes, and requirements come into play. Thus, this part of the eHome process is eHome-specific. Due to the multiplicity of eHomes installations, the cost driver is the eHome-specific part of the eHome process. So, the sales and distribution phase has to be considered carefully to reduce manual efforts and on-site expert service. The tasks and documents in this phase are depicted in Figure 3. This first task in this phase is executed by a sales person, who adds the annotated products to the product range. The customer can query the provider's product range to figure out products and functionalities he would like. The preselection of products is supported by gathered data about existing devices, other equipment, and structural data of the customer's home. For this gathering, the customer is supported by an interactive tool called *eHomeConfigurator* [2]. By the use of the configuration technique, the customer is provided with a customized offer which reflects the intended functionality *and* the current situation in the customer's home. This offer can contain additional equipment required by the selected products or can be restricted to the specified home environment. The process of gathering information and computation of customized offers can be repeated arbitrarily often with no negative impact because the computation task is completely automated. Once the offer fulfills the customer's wish, the product selection and the gathered data are refined again by the configuration

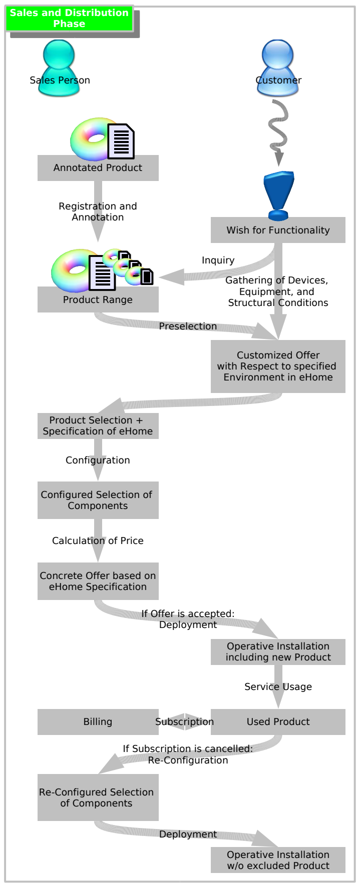

Sales and Distribution Phase
Sales Person
Customer
Annotated Product
Registration and Annotation
Wish for Functionality
Inquiry
Gathering of Devices, Equipment, and Structural Conditions
Product Range
Preselection
Customized Offer with Respect to specified Environment in eHome
Product Selection + Specification of eHome
Configuration
Configured Selection of Components
Calculation of Price
Concrete Offer based on eHome Specification
If Offer is accepted: Deployment
Operative Installation including new Product
Service Usage
Billing
Subscription
Used Product
If Subscription is cancelled: Re-Configuration
Re-Configured Selection of Components
Deployment
Operative Installation w/o excluded Product

Fig. 3. Sales and Distribution Phase of the eHome Process

technique resulting in a configured selection of components and their initial settings. Based on this, the accurate price can be calculated. If this *concrete offer* ist accepted, the specified set of components is automatically deployed in the customer's home. The operative installation including the new product is then used within a service subscription, allowing ongoing interaction between provider and customer for integration of remote digital contents, services, and maintenance. If the subscription is canceled, the configuration technique is again used to compute a set of components and their settings without the unsubscribed service which represents a still operative eHome installation. The resulting specification is again automatically deployed in the customer's home.

The described business process has some similarities to classical software development processes. The analogies can be seen (1) between specification of the eHome and requirements analysis in established development processes, (2) between configuration of components and the manual integration and linking of components, and (3) between automated deployment and the manual migration of component sets and their settings. By the introduction of the techniques for configuration and deployment, the once required *eHome-specific development process* has been eliminated. The tasks in the sales and distribution phase cover the complete life cycle of an operative eHome service. Especially, the retirement of services is taken into account due to the requirement of eHome systems being in consistent states at all times. This requirement reflects the principle of the least astonishment. In the literature, this phase has not yet being discussed in depth, maybe because of the difficulties of removing highly inter-related components of a long-running system. To tackle this problem, we utilize the configuration technique and the abstraction layers from PowerArchitecture [1].

3 Business Process

As aforementioned, the cost driver in the business process is the customizing and deployment of eHome services in specific eHomes. However, up to now there is no formalization of this process and, as a result, no suitable tool support for the execution and monitoring is available although the tools developed in the eHome Group for configuration and deployment allow for automated execution of subtasks. In this section, we will address different issues to enable the automated execution of the business process on the provider's side.

3.1 Structure of the Business Process

Following the previous section, the business process is embedded into the eHome process. The main roles included into the business process are the *provider* offering eHome services and the *customer* subscribing for these services. Figure 4 shows the structure of the process including the identified activities, the transitions between them, and the mentioned roles. Furthermore, the process is divided into 5 phases, indicated by the dark text fields. These phases are *specification and configuration*, *deployment*, *execution and billing*, *uninstallation*, and *updates and upgrades* and will be described in detail in the following. In general, the process is *not sequential* debited and represents the phases an eHome service undergoes in a suitable way. Furthermore, it is informative and it has an *repetitive* character, that is, it can be instantiated for each customer and service without modification. The possible dynamics due to different customer concerns are intercepted in the different activities and do not affect the general process structure.

The business process begins when the customer initiates the process execution (activity `StartConfigurator`) by using the *eHomeConfigurator* [4]. The *specification and configuration* phase is inspired by the *working area model* [5]. Actually, this phase combines the activities of the general tasks of *specification of customer demands*, *specification of the eHome*, and *configuration of the services* which are logical closed. The eHomeConfigurator supports the user by executing the corresponding activities in an arbitrary order without restrictions concerning an execution sequence. It provides a graphical user interface which the customer can use to specify the structure of his eHome and formalize his demands (activities `ServiceSelection` and `EnvironmentSpecification`). The structure of an eHome contains for example the rooms and the available devices whereas the customer demands indicate the services

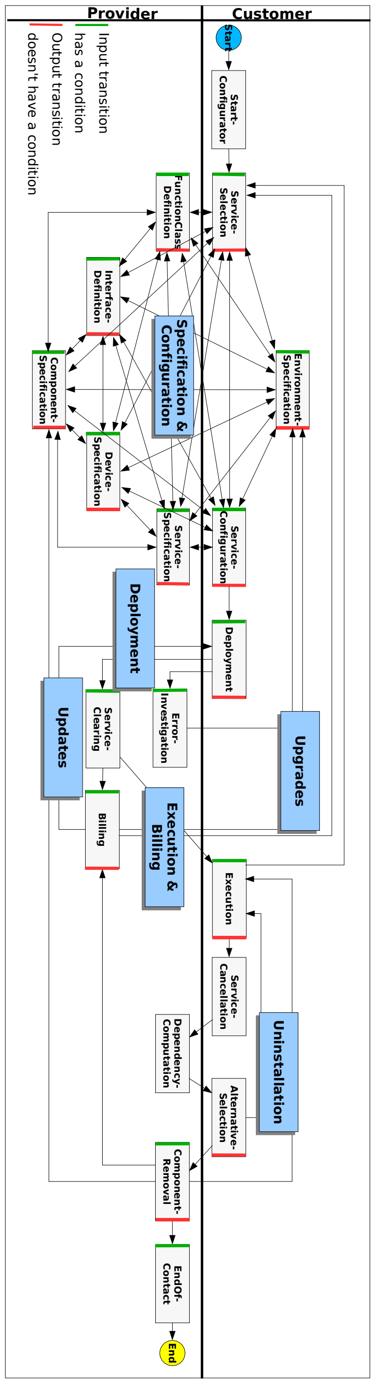


Fig. 4. The Formal Structure of the Business Process

the customer wants to subscribe and the scope of their functionality. Furthermore, the customer can configure the selected services (activity `ServiceConfiguration`), that is, which devices should be used and which constraints have to be taken into account etc., and will be notified about conflicts, missing data, and possible causes.

The provider uses the eHomeConfigurator to maintain generic information like the specification of provided services, supported devices, functionalities they offer and so on (activities `FunctionClassDefinition`, `InterfaceDefinition`, `ComponentSpecification`, `DeviceSpecification`, and `ServiceSpecification`). The provider can also execute the same activities as the customer to test new services in different environments, on the one hand, and to give remote support to the customer if needed, on the other hand. The customer, however, does not have access to the provider-specific activities. There exists an one-to-one mapping between the tabs in the eHomeConfigurator and the activities in the process. So, the way the "'next"' activity is chosen in this phase is determined by the task the user choses to execute next via the eHomeConfigurator.

The inspiration by the working area model addresses the issue of flexibility: Both customer and provider can proceed in any order. They can interrupt an activity, start another one and later resume the first one. The data will remain consistent because all activities are performing on the same data model. The structure of the specification and configuration phase builds up a complete graph, which reflects the flexibility of the the provided approach. For example, the customer can first begin to specify his eHome and then check which services he can subscribe for offhand. Nevertheless, he can also first select all services he wishes and then check which devices have to be bought for the installation of these services.

After the desired services have been configured successfully, the deployment of the services can be started by the *Deployer* (activity `Deployment`), a tool which automatically installs and instantiates the services belonging to the previously generated configuration. If errors occur during the deployment, the provider investigates these and triggers adequate steps for their correction (activity `ErrorInvestigation`) while the user is not confronted with this unpleasant task. Usually, the errors are due to inconsistent specification of the eHome environment. For example, the customer may have specified some devices incorrectly. This is why the transition from the activity `ErrorInvestigation` leads to `EnvironmentSpecification`. Note that this activity can also be executed by the provider as mentioned. The rare case of errors in the data maintained by the provider is not shown here for clarity purposes.

Successful deployed services need first to be cleared by the provider, for example after the first receipt of payment (activity `ServiceClearing`). Now the customer can use the subscribed services and the provider can bill the subscription fees (activities `Execution` and `Billing`). The activity `Execution` implicitly contains also the process of the customer interaction with the services, which is different for each service and need not to be specified here.

Like every software system, also eHome systems need to be he maintained. We distinguish two classes of maintenance activities, namely *updates* and *upgrades*. Both cause feedback flows in the process. However, they differ in their scope. An update is usually the replacement of components without side effects (e.g., bugfixing or optimizing). Thus, a redeployment of the corresponding components suffices. In contrast to updates, the scope of upgrades is wider because of cutback or extensions of service functionality. Hence, an upgrade causes reconfiguration and possibly changes on the eHome specification. While in Figure 4 updates are represented by the transition from the activity `Billing` to `Deployment`, upgrades are represented by the transition from `Billing` to `EnvironmentSpecification`.

Last but not least, the *uninstallion* deals with the removal of services whose subscription is canceled. Since eHome services are component-based, there may exist dependencies between different services. Therefrom, the removal of a service can result in inconsistent states of still running services. Hence, first and foremost the existing dependencies must be analyzed and the customer must be informed if such inconsistencies occur. In this case, the cancellation could be denied, alternative components could be integrated by reconfiguration, or the functionality of running services may be flattened. Otherwise, the corresponding service can easily be uninstalled.

In Figure 4, the dependencies of the canceled services (`ServiceCancellation`) are computed (`DependencyComputation`) using again the configuration infrastructure. Depending on the results of this computation, different alternatives are provided to the customer. The customer can then choose one of the alternatives (`AlternativeSelection`): The transition from `AlternativeSelection` to `Execution` means that the canceled service(s) are not removed contrary to the customer's primary intent. The transition from `ComponentRemoval` to `ComponentSpecification` stands for the replacement of removed components by others to avoid inconsistent states. The remaining outgoing transitions from `ComponentRemoval` to `Execution` and `Billing` denote that there might be still other services in use while some components have been removed. If the customer cancels all subscriptions, the business connection between the provider and the customer ends (`EndOfContact`).

In the literature (e.g., [6]), workflow patterns which often are used as building blocks for process modeling are described. Our process contains the basic patterns *Sequence*, *Parallel Split*, *Exclusive Choice*, *Simple Merge*, *Multi Choice*, and *Arbitrary Cycles*. The Sequence pattern is found between the `StartConfigurator` and the `ServiceSelection` activities. An example for the Parallel Split pattern is given by the parallel execution of the activities `Execution` and `Billing`. The Exclusive Choice pattern is used for the successors of the activity `Deployment`: Either the deployment is successful and the services can be cleared or the occurring errors must be investigated. The Simple Merge pattern is intensively used in our process. One example is that only one of the incoming transitions of the activity `Execution` can be active at one time. Therefore, they need not to be synchronized. The two outgoing transitions of the activity `AlternativeSelection` build an example for the Multi Choice pattern: Both of the transitions might be activated if only a part of the canceled services are going to be removed. However, at least one of the transitions has to be activated (remove canceled services or not). Last but not least, there are examples of the Arbitrary Cycles pattern in the process, which is represented by the different loops (e.g., between `ComponentRemoval` and `Execution`).

The business process in the eHome domain has a repetitive character so that it fits for requests of all eHome customers. Concerning this fact, it is reasonable to use established workflow management systems to execute and monitor different instances of the process. A precondition therefore is that the process must be modeled in a formal notation. The *Workflow Management Coalition* (WfMC) provides the *XML Process Definition Language* (XPDL) [7] as a standard for workflow process modeling. Workflow processes defined in XPDL can be imported, executed and monitored by different free available or commercial workflow management systems (e.g. *Shark*). Furthermore, there exist editors that allow for graphical modeling of workflow processes and export them to or import from XPDL (e.g. *JaWE*). Due to this, we decided to use JaWE for modelling our business process to provide the basis for its execution and monitoring by Shark. Figure 4 shows an overview of our process modeled by JaWE.

Summarizing, we specified a flexible business process for eHome systems covering the interactions of providers and customers in the sales and distribution as well as subscription phase of eHome services. The tool integration, which will be described in more detail in the following, is realized in a way such that the user is well supported in specifying his demands and his environment in a graphical way. The flexibility achieved by the combination of the specification and configuration phases was inspired by the ideas of the introduced working area model.

3.2 Tool Integration

The formalization of the business process now opens up the possibility to automatically execute the business process for eHome services. The automation is based on tool support for the development tasks in sales and distribution, namely *specification*, *configuration* and *deployment*. These tools and their application will be described in the following.

Figure 5 shows the involved tools and their application in the sales and distribution phase. Both the provider and the customer use the *eHomeConfigurator* [4] in the *specification and configuration* phase. The eHomeConfigurator is an interactive tool with a graphical user interface, that handles the tasks of gathering information for the specification of the service's environment, the automated computation of a valid configuration (i.e., a self-contained set of components and their initial settings to ensure operative condition after deployment), and the deployment (i.e., the transfer of the components, their settings, and finally their migration into the runtime environment). The computed results are stored as the *deployment document*. With these steps, the phases of the working area model are finished. This document is fed into the *Deployer* whose purpose is the fully automated instantiation of the components in a concrete home. The result is an operative eHome installation. In the succeeding phase of Execution and Billing, the customer benefits from the subscribed eHome services, thus, from the realized functionality. The last phase of Uninstallation is entered if the subscription is canceled for any reason. Within this phase, the tools for configuration and deployment, i.e. the eHomeConfigurator and the Deployer , are again used to compute a valid setup and deploy then the necessary modifications. All the actions to be taken are represented as one instance of the process specification. There is one such a process instance for each eHome service in each eHome. So, the set of process instances reflect not only the currently subscribed eHome services, but also the predecessor and successor steps in the sales

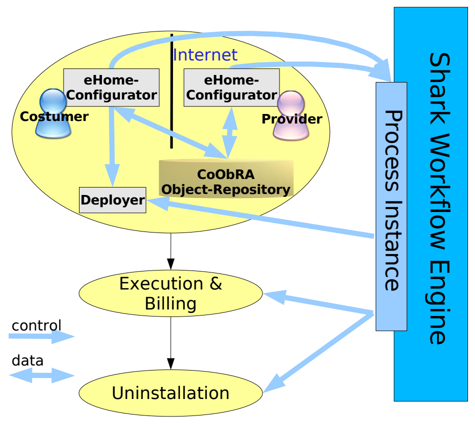


Fig. 5. Integrated Tools for the Business Process

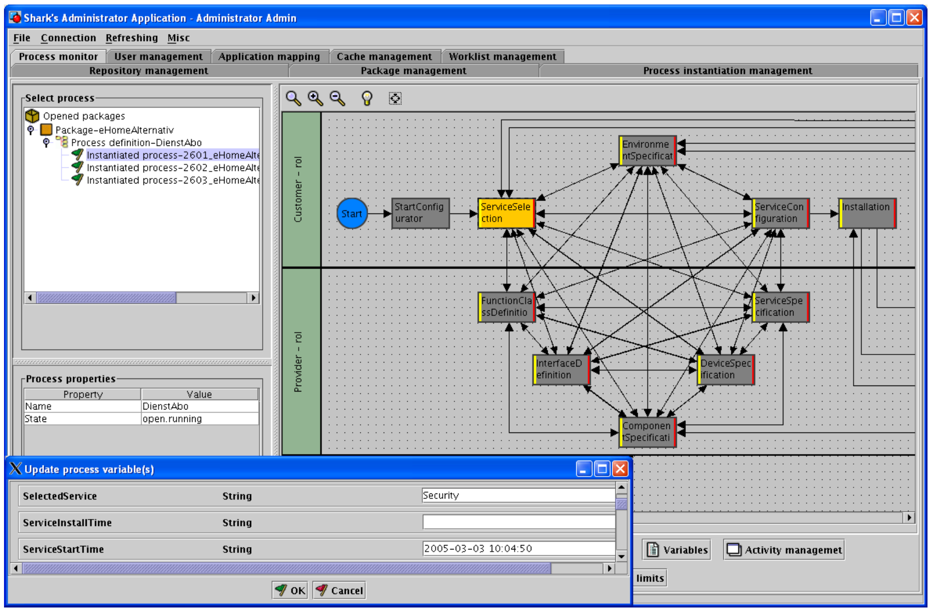
and distribution phase. For the instantiation and the control of the process, the workflow management system *Shark* is used. Shark is an open source workflow management system, which conforms to the workflow reference model of the WfMC [8] and fully supports XPDL. The ability to be triggered from the outside as well as to trigger actions in the outside adds up to Shark being a fully appropriate management system for the business process in eHome systems.

The eHomeConfigurator supports the different roles in the specification and configuration phase, but not the spatial separation. Enabling the provider and the customer working concurrently on the same data is necessary for the realisation of our idea behind the described process. For this purpose, we integrated a versioned object repository, namely *Concurrent Object Replication frAmework (CoObRA)* [9]. Using CoObRA, it is possible that the provider and the customer can work simultaneously on the same eHome-specific data. In case the customer has difficulties in carrying out specific tasks with the eHomeConfigurator, the provider can access the corresponding data an give immediately remote support also using an instance of the eHomeConfigurator. As a result, the need for cost-intensive on-site service can be reduced.

3.3 Execution and Monitoring of Process Instances

The formalization of the business process using XPDL has the advantage that, for each customer, process instances can be executed and monitored by established workflow management systems like Shark. As a result, the provider is able to manage the large number of customer relations by using such a system. In this section, we will describe how Shark can be used to keep track of the different process instances.

Figure 6 gives a snapshot of Shark's Administrator Application. In this snapshot, the process monitor is activated which is used to monitor the states of different process instances. For each eHome customer and eHome service, an instance of the business process is started when the customer initiates the configuration of a service by the



Process properties

Property	Value
Name	DienstAbo
State	open.running

SelectedService	String	Security
ServiceInstallTime	String	
ServiceStartTime	String	2005-03-03 10:04:50

Fig. 6. Execution and Monitoring of Business Process Instances

eHomeConfigurator. Therefore, we extended the eHomeConfigurator so that it can communicate with Shark in an appropriate way for creating and controlling different process instances. These instances persist until the services are uninstalled due to the cancellation of the concerning subscription. As a result, the provider can anytime monitor the status of the customer activities and the subscribed services by this customer.

In Figure 6, three process instances are shown as active on the top left side, from which the one with the number 2601 is selected for more detailed information. In the middle of the figure, the process structure is displayed where the currently active states are displayed light-colored. In this example, the customer currently selects desired services depicted by the activity `ServiceSelection`.

Down left in the figure, examples of process variables are displayed which can be used for different purposes like billing. The value of the `SelectedService` variable represents the eHome service for which the current process instance (2601) is started, namely our Security Service. Furthermore, the `ServiceInstallTime` shows that the Security Service is not installed yet. However, the customer already selected the Security Service for configuration purposes as denoted by the `ServiceStartTime` variable. The values of these variables are set by the eHomeConfigurator extended for these purposes as aforementioned.

All process instances have different states they can go into. The corresponding information about the states can be found at the left hand side above the variables in Figure 6. In this example, the process instance 2601 is in the state `open.running`, which means that the customer is actively interested in the security service. It might be that the customer later deselects the security service and focuses on others. In this case, the state of the corresponding instance would be changed by the eHomeConfigurator to `open.not_running.suspended`. Later on, it can be activated again. If the customer finally decides not to install the service, the corresponding process instance will be deleted.

4 Related Work

In the field of process support, several approaches have been developed. We will show, that the approaches heavily depend on the kind of the underlying process and the handled artifacts. We will start with a comparison of our business process to processes in other domains, namely complex non-repetitive development processes and strong repetitive processes. In the end of this section, we will argue why we have chosen XPDL for the formalization of our business process.

A sophisticated tool supporting complex development processes is the AHEAD system [10]. It focuses on development processes in software engineering and mechanical engineering. The AHEAD system deals – in contrast to most of the other approaches to management of development processes – with products, activities, and resources. It has been shown, that these processes have a highly dynamic character and to a lesser extent a repetitive nature. The process underlies a strong variability and numerous feedback flows. The AHEAD system provides the required flexibility in such development processes by enabling flexible switching between *planning*, *analyzing*, *executing*, and *monitoring* tasks. There is no strict separation between specification and execution phase, which is necessary for development processes, which cannot be completely specified in advance. This stands in contrast to the discussed requirements for the efficient handling of the business process in eHome systems. The huge number of eHome installations in parallel dictates a more foreseeable process definition. By the integration of the tools, this goal has been reached. So, the immanent development process in the business process has now been freed from the strong dynamic aspects. Thus, the tool support offered by the AHEAD system and alike is far beyond the required support in eHome systems. Also, the handling of such flexible and powerful tools is a time-consuming task, which stands in no positive relation to the achieved benefits.

The opposite type of processes can be found at insurance companies. Today, the concluding of insurance agreements follow a strict predefined process. Initial assessments trigger the further handling as contracts within regular bounds and exceptional contracts of high-risk. Depending on the assessment, the contract is guided through diverse steps, which are equal for all contracts of the same type. This stands in clear contrast to development processes. Because of this repetitive nature, insurance companies very often deploy workflow management systems to instantiate these processes, which then automatically assign tasks and process-related documents to resources and can be monitored. Our business process is similar in terms of the repetitive nature, but differs in the active players and the information of interest for the provider. While in the above described processes the workflow management system's main task is to assign tasks and documents to resources internal in a company, in eHome systems the important actions to be taken are the triggering of succeeding tasks *and* the monitoring of process instances in which *also* the customer is involved.

The assessment of tools for workflow process support is mostly based on technical details like underlying database systems, competitiveness of the vendor, or marketing strategy. We prefer a problem-oriented assessment based on the language elements of available standards and the functionality of tools supporting them. For the assessment on the *control flow* level, *workflow patterns* have been discussed as an appropriate means [6,11]. For the selection of the appropriate approach for process specification, not solely the pure number of supported patterns should be deciding. It is more important that all of the needed patterns are supported. The available tool support and means for *data flow* and *resource modeling* has to be taken into account as well. By analyzing our specified business process and different workflow specification languages, it came out that XPDL fulfills our requirements for the specification because it supports all of the required workflow patterns directly (see Section 3.1). In addition, XPDL offers constructs for the data flow and resource modeling. Furthermore, there are several workflow management tools which support the exchange format XPDL of which we selected the workflow engine Shark and the XPDL editor JaWE to have a complete workflow management system.

5 Summary and Outlook

Today eHome solutions are products made on specific request and thus, according to specific requirements. Considering the large number of eHome customer relationships to be managed in a future mass market, it is to expensive to manually consider each customer's specific demands. Rather, it is reasonable to develop an eHome service once and distribute it by an automated business process. Talking about automation, the activities for which the provider is responsible for have been automated. In contrast, the customer interacts with the provided tool to specify his requirements, his wishes, and the structure of his environment to avoid misunderstandings. Thus, the services are configured for different eHome environments by suitable tool support, followed by the automated deployment of the services. This constitutes a competence-oriented procedure.

To benefit from the advantages of a workflow management system, we formalized the process using a suitable workflow process definition language. As a result, the process can be instantiated arbitrarily often by a workflow management system, which allows the provider to monitor and easily manage the different states of his customer relationships. Furthermore, we integrated the tools which support both the customer and the provider by carrying out tasks in the specification, configuration, and deployment phases. These tools have also been extended to interact with the workflow management system. Thus, a glueless and flexible execution of the process has been achieved.

Summarizing, the automation of the business process by reducing the use of experts to deploy eHome services for specific customers results in cost reduction and, consequently, smooths the way for a future eHome mass market.

There are several areas for further elaboration, which will be sketched in the following. In this paper, we described the business process for eHome systems as a repetitive one. This process applies for all kinds of customers, because the dynamics of different customer behaviors are encapsulated inside the corresponding activities. In future, the process could be refined to handle different stereotypes of customers, e.g., laypersons, computer-skilled ones, and so on. These different stereotypes might influence the structure of the process because each of them would need different kinds of provider support. Furthermore, we assumed a single-provider situation. The effects of the collaboration of different providers in the form of a virtual enterprise and their influences on the structure of the process are of interest. In analogy, we considered the customer to subscribe to services for only one environment. It has to be researched whether the process has to be modified if the customer wants to use the subscribed services in different environments. This question is particularly important in virtual eHome environments. As mentioned above, the business process also has some characteristics of a development process, especially in the phases of configuration and uninstallation. This leads to future work on the interdependencies between the development and the business process.

References

1. Kirchhof, M., Linz, S.: Component-based Development of Web-enabled eHome Services. Personal and Ubiquitous Computing Journal **9** (2005) 323–332
2. Kirchhof, M., Norbisrath, U., Skrzypczyk, C.: Towards Automatic Deployment in eHome Systems: Description Language and Tool Support. In Meersman, R., Tari, Z., eds.: On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, Proceedings, Part I. Number 3290 in LNCS, Springer (2004) 460–476
3. Kirchhof, M., Norbisrath, U.: Configuration and Deployment in eHome-Systems. In Regentova, E., ed.: Proceedings of International Conference on Information Systems: New Generations (ISNG 2004), PHASE (2004) 23–28
4. Norbisrath, U., Salumaa, P., Schultchen, E., Kraft, B.: Fujaba based tool development for eHome systems. In: Proceedings of the International Workshop on Graph-Based Tools (GraBaTs 2004). Volume 127 of Electronic Notes in Theoretical Computer Science., Elsevier (2005) 89–99
5. Nagl, M., ed.: Building Tightly Integrated Software Development Environments: The IPSEN Approach. Number 1170 in LNCS. Springer (1996)
6. van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases **14** (2003) 5–51
7. WfMC: Workflow Process Definition Interface – XML Process Definition Language. Specification, Workflow Management Coalition (2002)
8. Hollingsworth, D.: The Workflow Reference Model (WfMC-TC-1003). Specification, Workflow Management Coalition (WfMC) (1995)
9. Schneider, C.: CASE Tool Unterstützung für die Delta-basierte Replikation und Versionierung komplexer Objektstrukturen. Master's thesis, Technische Universität Braunschweig (2003)
10. Jäger, D., Schleicher, A., Westfechtel, B.: AHEAD: A Graph-Based System for Modeling and Managing Development Processes. In Nagl, M., Schürr, A., Münch, M., eds.: Applications of Graph Transformations with Industrial Relevance. Proceedings of International Workshop, AGTIVE'99. Number 1779 in LNCS, Kerkrade, The Netherlands, Springer (2000) 325–339
11. Staab, S., van der Aalst, W., Benjamins, V.R., Sheth, A., Miller, J.A., Bussler, C., Maedche, A., Fensel, D., Gannon, D.: Web services: Been there, done that? IEEE Intelligent Systems **18** (2003) 72–85