An Online Approach for the Service Interaction Problem in Home Automation

Michael Wilson, Evan H. Magill, Mario Kolberg
Department of Computing Science and Mathematics
University of Stirling, Stirling. FK9 4LA
{mew, ehm, mko}@cs.stir.ac.uk

***Abstract*— Home automation is maturing with the increased deployment of networks and intelligent devices in the home. Along with new protocols and devices, new software services will emerge and work together releasing the full potential of networked consumer devices. Services may include home security, climate control or entertainment. With such extensive interworking the phenomenon known as service interaction, or feature interaction, appears. The problem occurs when services interfere with one another causing unexpected or undesirable outcomes. Whereas previous approaches to solving service interaction have focused on the service, the technique presented here concentrates on the device and its surrounding environment, as some interactions may happen through conflicting effects on the environment. The concept of environmental variables is introduced, a variable may be room temperature, movement or perhaps light. Drawing inspiration from the Operating Systems domain, access to the device and environmental variable is controlled. Using this technique, undesirable interactions are avoided.**

I. Introduction

The smart home, or intelligent home, is an active area of research with various trial homes being developed over recent years, including: e2-Home in Stockholm [1] and the Internet Home Alliances OnStar homes in Detroit [2]. To achieve this, a plethora of new networking protocols and networked appliances have been developed for the home to create the home network (Fig. 1).

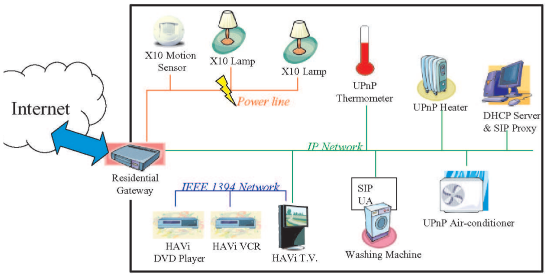

Fig. 1. The Home Network

On their own networked appliances add limited value to the home. Added value comes from services using a series of different devices. These services may include home security, entertainment or energy saving services. Safety services may be used in homes for the elderly or infirm. This allows them to spend more time in their own homes with the knowledge that if anything were to happen to them, the home would alert a carer, thus giving them more independence and improving their quality of life. It is envisioned that these services will be run from a central point within the home, a residential gateway which may take the form of a set top box. The OSGi (Open Services Gateway Initiative) gateway [3] is one such gateway which is able to run and manage services. Devices will register with the gateway, allowing services to use a multitude of devices. For example, instead of a security service only sounding an alarm, if a web camera and a VCR were available the service could record the intruder. If an SMS service was available, the owner could be notified via SMS. OSGi becomes the *glue* which connects devices and services [4]. However, with such interworking between different services and devices, negative interactions are inevitable.

Unexpected or undesirable outcomes are known as Service Interactions, or Feature Interactions[1] [5]. Along with other domains, for example, E-mail [6] and web-services [7], the problem can be found within the home automation domain. This is when the action of one service, or device, has an impact on another. The phenomenon is well understood and documented within the telephony domain as it has been the focus of research for over a decade with work widely published. Indeed a series of workshops have been been held since 1992, with the most recent held in 2003 [8].

In the home an example of service interaction may be between a Home Security and the Climate Control Service. Assume that the Security service is armed. If an air fan is switched on by the Climate Control Service, the fan causes movement which triggers the alarm unnecessarily. If the networked home is to succeed, these kinds of interactions have to be avoided [9].

In this paper a technique for avoiding negative interactions is presented. Whereas previous approaches to solving service interactions have been service centric, i.e., they have focused on the service, our technique concentrates on the device and its surrounding environment. Here, a device may be a heater or fan with an environmental variable being room temperature or movement. By restricting access to both the device and environmental variable interactions can be avoided. Although an earlier paper has been published [10], the technique has

[1] Although there is a difference between a service and feature, the difference is not important in this paper.

evolved and it has been fully implemented as a run-time manager on a OSGi gateway. Here, we concentrate on the implementation of the approach on an OSGi gateway.

The paper is split into six parts. Section 2 focuses on the home network and how services are to be deployed in the home. Section 3 discusses the service interaction issue within the home automation domain. Section 4 discusses our approach in detail and provides examples. Section 5 will explain how the technique has been implemented as a series of OSGi services. Section 6 will present our results. Finally, Section 7 concludes with a discussion on the approach presented along with further work.

II. The Home Network

Increasingly traditional home appliances are being developed with increased software and a network interface allowing them to be connected and controlled by one or a number of networked services. Protocols which these devices use vary depending on the particular device. The X10 [11] protocol is used for simple on and off appliances, UPnP [12] for larger devices with richer functionality and mainly HAVi [13] for audio and visual appliances. Fig. 1 also shows a SIP device and although SIP is primarily concerned with VoIP, extensions have been proposed to the SIP protocol for controlling appliances [14]. To release the full potential of these devices a platform which allows services to interact with devices, regardless of their underlying protocol, is required. The specification defined by the OSGi alliance is one such gateway which could be used within a residential gateway for service deployment.

A. Service Deployment in the Home

The OSGi gateway is a platform which allows service providers to deploy their services in the home, and indeed other environments [3]. The most likely location for such a service gateway in the home would be either in a Broadband modem or a Set Top Box [15].

Services, which are referred to as *bundles* in the OSGi gateway, can be placed onto the gateway, typically, by a service provider remotely. When the bundle is installed it can then be managed (removed, updated, configured, started or stopped) remotely.

When a service is started on the OSGi gateway it is registered with the service registry within the gateway. This allows other services to make use of it. A registered service may be part of a bundle, but can also be the representation of a device. For example, if the gateway had a UPnP driver, as new UPnP devices are introduced to the network the UPnP driver would extract the necessary information from the device. With this information the device would be added the gateway's service registry as a UPnP device service. Immense possibilities are apparent with the gateways ability to import devices from any network, but also to export devices from the gateway to other networks [4]. OSGi removes the barrier between different protocols.

At this point, devices in the home are registered with the gateway and can be used by services within the gateway. The OSGi gateway allows services to choose which devices they would like to use. Since different services are able to freely select which devices they would like to use, and services have been developed independently of one another, service interactions may occur. This problem is discussed in the next section.

III. Service Interaction in the Home

The Service Interaction issue has long been a problem in the telephony domain. The problem is when two, or more, services interact and cause undesirable or unexpected outcomes [5]. The problem is not badly written software, but simply different devices and services with conflicting goals. For example, a goal for one service may be to keep the house secure, while another service may want to open a window for ventilation. Since many services have been developed in isolation, the first time they meet is when deployed in the home [16].

Through our service interaction work, two types of interactions have been identified, *intra-service* and *inter-service* interactions [10]. The former, are interactions within the same service, these types should be discovered at design time of the service. Inter-service interactions are interactions between different services and can only be found at run-time. This is because services are often developed by different businesses and may only meet in the home. Intra-service interactions are also covered by the approach, if the service controls an appliance in a conflicting way.

A number of interactions have been identified within the home with one intra-service interaction and three inter-service interactions highlighted below:

- *Interaction within Home Security* – a feature of the home security service is to make it appear the owner is at home by opening curtains and turning on lights (away from home feature). While the monitoring aspect of the home security is active, the away from home feature must not try and open or close curtains within the home, otherwise the movement will trigger the alarm unnecessarily;
- *Home Security and Entertainment service* – the entertainment service records television shows at certain times and a feature of the home security service is to record intruders using the VCR. If the security service is triggered, the VCR records the intruder from the camera. At this point, the entertainment service must not be granted access to the VCR to record the owners favourite show;
- *Home Security and Climate Control service* – part of the functionality of the climate control service is to open windows within the home at certain times. While the home security service is active, the climate control service must not be allowed to open a window, otherwise the alarm will be triggered unnecessarily;
- *Power Saving and Climate Control* – during cold spells the climate control service keeps the home at a stable, warm, temperature. If the owner is not at home the power saving service will turn appliances off to save energy. In

doing this, it interacts with the climate control service by disabling devices (including heaters) which means the temperature of the home plummets and water pipes freeze causing damage to the home.

The interactions listed above are only a sample of the problems a networked home may encounter. These 'surprises' must be resolved if the networked home is to be a success. The service interaction problem has been recognised in other domains, yet little work has been carried out in the home automation domain. The next section presents a run-time manager which avoids negative interactions.

IV. Avoiding Interactions in the Home

Work has been carried out in the home automation and office domain for service interaction avoidance [9], [17], [18], however, these approaches concentrate on the services. In contrast, the approach presented here focuses on the device and its surrounding environment. The approach makes use of concepts from the operating systems domain where controlling access to resources (e.g. files) is achieved through locking. Drawing inspiration from this domain and adapting the locking technique, a run-time manager has been developed. The manager runs on the same gateway as the home services, intercepting messages to devices, analysing the messages and working out whether the action will cause an interaction. If the action will cause an interaction the message will be rejected, otherwise it will be forwarded to the device.

The technique developed has three layers. The top layer is the service layer which contains the home services. These may include climate control, entertainment or home security services. These services may use one or a combination of home appliances (devices) which are located in the second layer. Devices may include a heater, television or perhaps a thermometer. Finally the bottom layer is the environmental layer containing environmental variables which are a representation of the rooms' environment. An environmental variable may be room temperature, room movement or perhaps room lighting levels. By using these variables, it is possible to see how a device affects its surrounding environment, hence showing how devices interact. For example, a heater device would affect room temperature, therefore if a heater is active room temperature would increase. An air conditioner, when active, would decrease room temperature, therefore, it would be undesirable to have both active at the same time as they have conflicting goals. By using the environmental variables we can see the links between devices, further, by controlling access to the variables it is possible to avoid interactions.

Controlling access to a device or an environmental variable is achieved through locking. If a service wishes to lock a device or a device wants to lock a variable, they must be locked with one of four options:

- *NS* : Not Shared. The variable or device is locked and may not be altered by any other device or service;
- *S+* : Shared, but increase only. The variable is shared on the condition that anyone wishing to use the variable must increase it, e.g., allow two heaters to operate;
- *S–* : Shared, but decrease only. Like the previous setting, the variable is shared on the condition that anyone wishing to alter the variable must decrease it;
- *S±* : Shared. The variable or device is shared and it is unknown whether the variable will be decreased or increased in value. This can also be used for binary values, e.g., whether there is movement or not.

In telephony the notion of a session is clear, the session starts when the receiver is lifted off-hook, and finished when the handset is placed on-hook. In the home domain, the notion of a session is less clear. This approach assumes that a session begins when a service starts using a device, e.g., opening a window or turning a heater on, and finishes when the service closes or switches the device off. Therefore, when a lock is placed on a device, the lock is valid until the service turns the device off, in a similar fashion to how an operating system would handle a lock on a file.

The service interaction manager has an internal representation of devices and their states, also, which service is using which devices. The representation is kept up-to-date and consistent in two ways, firstly, by monitoring the gateway for new devices. When a device is added to the gateway, the manager determines which type of device has been added and consults an XML file held at a remote site that determines which variables the device affects. With this information the device is added to the manager's internal view. Secondly, after an action message sent to the device has been intercepted, analysed and authorised, the devices' and environmental variables' new states are recorded.

Service priorities have been introduced to the technique. This is to allow safety services to override trivial services. For example, if the home was being burgled and the VCR was in use by the entertainment service to record the owners favourite show, it would be more beneficial for the home security service to terminate the recording of the television show and record the intruder. It is anticipated that the service provider, or user, would assign priorities to services.

Using the approach described above, an example is worked through. An inter-service interaction has been chosen, where the services have been developed in isolation and are meeting for the first time at runtime. The example shows how the approach avoids the interaction if the home security service is active first and the climate control comes in second, however, it goes on to discuss how priorities can be used if the climate control service is active first.

A. Interaction between Climate Control and Security

Under certain circumstances an interaction occurs between the home security service and the climate control service. When armed the security service monitors the home for movement and rings an alarm bell when an intruder is detected. The climate control monitors room temperature, opens windows, controls heaters and air conditioners accordingly. If the alarm is set it is not appropriate to open a window, as this would cause an interaction.

Fig. 2 shows how the interaction is avoided using the approach. In the figure there are two services and seven devices. The thermometer is a sensor device, this means it does not affect a variable, but simply monitors a variable. The heater and air conditioner both have a link to room temperature, as they will affect this when active. When open, the window (which is motorised) will affect movement and room temperature. Although the climate control service may use the heater, the link (arrow) between climate and heater is not shown until the service actually uses the device. The alarm control panel is used by the owner to set the alarm. The motion sensor, which is a sensor device, monitors movement and reports to a service. Lastly, the alarm bell has a link to room sound. These links are shown by the arrows in Fig 2.

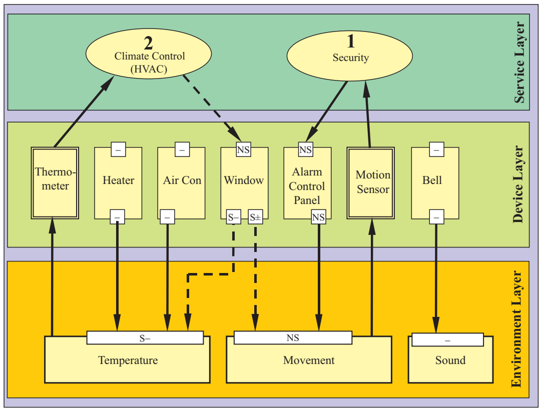

Fig. 2. Avoiding Interaction between Climate and Security service

When the security service is armed the alarm control panel locks the movement variable with *NS* as any movement is interpreted by the service as an intruder. The home is secure and the alarm is fully armed. If the home gets too warm the climate control may try to open a window to let some air in. The climate control gains access to the window device, which is available. Next, the window device must gain access to the environmental variables. The system determines the outdoor temperature to be cooler and *S*– is used for the room temperature variable. As this variable is unused, access is granted and is set with *S*–. However, the window must also gain access to the movement variable. If the window is to open it will create movement, therefore *S*± is used. However, the movement variable is locked with *NS* and since climate control has a lower priority than security it can not be overwritten, therefore access has not been granted. The window is unable to open and the interaction has been successfully avoided.

The example described above assumes the home security service is set first. Assume the home security service is inactive and the climate control service is currently active and has opened the window. The security service requires windows to be closed before the alarm can be set, to achieve this, the security service has to override the climate control service, to do this, priorities are used.

Fig. 2 shows two services, Climate Control with a priority of 1 and Security with a priority of 2. Since security has a higher priority it will be able to override what the climate control does. If climate control has access to the window, the window device would be set with *NS*, and the variables temperature and movement would be set with *S*– and *S*± respectively. At this point, the window is able to open and close. When the alarm is to be set, the window device is already locked with *NS* by the climate control service, however since security has a higher priority, it is allowed access and is able to close the window. The variables are now set as shown in Fig. 2. If the climate control service does try and open the window again, since the environmental variables are locked, the request will be denied as security has a higher priority.

The technique discussed has been implemented as a series of services and bundles on an OSGi gateway. It has been found to work successfully in the laboratory. Positive interactions are allowed, allowing devices to work together, for example, two heaters heating a room, and negative interactions are avoided.

V. Implementation of the Technique

The technique has been implemented in Java (Sun J2SE) using the Eclipse IDE and executed on IBM's OSGi implementation, Service Management Framework (SMF) [19] version 3.5.1. In total four bundles for the Feature Interaction Manager have been implemented. The dark rectangular boxes in Fig. 3 are the implemented bundles, the lighter dashed rectangular boxes are examples of bundles which could potentially be installed.

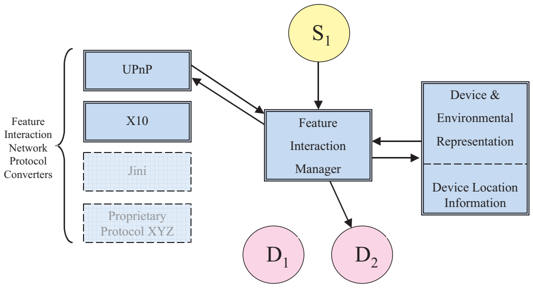

Fig. 3. The Technique within the OSGi Gateway

Fig. 3 shows the implemented technique within the gateway. The rectangular boxes represent components of the feature interaction manager and the figure shows a request from a service (S_1) being sent to the device (D_2). The request has been intercepted by the manager, analysed and as no interactions have been detected the message is forwarded to the device. During the analysis stage inside the manager, the original message is parsed and passed through various bundles.

In the centre of Fig. 3, the Feature Interaction Manager resides. The primary role of this bundle is to intercept messages being sent to the device and decide whether an interaction will occur. Messages received by the manager have to be changed from a protocol specific format to one the manager can understand. It is the role of the *Feature Interaction Network Protocol Converters (FINPC)* to carry out this task.

On the left of Fig. 3 there are four protocol converter bundles. When a new networking protocol is added to the network, for example HAVi, a HAVi FINPC bundle would be required for the manager to decipher HAVi messages. A secondary role of the FINPC is to inform the feature interaction manager of any change in their devices, for example, if an UPnP device is switched on by the owner (or plugged into the network), the change of state would be received by the module and sent to the feature interaction manager which can update the internal representation accordingly. If an X10 devices' state changes, an update message is passed from the X10 FINPC to the manager, which updates the internal view held by the Device and Environment component.

On the right of Fig. 3 is the Device and Environmental Representation (DER) together with the Device Location Information (DLI). Essentially, it is the DER component which authorises a message and works out whether an interaction will occur. The feature interaction manager bases its decision on the result from the DER component. The component has two important roles. Firstly, the DER component must handle the addition and removal of locks depending on what information the feature interaction manager sends. Secondly, the DER component listens for new devices registering with the gateway. When this occurs, the component will try and discover which room the device is in and of what type the device is, both achieved by interrogating the device. When room and device type are known, an XML file is consulted and returns environmental variables this device affects. At this stage, room, device type and variables are known and this information is placed in the internal representation, shown in Fig. 4. If the DER component can not work out which room the device is in, or what type of device, the user can manually input this information via a web interface. If this is a completely new device which the XML file does not recognise, the user can input the type of device and which variables the device will affect, again via an easy to use web interface. These new details will be added to the XML file for future use. It is envisioned that having the user enter details of environmental variables would be a rare event as the XML file will be comprehensive and contain most household devices.

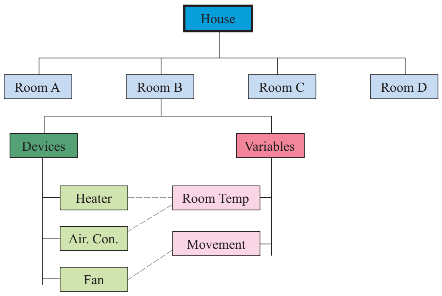

Fig. 4. The Hierarchical Structure Inside the Device and Environmental Component

The DER representation adds devices to its internal view, similarly, when a device is removed from the gateway, the device is removed from the internal representation and any locks the device may hold on environmental variables are released.

The internal representation held by the DER component is of a hierarchical structure, see Fig. 4. At the top level is the *House*, within a house there are *Rooms* and within rooms there are *Devices* and *Environmental Variables*. Fig. 2 shows a more detailed relationship between the device and environmental variable, Fig. 4 simply shows pointers between devices and variables.

Part of the DER component is the Device Location Information (DLI). This information can potentially play an important role for other services. For example, if an entertainment service required a television in the kitchen, it could consult the DLI and a device object would be returned which may be used.

A number of services have been developed, including those to recreate the scenarios depicted in section 3. When these services and devices were added to the gateway, the DER representation correctly identified the devices adding them to its internal representation. Next, services issued commands and the manager intercepted the messages, analysed them and correctly identified interactions.

Importantly, since this is a run-time approach the processing time of a message has to be short. Through experimentation, it has been found that the differences in time between executing an action with and without the service interaction manager was less than one second, however, this is without using any optimisation techniques which requires further research. Current results from testing are discussed in the next section.

VI. Results

Several services have been developed for testing. The Climate Control Service (CCS) keeps the home at a comfortable temperature by opening windows, controlling heaters and air conditioners. The Home Security Service (HSS) monitors the home for intruders, however, has an additional feature which makes it appear the owner is at home when they are actually away (away from home feature). The Entertainment service (ES) records the owners television shows at certain times. Finally, the Power Saving Service (PSS) turns appliances off when the owner is not at home to save energy. Table I shows interactions within, and between, these services and whether the approach is able to avoid these interactions.

The column on the right of Table I shows whether the approach avoided the interaction. A tick ($\surd$) implies the approach did avoided the interaction, however, a cross ($\times$) means the approach did not detect the interaction. There was only one instance where the approach failed. Currently, the approach can not detect loops. In this particular scenario within CCS, the heater is turned on for a certain period of time and switched off. As soon as the heater is switched off the air conditioner comes on. When the air conditioner goes off, the heater then comes back on. The loop continues. The approach does not detect this interaction as the temperature variable is unlocked when the device is set to off, and locked when the device is on. However, since this is an intra-service interaction, the problem should be spotted by the service developer at design time. Work to detect and avoid loops is underway.

Service	*Interaction Description*	*Avoided By Approach*
Within CCS	• Both heater and air conditioner active. • Open window and air conditioner active. • Continuous loop between cooling and heating.	√ √ ×
Within HSS	• Away from home feature and monitoring feature.	√
CCS & HSS	• Fan turned on, triggers alarm.	√
HSS & ES	• Both services require VCR device.	√
PSS & HSS	• PSS not allowed to turn sensors off.	√
PSS & CCS	• Turns off heater, CCS can not raise temperature, water pipes would get damaged.	√

TABLE I

Interactions and the Approach

VII. Conclusions

There are a plethora of sophisticated protocols used for home networking. However, the service interaction issue is a real threat in this domain. If the home of the future, or smart home, is to be commercially viable it has to work, customers will not accept surprises such as the ones outlined in this paper.

We have presented a technique which we have shown to work, and we have described how the technique has been implemented as a series of OSGi services. More testing is required with more services, however results so far have been encouraging. Some further work is still required to investigate the relationship between environmental variables, such as an increase in temperature may also increase humidity and work is at an early stage on this. Also, the relationship between rooms still needs further investigating. The technique does support this, but full testing is still to commence. Using a remote XML file to hold details of devices and environmental variables may be reconsidered and replaced with a database to hold details. This may prove quicker and more scalable.

Also further work exploring ways of exporting the approach to other domains, such as telephony are underway.

References

[1] e2 Home. http://www.e2-home.com, viewed: 12/08/2004.

[2] OnStar at Home Pilot. http://www.internethomealliance.com/pilots_projects/family/onstar_at_home/, viewed: 05/07/2004.

[3] OSGi: The Open Services Gateway Initiative. http://www.osgi.org.

[4] C. Kurzke P. Dobrev, D. Famolari and B.A. Miller. Device and service discovery in home networks with OSGi. *IEEE Communications Magazine*, 40(8), 2002.

[5] E. J. Cameron, N. Griffeth, Y.-J. Lin, M. E. Nilson, W. Shnure, and H. Velthuijsen. Towards a Feature Interaction Benchmark for IN and Beyond. *IEEE Communications Magazine*, 31(3):64–69, March 1993.

[6] R. Hall. Feature interactions in electronic mail. In *[20]*, pages 67–82, May 2000.

[7] M. Weiss. Feature interactions in web services. In *[8]*, pages 149–156, June 2003.

[8] D. Amyot and L. Logrippo, editors. *Feature Interactions in Telecommunications and Software Systems VII*. IOS Press (Amsterdam), June 2003.

[9] X. Wu and H. Schulzrinne. Feature interactions in internet telephony end systems. *Department of Computer Science, University of Columbia Technical Report*, January 2004.

[10] M. Kolberg, E. Magill, and M. Wilson. Compatibility issues between services supporting networked appliances. *IEEE Communications Magazine*, 41(11), 2003.

[11] X10 Technology and Resource Forum. http://www.x10.org.

[12] UPnP: Universal Plug and Play Forum. http://www.upnp.org.

[13] HAVi: Home Audio Video Interoperability. http://www.havi.org.

[14] S. Moyer, D. Marples, and S. Tsang. A protocol for wide area, secure networked appliances communication. *IEEE Communications Magazine*, 38(10), October 2001.

[15] C. M. de Jong J. J. B. Kwaaitaal F. T. H. den Hartog, M. Balm. Convergence of residential gateway technology. *IEEE Communications Magazine*, 42(5), May 2004.

[16] M. Calder, M. Kolberg, E. H. Magill, and S. Reiff-Marganiec. Feature interaction: A critical review and considered forecast. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 41(1):115–141, 2003.

[17] M. Kolberg, E. Magill, D. Marples, and S. Tsang. Feature interactions in services for networked appliances. In *IEEE International Conference on Communications (ICC-2002), New York, USA.*, April 2002.

[18] A. Metzger and C. Webel. Feature interaction detection in building control systems by means of a formal product model. In *[8]*, pages 105–122, June 2003.

[19] IBM Service Management Framework (SMF) 3.5.1. http://www-306.ibm.com/software/wireless/smf.

[20] M. Calder and E. Magill, editors. *Feature Interactions in Telecommunications and Software Systems VI*. IOS Press (Amsterdam), May 2000.