Service Interactions Beyond IN: The new Challenge for Multimedia and Convergence*

Wolfgang Kellerer[1], Peter Sties[1] , Peter Moritz[2]
Email: {kellerer|sties}@ei.tum.de, Peter.Moritz@icn.siemens.de

[1] Lehrstuhl für Kommunikationsnetze, Technische Universität München, Arcisstr. 21, 80333 München, Tel.: +49 89 28 92 35 05, Fax: +49 89 28 96 35 05

[2] Siemens AG ICN CA IN, Otto-Hahn-Ring 6, 81739 München

Summary

Interactions between services and features are a challenging problem in the development and operation of telecommunication services. Multimedia services and the convergence of services from different areas like Internet or broadcasting intensify the problem. This paper brings new problems to the front that arise by new, comprehensive services. Many example scenarios of service interaction problems are shown and a systematization is presented. After a detailed discussion, possible solutions are described.

Résumé

Les interactions entre les services sont l'un des principaux problèmes rencontré lors du développement et de l'utilisation des services de télécommunication. Cette difficulté est encore accrue par le développement simultané de services dans des domaines différents et par l'implémentation multimedia. Le présent rapport décrit et présente de manière systématique les nouvelles problématiques générées par les nouveaux services globaux. L'étude exhaustive de ces questions sera suivie d'une présentation des solutions possibles.

1 Introduction: New Aspects of Service Interactions

Service interactions specify effects between interworking services within communications systems. Thereby unexpected and unwanted behavior can occur, that injures the service functionality.

This paper describes the problems of service interactions, which are an ongoing problem of the service engineering discipline. Up to now, the work on service interactions and problem solutions focuses on supplementary telephone services. New aspects in service interactions, which arise from multimedia and the convergence of services from telecommunications, data communications and broadcasting, have not really been taken into account yet.

In addition to the well known causes of service and feature interactions, such as limited signaling support or address resolution problems, we will in the following examine interactions that result from the recent evolution in informations- and communications services and their future development. *Multimedia* services cause unwanted interactions that result from the combination of different media and their QoS requirements within one service. The *convergence* of services and networks in the areas of telecommunications (e.g. phone, fax), data communications (e.g. email, ftp) and broadcasting (e.g. TV, radio), leads to new, so far unknown interaction problems that arise from the move of services to a new context [SHJ+98]. Another source of interactions in this respect is the *deregulation* of the telecommunications market. Within a deregulated informations- and communications market, services are provided by more than one interworking provider and provided services are crossing domain borders. Therefore services can be interpreted and executed differently in the different contexts.

* This work was supported by the Bayerische Forschungsstiftung and the Siemens AG as part of the research consortium "Bayerischer Forschungsverbund Software Engineering (FORSOFT)" in the project C2 "Service Engineering".

The paper is organized in the following way. First we present new interaction scenarios and systematize their relevance in the service software life cycle. After a general introduction of feature interaction solutions in Section 3, we present a strategy for interaction handling by interaction avoidance. A conclusion finalizes our paper.

2 Service Interaction Scenarios

The service and feature interaction issue is, as we have shown above, not a strictly determined problem but produces a variety of questions and problems throughout the service life cycle. Some problems are well understood as we can see in the work on feature interactions in telephone systems (e.g. [KCV98]). But even there, unsolved problems remain. Others remain a great challenge for future research. Therefore it is necessary in the first way to describe the new problems in interaction scenarios and to derive general ways for solutions [CL98].

2.1 Feature Interaction Benchmark

According to the feature interactions benchmark of Cameron et.al. [CGL+93], which describes a comprehensive overview of feature interaction scenarios in telephone and IN systems, interactions can be systematized by their causes in i) violation of assumptions, ii) limited network support and iii) problems of distributed systems.

Violation of assumptions: Assumptions about feature behavior are broken when a feature's context is changed by another feature.

For example: Terminating Call Screening (TCS) prevents calls coming from dedicated users. Automatic Call Back (ACB) prepares the last incoming call, that has not been answered, for callback. A user, who has activated both of the two features, can be faced with the situation, that he automatically calls back a user who is actually on his screening list. Both features process correctly but the ACB alternates the context for the TCS, since it transforms an incoming call in an outgoing call, which is permitted. The assumption of being shielded from dedicated calls is broken.

Limited network support: Interactions are caused by limited communication between network components and poor signaling capabilities of the terminals.

The set of signals of a common telephone set is limited to ten digits and a few other symbols. Many telephony features come in conflict when different features interpret the same signal e.g. flashhook as a trigger signal.

Mapping and re-mapping of signaled information elements at the borders of different signaling systems is another example where limited network support becomes evident.

Problems of distributed systems: Features are realized in distributed nodes and the coordination and synchronization is insufficient.

For example: User A and user B have activated Call Back on Busy Subscriber (CBBS) and Automatic Call Back (ACB) respectively. The activation of both features for the same call can lead to a line blocking, since both lines are constantly busy due to the (failing) features' call attempts.

2.2 New Scenarios

In addition to this benchmark of interaction scenarios, new causes, which have not been considered yet, arise by *multimedia* services, service and network *convergence* and *deregulation*. The following examples are selected from a more comprehensive collection of benchmark scenarios in [Kel99].

2.2.1 Multimedia

The combination of different media of varying quality within one service and multi-party calls bring a number of potential conflict areas. Interactions caused thereby go beyond the traditional speech oriented telecommunication systems.

Multimedia and QoS: (Example 1) Certain high priced services guarantee a high quality to the subscriber. If the QoS in a multimedia conference exceeds the preset limit, e.g. due to an overload, the conference will be interrupted or even terminated. The conference participants probably would have agreed to a slightly reduced quality, if they were asked.

Multimedia and Compatibility: (Example 2) The subscriber of a multimedia information service (e.g. personal newspaper), who receives information via a connectionless push mechanism, changes decoding formats of his terminal. Neither the subscriber nor the service provider notice that information is lost by this configurational change. Information, the subscriber is eventually charged for.

Role conflicts: (Example 3) The conference coordinator of a multimedia conference receives a second call. He leaves the conference (suspend) and responds to the second call. During this time all the features of the conference coordinator are disabled, e.g. acceptance of a new participant.

2.2.2 Convergence

On the one hand convergence means the combination of services or features of the areas of telecommunication (phone, fax), data communication (email, WWW) and broadcasting (TV, radio), which were traditionally separated. On the other hand the convergence also includes all problems of heterogeneity, since networks and terminals have to support new service combinations.

New service combination: (Example 4) With interactive television the subscriber is able to set bookmarks within certain programs, to request further information from a corresponding subject, or to obtain markers on succeeding relevant programs. Since the bookmark is set on a sequence of pictures (e.g. within a news spot) it is difficult to ensure that subscriber's requests are fulfilled by the information, pointed at by the bookmark.

Multimedia terminals: (Example 5) PC based terminals are normally equipped with a screen saver function that switches off the display if the terminal is not in use for a certain period of time. The use is determined by keypad or mouse/cursor-actions. This works well for interactive services that are based on a data exchange and require user input. However, it is a problem for services that are working with audio or video, such as video telephony or video-on-demand. It is more than inconvenient if the display switches to black during a video presentation. Imagine the alert of a video telephone call, where the called user has to deactivate the screen saver by inputting his password.

This interaction problem occurs, because features of the terminal and the communication services have converged, and have moved the services into another context that they were not designed for.

Performance: (Example 6) The execution of several services in parallel on one terminal, may cause dramatic decrease in performance. This affects the QoS that is guaranteed for running services and may even cause the release of a service, although the service provider still fulfils his QoS guarantee.

2.2.3 Deregulation

The market for communication and information services is currently reorganized by deregulation efforts to the principles of a free market economy. Many different providers cooperate or compete in the deployment of services. Missing interworking rules or proprietary interfaces cause wrong behavior of services.

Multi-Provider: (Example 7) The activation of the callback-on-busy-subscriber feature during a call via a Call-by-Call provider has the result, that the callback is established via the standard or preselect provider, since the service is agnostic of a Call-by-Call prefix.

Interworking: (Example 8) In Germany the introduction of Call by Call phone service providers, lead to billing problems, for example in hotels, since the charge indication tone was missing in non-Telecom calls.

2.3 Software Life Cycle

Today's systems for information and communication services are highly distributed and complex systems, which have to cope with periodic changes and adaptations. The software part of the systems is very high. Therefore we will have a closer look on the software system and its phases of development.

Since service and feature interactions represent a class of service specific software problems, we will systematize the service interaction scenarios, according to the main phases of the software life cycle. In the following, typical interaction problems in each phase are described.

- The *requirements specification* lacks in describing the functionality of a service, so that its functionality within the context of other services remains undefined. A well-known example from IN features is the concurrent usage of call forwarding and call screening [Fri95]. See also example 5.
- The system specification (*service design*) defines the behavior of the complete system and its components. Here, some functions are (implicitly) defined, that are not part of the requirements specification. These functions must not lead to inexact behavior. See examples 1, 3 and 7.
- Wrong behavior in the interworking of different services i.e. interactions can result from the mapping of the service design on the *implementation environment*. Unexpected behavior occurs when a running service conflicts with others. See examples 2 and 8.
- The *execution* of services leads to dynamic conflicts in the service usage. Interactions are caused for example by the concurrent use of service resources, which are capable of concurrent access, but lack in performance or real-time response for complex services. See example 6.
- Wrong or misleading assumptions made by the user regarding service functionality are caused by inaccurate *documentation*. Such a gap has been shown in example 4. The rising number of upcoming services will increase the problem of misleading documentation as a main cause for interaction problems. And a second point adds to this issue: Services are intended for an intuitive use, but the short time to market leaves little time for clear design or accurate documentation, and thorough acceptance testing.

The classification of interactions by the phases *requirements, design, implementation, execution* and *documentation,* supports the elaboration of adequate solutions in each of the phases. With respect to a method based approach in software engineering, it is easier to have strategies for solving interaction problems dedicated to certain phases of development, and to integrate them into the development method.

3 Solutions to interaction issues

The approaches to resolve feature interactions can be classified by interaction detection, interaction resolution and interaction avoidance [BV94]. We propose avoidance to be the method for future interaction handling.

3.1 Interaction Detection

Basically, the detection of interactions can be divided into off-line and on-line detection. Off-line detection approaches rely on suitable modeling of the services in the service specification phase, e.g. by formal description techniques as SDL [KCK94], LOTOS, Petri-Nets. By the use of software tools interactions can be detected by automated validation, verification and simulation.

The state space of complex systems, by exhibiting a large number of variables and parameters, very quickly becomes too large for tool-based explorations. Here, on-line interaction detection has to be used, which only takes the current parameter settings for the analysis.

3.2 Resolution

After having identified interactions, appropriate actions for their resolution have to be undertaken. The service software has to be reprogrammed or even redesigned (off-line). Approaches for on-line interaction resolution are based on decision tables, role models [Fri95], agent systems or the supervisory control theory [CLL97].

3.3 Avoidance

Interaction avoidance means to reduce the number of potential interactions by the way the software architecture is designed. Open system architectures, adapted signaling protocols, service creation guidelines and service creation environments, are means for interaction avoidance as well as exact specification methods, which are especially designed for service engineering. Unfortunately there is only little research following this approach [CL98].

As can be seen from the above scenarios, the service and service control software will increase dramatically. Methods based on the formal description and evaluation of the state space will very quickly reach their limits. Therefore in our work, we propose the strategy to avoid interactions as early as possible in the service engineering. In the long run this is the best way to cope with the increasing complexity issue.

3.4 Avoidance by Formalized Requirements

First it is necessary to reduce problems that have their origin in the *requirements specification* to a minimum.

In order to avoid all interactions, caused by misleading interpretations of the service functionality (see examples 4 and 5), a formalized description method of service requirements is needed. Though the lacking of such formalized description method is well understood by the industry so far no relevant approaches are known that can solve this problem comprehensively [CL98].

3.5 Designing Interaction Avoiding Architectures

The use of dedicated architectures in the *specification and design phase*, avoids the upcoming of a number of interaction problems.

Traditional telecommunications architectures like POTS or ISDN do not separate a service or a feature control nor they have interfaces for service creation. Thus feature updates due to interaction problems cause enormous costs. The Intelligent Network (IN) architecture separates service control from call control and provides a flexible service provisioning. Several research projects focus on feature interactions of the IN [KCV98]. But the IN concept - as currently standardized - only deals with telephony services and features. Concerning the above sketched scenarios we have to look on further architectural concepts.

3.5.1 Layered Architectures

First of all for interaction avoidance, the complete separation of the service control from the network and connection control, is essential to overcome drawbacks in the modeling of connections and call states (e.g. the limitations in signaling). Regarding interactions, the IN still suffers from its call model, which limits the modeling of future multimedia services and causes interaction problems.

A clear separation of the layers of signaling protocols (service control, call control, resource control and connection control) contributes to systems built up by independent components. This is essential for services which base on the convergence of different networks. Also conflicts caused by different providers are overcome by precise interfaces. (See examples 7 and 8)

The present trend towards a *Layering by APIs* is in line with this recommendation.

3.5.2 Session Concept

Several studies have proposed the *session concept* to deal with the complexity of information and communication services. This concept describes a temporal relationship, in that all communication relationships of all involved parties, belonging to one service instance, are controlled by one central management component (session manager). Furthermore *service sessions* are distinguished from *access session* or *communication sessions*.

In this way the concentration of all session context in the session manager provides the necessary overview to handle service interaction conflicts caused by different active features. Also role conflicts (example 3) can be avoided by the central session management.

On the other hand the separation of concerns by different kinds of session system reduces complexity. There is not one single super session manager which is responsible for all services, but there are sessions dedicated to users (access session), service session instances controlling the specific usage of a service, and communication sessions controlling the information flow independently.

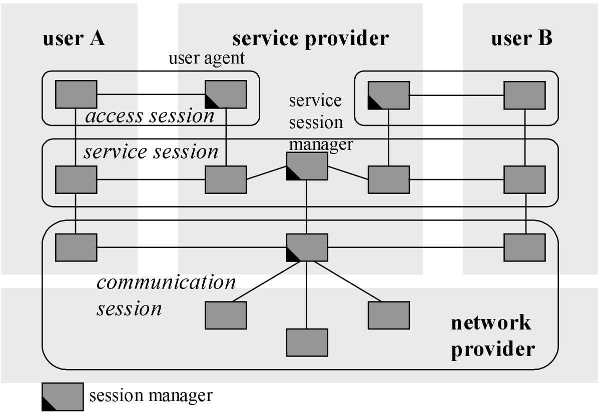

Figure 1: The *session* architectural concept

3.5.3 Object Orientation

Session based architectures, like the Telecommunication Informations Networking Architecture [TINA], are composed of interacting objects. Each object is dedicated to perform certain capabilities. The session manager for example is such an object.

Other objects represent participants in a communication service system. These so called user agents act on behalf of the user within the services provider system domain. They have full knowledge of the user‘s communication capabilities and preferences (e.g. concerning Quality of Service). Thus interaction problems resulting from incompatibility (see examples 1, 2 and 7) can easily be handled [KM98].

Summarizing, we can see that a strict distribution of functions to dedicated components within a service architecture will not only reduce software complexity, but also help avoiding interactions in an early phase of the software development life cycle, the design phase.

3.6 Implementation

To avoid (remaining) interactions in the implementation phase, it is desirable to automate implementation by appropriate software tools and code generators.

4 Conclusions

We have presented a number of new upcoming problems concerning service and feature interactions in the future information and communications services world. The description of our service interaction benchmark provides the basis for the approaches to deal with the problems. Since the service and feature interaction issue in fact consists of a number of problems caused by quite different reasons in different software development phases, it is first necessary to show and classify upcoming obstacles in service engineering.

After a short introduction of approaches for interaction handling, we propose the strategy that interaction avoidance is the main means to cope with the increasing complexity of services and of service interactions. Some studies for open component based service architectures already support concepts that are suitable to overcome interaction causes. Especially the introduction of a feasible session concept will avoid interactions that result from call modeling and call control limitations of conventional service architectures. Since the study of appropriate architectures to avoid interactions is still a neglected aspect [KM98], research has to be focused on this area.

5 References

[CGL+93] J. Cameron, et. al. A Feature Interaction Benchmark for IN and Beyond. IEEE Communications Magazine, Vol.31, No.3, March 1993, pp.64-69.

[SHJ+98] U. Schön, et. al. Convergence Between Public Switching and the Internet. IEEE Communications Magazine, Vol.36, No.1, January 1998, pp.50-64.

[KCK94] B. Kelly, et. al. Feature Interaction Detection using SDL Models. Proceedings of IEEE Globecom, 1994, pp.1857-1861.

[CLL97] Y. Chen, et. al. Resolving Feature Interactions Using Modular Supervisory Control with Priorities. Proceedings of FIW IV, 1997, pp.108-122.

[Fri95] N. Fritsche. Runtime Resolution of Feature Interactions in Architectures with separated Call and Feature Control. Proceedings of FIW III, 1995, pp.43-63.

[KCV98] K. Kimbler, C. Capellmann, H. Velthuisen. Comprehensive approach to service interaction handling. Computer Networks and ISDN Systems, Vol.30, No.15, Elsevier Science, 1998, pp. 1363-1387.

[TINA] TINA-C Deliverable: Telecommunications Information Networking Architecture - Service Architecture 5.0, 1997. Http://www.tinac.com

[KM98] M. Kolberg, E. Magill. Service and Feature Interactions in TINA. Proceedings of FIW V, 1998, pp.78-84.

[KCV98] K. Kimbler, et. al. Comprehensive approach to Service Interaction Handling. Computer Networks and ISDN Systems, Vol.30, No.15, 1998, pp.1363-1387.

[CL98] J. Cameron, F. Lin. Feature Interactions in the New World. Proceedings of FIW V, 1998, pp.3-9.

[Kel99] W. Kellerer. Service and Feature Interactions – neue Aspekte durch Multimedia und Dienstekonvergenz. Technical Report TUM-LKN-TR-9901. Munich University of Technology, Institute of Communication Networks, Munich, 1999.

[BV94] W. Bouma, H. Velthuijsen. Feature Interactions in Telecommunications Systems (Editorial). Proceedings of FIW II, 1994.