

Kent Academic Repository

Full text document (pdf)

Citation for published version

Blair, Lynne and Blair, Gordon S. and Pang, Jianxiong and Efstratiou, Christos (2001) `Feature' Interactions outside a Telecom Domain. In: Workshop on Feature Interactions in Composed Systems, ECOOP2001, June 2001, Budapest.

DOI

Link to record in KAR

<http://kar.kent.ac.uk/38795/>

Document Version

UNSPECIFIED

Copyright & reuse

Content in the Kent Academic Repository is made available for research purposes. Unless otherwise stated all content is protected by copyright and in the absence of an open licence (eg Creative Commons), permissions for further reuse of content should be sought from the publisher, author or other copyright holder.

Versions of research

The version in the Kent Academic Repository may differ from the final published version.

Users are advised to check <http://kar.kent.ac.uk> for the status of the paper. **Users should always cite the published version of record.**

Enquiries

For any further enquiries regarding the licence status of this document, please contact:

researchsupport@kent.ac.uk

If you believe this document infringes copyright then please contact the KAR admin team with the take-down information provided at <http://kar.kent.ac.uk/contact.html>

'Feature Interactions Outside Telecom Domain

Lynn Blair¹ Gordon Blair¹ Jianxiong Pang² Christof Efstratiou²

¹ Department of Computer Science Faculty of Science, University of Tromsø N-9037 Tromsø Norway.
Tel: +47 76 5209 Fax: +47 76 4480

² Computing Department Lancaster University Bailrigg Lancaster LA4 4YRU.K.
Tel: +44(0)1524 593802 Fax: +44(0)1524 593608
email {lgordon, efstrati}@comp.lancs.ac.uk pang@lancaster.ac.uk

Abstract. Feature interactions in the original sense of the term (i.e. within a telecommunications domain), have now been the subject of significant research activity for over ten years. This paper considers several different sources of interaction in the domain arising during the course of our research at Lancaster. These interactions are taken from a variety of areas within the field of Distributed Systems and stand to benefit greatly from the application of techniques developed in the feature interaction community. Furthermore, we believe they represent a potentially important generalisation of feature interaction research.

1 Introduction

The term *feature interaction* can simply be viewed as an “interference between services or features” [Calder00]; more specifically, such an interaction occurs when “the behaviour of one feature is affected by the behaviour of another feature or another instance of the same feature” [Kimble95]. Several taxonomies have been proposed in order to try and classify different types of interaction (including [Cameron94] [Kimble95] and [Hall98]). As simple, yet we believe helpful distinction from [Kimble95] is between:

- interactions that occur because the requirements of multiple features are not compatible and
- interactions that occur when a feature behaves differently in the presence of other features.

Within the telecommunications domain, there are numerous well-documented cases of feature interactions. For example, we refer to the series of workshops in Feature Interactions in Telecommunications (and Software Systems), e.g. [Dini97] [Kimble98] and [Calder00]. However, recently it has become increasingly obvious that research into methods to detect and solve such interactions in telecommunications systems is also of great significance outside the telecom domain. In fact, as recognised in [Calder00], “the subject has relevance to any hardwired system where separate software entities control shared resources”. Furthermore, interactions can often be traced back to the fact that “two features manipulate the same entities in the base system and in doing so violate some underlying assumptions about these entities that the other features rely on” [Plath98].

In our earlier position paper [Blair00], we describe some interaction problems arising from Internet-based and multimedia/mobile systems that we describe in detail in the recently funded “FILBETT” project Feature Interactions: Life Beyond Tradition. This current paper extends our earlier one by providing new examples of interactions that

¹ Currently on leave from the Computing Department Lancaster University.

have arisen in research within the Distributed Multi-University, in addition to cataloguing few other in the literature.

In the remainder of this paper, we first provide a brief project below (section 2) and then document some of what we have come across (section 3). Finally, we discuss some of the detection and resolution of feature interaction (section 5).

2 FILBETT: Lif Beyond Traditional Telephony

2.1 Overview

Motivation for this project came from a number of examples we encountered when looking at Internet-based and multimedia services. As expressed by Heilmeier, "the telecom industry is quickly evolving from 'POTS' (plain old telephone services) to 'PANS' (pretty awesome new services)" [Heilmeier98]. These new services are able to utilise the power of Internet-based (IP-based) and multimedia/mobilis systems and as the number of new services grows, the potential for interactions between services will inevitably explode.

The project addresses the main goal of the FILBETT project: to consider various new and emerging types of *feature interaction* that are likely to arise from the increasing popularity of *mobile* systems and services. A secondary, but still important, goal is to consider *IP-based* services and *multimedia* services. We plan to use formal modelling and analysis methods in this work, building on earlier work that we have done. Although the project is still in its early stages, some initial interaction examples are presented below.

2.2 Some non-traditional interaction scenarios

A number of interactions were identified in our previous position paper; these are listed below for detail the reader is referred to [Blair00].

- combining traditional telecommunication services with Internet access
- potential interactions with one-to-many services
- TCP flow-control mechanisms and protocol interaction in general
- sharing demand on network bandwidth between browsing and viewing wide-area stream
- interactions occurring with multipoint conferencing units (MCUs)
- mobile resource interactions concerning bandwidth and power management
- problems with TCP over wireless networks

A number of further interactions have been identified in our work at Lancaster on mobile computing (see [Efstratiou00] and [Efstratiou01]). In a summary, most of these scenarios concern conflicting *adaptation policies*. In an attempt to maintain an appropriate level of quality of service, many mobile systems employ various adaptation mechanisms. However, adaptation mechanisms are employed for different attributes. Examples include mechanism to adapt network bandwidth proxy behaviour (e.g. in web browsing) and choice of location sensing mechanism. For example, consider a mobile device that employs two independent adaptation mechanisms: one for managing power and the other for managing network bandwidth. If power is running low, the power management mechanism

media Research Group at Lancaster on-traditional feature interactions from

of details of the scope of the FILBETT the 'non-traditional' interactions we of the major techniques that exist for (section 4) and then draw our conclusions

examples of interactions that we in media/mobilis services. As expressed ing from 'POTS' (plain old telephone Heilmeier98]. These new services are ed) and multimedia/mobilis systems and as nteractions between services will

project is to consider various new goal is to consider 'IP-based' services and 'multimedia' services. We plan to use formal modelling and ork that we have done. Although the al interaction examples are presented below.

ious position paper; these are listed

with Internet access

in general

and viewing wide-area stream

units (MCUs)

power management

in our work at Lancaster on mobile n summary, most of these scenarios us adaptation mechanisms. However, ation mechanisms are employed for adapt/manage power consumption, ng) and choice of location sensing employs two independent adaptation r managing network bandwidth. If will request applications that are

using network bandwidth to postpone this use, so as mode. However, as a consequence, the network adaptation unused bandwidth and will notify applications that direct conflict with the power management adaptation

to place the network device in sleep mechanism will now detect they can use this spare bandwidth, in mechanism!

Note that a further interesting dimension to adapt where by users can express preferences over different her *context*. For example, power management mechanisms may be crucial if the user is working in the field but less important in the office

ion is user-configuration of devices adaptation policies depending on his/her *context*. For example, power management mechanisms may be crucial if the user is working in the field but less important in the office where an alternative power supply exists.

3 Additional Interaction Scenarios

Whilst FILBET is primarily concerned with the generation of mobile, multimedia and P-based services, into such interactions does not stop here. Two further from the application of feature interaction research additional examples of ‘non-traditional’ interaction [Fireworks97] relating to email systems and variable system tape-deck system and network ticketing

alization of feature interaction scenarios is clear that the value of research areas from our work that would benefit are described below. As a side, some ns can be found in [Hall00] and ty of miscellaneous examples (including system etc.) respectively.

3.1 Component-based middleware

At Lancaster, we are interested in component-based middleware platforms such as the CORBA Component Model of CORBA v3, .NET or Enterprise Java Beans. Associated component-based development methodologies focus on the provision of means for specifying individual components together with their composition (i.e. an architecture) [Szyperki98]. By allowing new components to be added, and existing components, we obtain an incremental development model for architectures. However, this raises two key questions:

middleware platforms such as the CORBA Component Model of CORBA v3, .NET or Enterprise Java Beans. Associated component-based development methodologies focus on the provision of means for specifying individual components together with their composition (i.e. an architecture) [Szyperki98]. By allowing new components to be added, and existing components, we obtain an incremental development model for architectures. However, this raises two key questions:

- When we *compose* an architecture, how can we be confident that components will work together that there are no unwanted subtle interactions and that the results are coherent?
- When we *adapt* an architecture, how can we be confident that replacement or updates behave as expected especially in tandem with other components?

Existing component-based methodologies provide little in the way of support for these problems. Typically, architectures are verified in terms of type compatibility between (required and provided) interfaces. In addition, checks may be carried out on the validity of architectures against certain style rules [Shaw96] [Medvidovic00]. However, this is not sufficient to capture the more subtle problems associated with unwanted interaction between components. This is an area that would benefit greatly from feature interaction research. In particular, a *design-time* check could be carried out on initial architectures and expected variations, whereas *run-time* techniques could be used to discover problems after re-configuration and also to catch problems not foreseen from static analysis.

Existing component-based methodologies provide little in the way of support for these problems. Typically, architectures are verified in terms of type compatibility between (required and provided) interfaces. In addition, checks may be carried out on the validity of architectures against certain style rules [Shaw96] [Medvidovic00]. However, this is not sufficient to capture the more subtle problems associated with unwanted interaction between components. This is an area that would benefit greatly from feature interaction research. In particular, a *design-time* check could be carried out on initial architectures and expected variations, whereas *run-time* techniques could be used to discover problems after re-configuration and also to catch problems not foreseen from static analysis.

Adding an extra dimension to this analysis, we are also interested in *reflective middleware* whereby component-based approaches apply not only to the application/service level but also to the structure of the middleware itself [Blair98][Blair01]. Reflection is then used to provide introspection and adaptation of the middleware structure via a *meta-level*. This approach enables middleware to be customized for particular application domains, e.g. a small footprint system for an embedded device, and also to be re-configured if environmental assumptions change, e.g. to change a transport protocol or compression

Also interested in *reflective middleware* whereby component-based approaches apply not only to the application/service level but also to the structure of the middleware itself [Blair98][Blair01]. Reflection is then used to provide introspection and adaptation of the middleware structure via a *meta-level*. This approach enables middleware to be customized for particular application domains, e.g. a small footprint system for an embedded device, and also to be re-configured if environmental assumptions change, e.g. to change a transport protocol or compression

strategy for operating over a wireless link. Essentially, this provides the extra capability of being able to adapt the non-functional properties of an application (real-time performance, security, availability, etc). Again, it is vital to know if there are any unwanted side-effects of the changes, e.g. does the new availability policy conflict in any way with security requirements).

ntially this provides the extra capability of being able to adapt the non-functional properties of an application (real-time performance, security, availability, etc). Again, it is vital to know if there are any unwanted side-effects of the changes, e.g. does the new availability policy conflict in any way with security requirements).

3.2 Behaviour in Co-operative Virtual Environments

PING is an EU-funded project looking at the development of an object-oriented framework for the support of distributed and co-operative virtual environments, which can then be specialised on a per-application basis (IST-1999-11488) Lancaster University is responsible, along with others, with the modelling of behaviour in such virtual environments. The approach is to represent all entities in a virtual environment as passive or active objects. A passive object essentially consists of a set of public attributes that are exposed to other objects in the environment. An active object on the other hand also includes behaviour scripts written in the Junior scripting language (which in Ping is expressed as a series of reactive aspects) [Boussinot01]. We are investigating an such behaviours using Junior, with consideration of (virtual) world physics (gravity, inertia, etc), distributed system policies such as replication and consistency management, reaction to collisions and autonomous behaviour relating to the object. Furthermore, it is important in PING to be able to adapt behaviour as environmental conditions change, e.g. to minimise dissemination of information foraging and adaptation will be modelled as further behavioural aspects, resulting in self-adapting object behaviours (c.f. reflection above).

ent of an object-oriented framework for the support of distributed and co-operative virtual environments, which can then be specialised on a per-application basis (IST-1999-11488) Lancaster University is responsible, along with others, with the modelling of behaviour in such virtual environments. The approach is to represent all entities in a virtual environment as passive or active objects. A passive object essentially consists of a set of public attributes that are exposed to other objects in the environment. An active object on the other hand also includes behaviour scripts written in the Junior scripting language (which in Ping is expressed as a series of reactive aspects) [Boussinot01]. We are investigating an such behaviours using Junior, with consideration of (virtual) world physics (gravity, inertia, etc), distributed system policies such as replication and consistency management, reaction to collisions and autonomous behaviour relating to the object. Furthermore, it is important in PING to be able to adapt behaviour as environmental conditions change, e.g. to minimise dissemination of information foraging and adaptation will be modelled as further behavioural aspects, resulting in self-adapting object behaviours (c.f. reflection above).

Although this is a rather different application domain considered above. In particular, we are concerned about subsequent re-configuration of a platform. In this interaction between behaviours at both an inter- and intra-object level. Importantly, in this work we already have a significant advantage in the semantic (expressed in a set of writing rules) [Boussi

ain, the problem is similar to those considered above. In particular, we are concerned about subsequent re-configuration of a platform. In this interaction between behaviours at both an inter- and intra-object level. Importantly, in this work we already have a significant advantage in the semantic (expressed in a set of writing rules) [Boussi

4 A Brief Summary of Feature Interaction Detection

Existing analysis techniques can be seen to fall into two broad categories: *off-line* (or design-time) techniques, *on-line* (or run-time) techniques and

and Resolution Techniques

Existing analysis techniques can be seen to fall into two broad categories: *off-line* (or design-time) techniques, *on-line* (or run-time) techniques and

With *off-line* techniques, a model of the base system and the additional features are specified in a formal language whilst the properties that the system should exhibit are (typically) specified through a set of temporal logic. A wide range of modelling languages have been used, including Finite State Machines (FSMs), LOTOS, Petri-Nets, Promela and SDL. However, as the number of services grows, there is clearly an issue of the scalability of such techniques and tools. Importantly, though, major improvements have been forthcoming in model-checking techniques recently for example through the use of on-the-fly and symbolic techniques and also through the use of abstraction and symmetry. Such techniques can help to greatly reduce the state-space explosion problem. A further problem however is that the development of successful techniques is dependent on the accuracy of the abstraction and the specified properties. An inaccurate or rather non-precise abstraction can lead to missed interactions (as occurred in [Bousquet99]). Off-line techniques must also rely on a priori knowledge of the behaviour of the individual services and features.

With *off-line* techniques, a model of the base system and the additional features are specified in a formal language whilst the properties that the system should exhibit are (typically) specified through a set of temporal logic. A wide range of modelling languages have been used, including Finite State Machines (FSMs), LOTOS, Petri-Nets, Promela and SDL. However, as the number of services grows, there is clearly an issue of the scalability of such techniques and tools. Importantly, though, major improvements have been forthcoming in model-checking techniques recently for example through the use of on-the-fly and symbolic techniques and also through the use of abstraction and symmetry. Such techniques can help to greatly reduce the state-space explosion problem. A further problem however is that the development of successful techniques is dependent on the accuracy of the abstraction and the specified properties. An inaccurate or rather non-precise abstraction can lead to missed interactions (as occurred in [Bousquet99]). Off-line techniques must also rely on a priori knowledge of the behaviour of the individual services and features.

In contrast, *adaptive on-line* techniques address this latter issue. Such approaches have been developed from a much more pragmatic perspective and have evolved over time to

In contrast, *adaptive on-line* techniques address this latter issue. Such approaches have been developed from a much more pragmatic perspective and have evolved over time to

become increasingly (dynamically) adaptive. Adaptation is powered by a knowledge database, such as predefined data types and user agent rules. For example, in [Griffiths94] unknown new features are accommodated through a 'adaptive agent regime' architecture where a user agent engages in negotiation to settle the discerned conflict between

features. Strategies have typically been tables, state transition rules, abstract

Finally, recognition of the advantages (and certain techniques), a hybrid approach is proposed in [Calder99]. This approach is a resolution of interactions between new services and legacy services and combines a non-line, transactional approach with off-line formal analysis (see also [Marple00]).

drawbacks) both off-line and on-line. This approach is a resolution of interactions between new services and legacy services and combines a non-line, transactional approach with off-line formal analysis (see also [Marple00]).

5 Conclusions

It should be apparent from the discussions above that there are many benefits from results in the field of feature interaction where we believe this to be the case including mobile-based and reflective middleware and also behavioural

many areas of computer science can be identified. We have identified a number of areas including multimedia systems, component-specific interaction in virtual environments.

An interesting first line of research is to consider interaction problems including for example the potential (required/provided interfaces) [Szyperki98]. More generally, further research is clearly required to bring the different research communities, in order to more fully understand the relationships between feature interaction and these other areas.

the impact of components on the feature interaction. Explicit context dependencies simplify analysis of potential interactions, including strong collaboration between fully understand the relationships

Acknowledgements

The authors would like to thank a number of researchers to discussions on the interactions documented above. Adrian Friday and Keith Cheverst for their insights members of the OpenORB project for discussions on middleware (<http://www.comp.lancs.ac.uk/computing/research/mpg/reflection/memb.html>), and Paul Kandath for his thoughts on behavioural interaction in virtual environments.

at Lancaster which have contributed. In particular, we would like to thank into mobile computing, the various component-based and reflective interactions in virtual environments.

References

- [Blair98] Blair G.S., Coulson G., Robin P., Papatomas M., "An Architecture for Next Generation Middleware" Proceedings of the Fifth International Conference on Distributed System Platforms and Open Distributed Processing Middleware '98, Springer, 1998.
- [Blair00] Blair L. & Pang J., "Feature Interactions: Life Beyond Traditional Telephony" In [Calder00] pp. 83-93, 2000.
- [Blair01] Blair G.S., Coulson G., Andersen A., Blair L., Clark M., Cost H., Fitzpatrick T., Johnston L., Moreira R., Parlavantzas N., Saikoski K., "The Design and Implementation of OpenORB 2" To appear in IEEE Software Special Issue on Reflective Middleware, 2001.
- [Bousquet99] du Bousquet L., "Feature Interaction Detection using Testing and Model-Checking: Experience Report" In World Congress on Formal Methods, Toulouse, France, Springer, 1999.
- [Boussinot01] Boussinot F., Susin J.F., Dang Tran F., Hazard L., "A Reactive Framework for Dynamic Virtual Worlds" Proceedings of the Web3D 2001 Conference, Paderborn, Germany, February 2001.
- [Boussinot00] Boussinot F., Susin J.F., "Junior Rewrite Semantics" Inria Research Report, Available from <http://www-sop.inria.fr/meije/rp/junior/Semantics/rewrite-semantics.pdf>, October 2000.

- [Calder99] Calder M., Magil E., Marple D., "Hybrid Approach to Software and Interworking Problems Managing Interaction between Legacy and Evolving Telecommunication Software", IEE Proceedings Software, Vol 146 No 3 pp 167-175 June 1999.
- [Calder00] Calder M., Magil E. (eds), "Feature Interactions in Telecommunications and Software Systems VI" Glasgow, Scotland Amsterdam IOS Press 2000.
- [Cameron94] Cameron E.J., Griffeth N.D., Lin Y.J., Nilson M.E., Schnure W.K., Velthuiser H., "A Feature Interaction Benchmark for Now and Beyond" Proceedings of the 2nd International Workshop on Feature Interactions in Telecommunication Systems Bouma W., Velthuiser H. (eds), Amsterdam IOS Press pp 1-23, 1994.
- [Dini97] Din P., Boutab R., Logripp L. (eds), "Feature Interactions in Telecommunications Network IV" Montreal Canada Amsterdam IOS Press, 1997.
- [Efstratiou00] Efstratiou C., Chevers K., Davies N., Friday A., "Architecture Requirements for the Effective Support of Adaptive Mobile Applications" workshop paper in Middleware 2000, New York, April 2000.
- [Efstratiou01] Efstratiou C., Chevers K., Davies N., Friday A., "Architecture for the Support of Adaptive Context-Aware Applications" Proceedings of Mobile Data Management (MDM 2001) Hong Kong January 2001.
- [Fireworks97] "FIREworks Feature Integration in Requirements Engineering" Espri Working Group 23531 MRyan (Coordinator) started 1997 See <http://www.cs.bham.ac.uk/~mdr/fireworks/casestudies.html>.
- [Griffeth94] Griffeth N.D., Velthuiser H., "The negotiating agents approach to runtime feature interaction resolution" Proceedings of the 2nd International Workshop on Feature Interactions in Telecommunication Systems Bouma W., Velthuiser H. (eds), Amsterdam IOS Press, 1994.
- [Hall98] Hal R.J., "Feature Combination and Interaction Detection in a Foreground/Background Model" In [Kimble98] pp 232-246, 1998.
- [Hall00] Hal R.J., "Feature Interactions in Electronic Mail" In [Calder00] pp 67-82, 2000.
- [Heilmeier98] Heilmeier G.H. chairman emeritus of Bellcore (Morristown, N.J.) keynote address at the 35th Design Automation Conference reported in "New telecom service keeps vendor on the toes" Santarin M. http://eetimes.com/dac98/news_telecom.html.
- [Kimble95] Kimble K., Velthuiser H., "Feature Interaction Benchmarking" Discussion paper for the paneb Benchmarking FIW'95 (Feature Interaction Workshop), 1995.
- [Kimble98] Kimble K., Bouma L.G. (eds), "Feature Interactions in Telecommunications and Software Systems V" Lund, Sweden Amsterdam IOS Press, 1998.
- [Marple00] Marple D., "Detection and Resolution of Feature Interactions in Telecommunication Systems During Run-time" PhD thesis, available from Dept of Electrical and Electronic Engineering, University of Strathclyde, Glasgow, August 2000.
- [Medvidovic00] Medvidovic N., Taylor R.N., "A Classification and Comparison Framework for Software Architecture Description Languages" IEEE Transactions on Software Engineering, Vol 26 No 1 pp 70-93, January 2000.
- [Plath98] Plath M., Ryan M., "Plug-and-play features" In [Kimble98], pp 150-164, 1998.
- [Shaw96] Shaw M., Clement P., "A Field Guide to Boxology: Preliminary Classification of Architectural Styles for Software Systems" Computer Science Department and Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213, April 1996.
- [Szyperski98] Szyperski C., "Component Software Beyond Object-Oriented Programming", Addison-Wesley, 1998.