# 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: Proceedings of ECOOP 2001. Kluwer, B.V., Deventer, The Netherlands pp. 233-248.

### DOI

### Link to record in KAR

https://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 a Telecom Domain

Lynne Blair [1], Gordon Blair [1], Jianxiong Pang [2], Christos Efstratiou [2]

[1] *Department of Computer Science, Faculty of Science, University of Tromsø, N-9037 Tromsø, Norway.*
*Tel: +47 77645209, Fax: +47 77644580*
[2] *Computing Department, Lancaster University, Bailrigg, Lancaster, LA1 4YR, U.K.*
*Tel: +44 (0)1524 593802, Fax: +44 (0)1524 593608*
*email: {lb, gordon, efstrati}@comp.lancs.ac.uk, j.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 interactions in other domains, 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 for 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” [Kimbler95]. Several taxonomies have been produced in order to try and classify different types of interaction (including [Cameron94], [Kimbler95] and [Hall98]). A simple, yet we believe helpful, distinction from [Kimbler95] 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 examples, we refer the reader to the series of workshops in Feature Interactions in Telecommunications (and Software) Systems, e.g. [Dini97], [Kimbler98] and [Calder00]. However, recently it has become increasingly obvious that research into methods to detect and resolve such interactions in telecom systems is also of great significance outside the telecom domain. In fact, as recognised in [Calder00], “the subject has relevance to any domain where separate software entities control a shared resource”. 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 an earlier position paper [Blair00], we describe some interaction problems arising from Internet-based and multimedia/ mobile systems that led to the recently funded “FILBETT” project: Feature Interactions – Life Beyond Traditional Telephony (EPSRC GR/N35939/01). 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 Multimedia Research Group at Lancaster University, in addition to cataloguing a few other 'non-traditional' feature interactions from the literature.

In the remainder of this paper, we first provide brief details of the scope of the FILBETT project below (section 2) and then document some of the 'non-traditional' interactions we have come across (section 3). Finally, we discuss some of the major techniques that exist for the detection and resolution of feature interactions (section 4) and then draw our conclusions (section 5).

## 2. FILBETT – Life Beyond Traditional Telephony

### 2.1. Overview

Motivation for this project came from a number of examples of interactions that we encountered when looking at Internet-based and multimedia/mobile 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/mobile systems and, as the number of new services grows, the potential for interactions between services will inevitably explode.

To help to address this, the main goal of the FILBETT project is 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 our 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, but for details the reader is referred to [Blair00].

- combining a traditional telecommunications service with Internet access
- potential interactions with one-to-many services
- TCP flow-control mechanisms, and protocol interactions in general
- sharing demand for network bandwidth: web browsing and viewing a video 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 work at Lancaster on mobile computing (see [Efstratiou00] and [Efstratiou01]). In 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, problems (interactions) may arise if separate adaptation mechanisms are employed for different attributes. Examples include mechanisms to adapt/manage power consumption, 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 will request applications that are using network bandwidth to postpone this use, so as to place the network device in sleep mode. However, as a consequence, the network adaptation mechanism will now detect unused bandwidth and will notify applications that they can use this spare bandwidth, in direct conflict with the power management adaptation mechanism!

Note that a further interesting dimension to adaptation is user-configuration of devices whereby a user can express preferences over different 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 FILBETT is primarily concerned with the generalization of feature interactions to a new world of mobile, multimedia and IP-based services, it is clear that the value of research into such interactions does not stop here. Two further areas from our work that would benefit from the application of feature interaction research are described below. As an aside, some additional examples of 'non-traditional' interactions can be found in [Hall00] and [Fireworks97], relating to email systems and a variety of miscellaneous examples (including a lift system, a tape-deck system, a metro ticketing 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) [Szyperski98]. By allowing new components to be added, and existing software to be packaged as components, we obtain an incremental development model for evolutionary and dynamic architectures. However, this raises two key questions:

- When we *compose* an architecture, how can we be confident that components work well together, that there are no unwanted or subtle interactions and that the result is coherent?
- When we *adapt* an architecture, how can we be confident that replacements 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 interactions between components. This is an area that would benefit greatly from the application of techniques from feature interaction research. In particular, a *hybrid approach* (see below) would allow *design-time* checks to 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 this middleware structure via a *meta-level*. This approach enables middleware to be customized for a particular application domain, 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 if now 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).

### 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 publicly exposed attributes that can be altered in interaction with other objects; an active object on the other hand also includes behaviour (which in Ping is expressed as a series of reactive scripts written in the Junior scripting language [Boussinot01]). We are investigating an *aspect-oriented* approach to composing such behaviours using Junior, with consideration of many aspects including the capturing of (virtual) world physics (gravity, inertia, etc), distributed systems policies such as replication and consistency management, reaction to collisions, and also 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 event dissemination if operating over a modem. We are investigating an approach whereby monitoring 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, the problems are similar to those considered above. In particular, we are concerned about the initial configuration and the subsequent re-configuration of a platform. In this case however, we are concerned about interaction between behaviours at both an inter- and intra-object level. Importantly, in this work we already have a significant advantage in that Junior has a formal operational semantics (expressed using rewriting rules) [Boussinot00], thus aiding formal analysis.

## 4. A Brief Summary of Feature Interaction Detection and Resolution Techniques

Existing analysis techniques can be seen to fall into 3 broad categories: *off-line* (or design-time) techniques, *on-line* (or run-time) techniques and *hybrid* techniques [Calder99].

With *off-line* techniques, a model of the base system and the additional services or features are specified in a formal language whilst the properties that the system should exhibit are (typically) specified through the use 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 grow, 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 the use of abstractions or symmetries. Such techniques can help to greatly reduce the state-space explosion problem. A further problem however is that the level of success is dependent on the accuracy and level of abstraction of the specified properties. An inaccurate (or rather, not precise enough) property specification will inevitably 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 become increasingly (dynamically) adaptive. Adaptation strategies have typically been poweredbyaknowledgedatabase,suchaspredefined tables,statetransitionrules,abstract data types and user agent rules. For example, in [Griffeth94] unknown new features are accommodatedthroughanadaptive“agentregime”architecturewhereuseragentsengagein negotiationtosettlethediscernedconflictsbetweenfeatures.

Finally,inrecognitionoftheadvantages(andcertaindrawbacks)ofbothoff-lineandon-line techniques, a *hybrid* approach is proposed in [Calder99]. This approach is targeted at resolving interactions between new services and legacy services and combines an on-line, transactionalapproachwithoff-lineformalanalysis(seealso[Marples00]).

## 5.Conclusions

Itshouldbeapparentfromthediscussionsabovethatmanyareasofcomputersciencecan benefitfromresultsinthefieldoffeatureinteraction.Wehaveidentifiedanumberofareas wherewebelievethistobethecaseincludingmobileandmultimediasystems,component-basedandreflectivemiddleware,andalsobehaviouralspecificationinvirtualenvironments.

Aninterestingfirstlineofresearchistoconsidertheimpactofcomponentsonthefeature interactionproblem,includingforexamplethepotentialroleofexplicitcontextdependencies (required/providedinterfaces)[Szyperski98]insimplifyinganalysisofpotentialinteractions. Moregenerally,furtherresearchisclearlyrequired,includingstrongcollaborationbetween the different research communities, in order to more fully understand the relationships betweenfeatureinteractionsandtheseotherareas.

## Acknowledgements

Theauthorswouldliketothankanumberofresearchers at Lancasterwhohavecontributed todiscussionsontheinteractionsdocumentedabove.Inparticular,wewouldliketothank Adrian Friday and Keith Cheverst for their insights into mobile computing, the various members of the Open ORB project for discussions on component-based and reflective middleware (http://www.comp.lancs.ac.uk/computing/research/mpg/reflection/memb.html), andPaulOkandaforhisthoughtsonbehaviouralinteractionsinvirtualenvironments.

## References

[Blair98] BlairG.S.,CoulsonG.,RobinP.,PapathomasM.,“AnArchitecture forNext GenerationMiddleware”,ProceedingsoftheIFIPInternationalConferenceonDistributed SystemsPlatformsandOpenDistributedProcessing(Middleware’98),Springer,1998.

[Blair00] BlairL.&PangJ.,“FeatureInteractions-LifeBeyondTraditionalTelephony”,In [Calder00],pp83-93,2000.

[Blair01] BlairG.S.,CoulsonG.,AndersenA.,BlairL.,ClarkeM.,CostaF.,Duran-Limon H.,FitzpatrickT.,JohnstonL.,MoreiraR.,ParlavantzasN.,SaikoskiK.,“TheDesignand ImplementationofOpenORBv2”,ToappearinIEEEDSOnline,SpecialIssueonReflective Middleware,2001.

[Bousquet99] duBousquetL.,“FeatureInteractionDetectionusingTesting andModel-Checking: ExperienceReport”,InWorldCongressonFormalMethods,Toulouse,France,Springer,1999.

[Boussinot01] BoussinotF.,SusiniJ.F.,DangTranF.,HazardL.,“AReactiveBehavior FrameworkforDynamicVirtualWorlds”,ProceedingsoftheWeb3D2001 Conference, Paderborn,Germany,February2001.

[Boussinot00] BoussinotF.,SusiniJ.F.,“JuniorRewriteSemantics”Inria ResearchReport, Availablefromhttp://www-sop.inria.fr/meije/rp/junior/Semantics/rewrite-semantics.pdf, October2000.

[Calder99] CalderM.,MagillE.,MarplesD.,“HybridApproachtoSoftw areInterworking Problems:ManagingInteractionsbetweenLegacyandEvolvingTele communicationsSoftware”, IEEProceedings-Software,Vol.146,No.3,pp167-175,June1999.

[Calder00] CalderM.,MagillE.(eds),“FeatureInteractionsinT elecommunicationsand SoftwareSystemsVI”,Glasgow,Scotland,Amsterdam:IOSPres s,2000.

[Cameron94] CameronE.J.,GriffethN.D.,LinY.J.,NilsonM.E.,SchnureW.K.,V elthuijsenH., “AFeatureInteractionBenchmarkforINandBeyond”,Proceedingsof the2nd International WorkshoponFeatureInteractionsinTelecommunicationsSystems,Bouma W.,VelthuijsenH. (eds),Amsterdam:IOSPress,pp1-23,1994.

[Dini97] DiniP.,BoutabaR.,LogrippoL.(eds),“FeatureInteractionsin TelecommunicationsNetworksIV”,Montreal,Canada,Amsterdam:IOS Press,1997.

[Efstratiou00] EfstratiouC.,CheverstK.,DaviesN.,FridayA.,“A rchitecturalRequirementsfor theEffectiveSupportofAdaptiveMobileApplications”,workinprogre sspaperinMiddleware 2000,NewYork,April2000.

[Efstratiou01] EfstratiouC.,CheverstK.,DaviesN.,FridayA.,“A nArchitecturefortheSupport ofAdaptiveContext-AwareApplications”,ProceedingsofMobileData Management(MDM 2001),HongKong,January2001.

[Fireworks97] “FIREworks:FeatureIntegrationinRequirementsE ngineering”,EspritWorking Group23531,M.Ryan(Coordinator),started1997.See http://www.cs.bham.ac.uk/~mdr/fireworks/casestudies.html.

[Griffeth94] GriffethN.D.,VelthuijsenH.,“Thenegotiatingagents approachtoruntimefeature interactionresolution”,Proceedingsofthe2nd InternationalWorkshoponFeatureInteractionsin TelecommunicationsSystems,BoumaW.,VelthuijsenH.(eds),Amsterda m:IOSPress,1994.

[Hall98] HallR.J.,“FeatureCombinationandInteractionDetectionvi aForeground/ BackgroundModels”,In[Kimbler98],pp232-246,1998.

[Hall00] HallR.J.,“FeatureInteractionsinElectronicMail”,I n[Calder00],pp67-82,2000.

[Heilmeier98] HeilmeierG.H.,chairmanemeritusofBellcore( Morristown,N.J.),keynoteaddress atthe35th DesignAutomationConference,quotereportedin“Newtelecomservic eskeep vendorsontheirtoes”,SantariniM.,http://eetimes.com/dac98/news_tele com.html.

[Kimbler95] KimblerK.,VelthuijsenH.,“FeatureInteractionBenchma rk”,Discussionpaperfor thepanelonBenchmarkingatFIW’95(FeatureInteractionWorkshop),1995.

[Kimbler98] KimblerK.,BoumaL.G.(eds),“FeatureInteractionsinT elecommunicationsand SoftwareSystemsV”,Lund,Sweden,Amsterdam:IOSPress,1998.

[Marples00] MarplesD.,“DetectionandResolutionofFeatureInterac tionsin TelecommunicationsSystemsDuringRun-time”,PhDthesis,Availabl efromDept.ofElectrical andElectronicEngineering,UniversityofStrathclyde,Glasgow,Aug ust2000.

[Medvidovic00] MedvidovicN.,TaylorR.N.,“AClassificationandCompar isonFrameworkfor SoftwareArchitectureDescriptionLanguages”,IEEETransactions onSoftwareEngineering, Vol.26,No.1,pp.70-93,January2000.

[Plath98] PlathM.,RyanM.,“Plug-and-playfeatures”,In[Kimbler98], pp150-164,1998.

[Shaw96] ShawM.,ClementsP.,“AFieldGuidetoBoxology:Preliminar yClassificationof ArchitecturalStylesforSoftwareSystems,ComputerScience DepartmentandSoftware EngineeringInstitute,CarnegieMellonUniversity,Pittsburgh,PA 15213,April1996.

[Szyperski98] SzyperskiC.,“ComponentSoftware:BeyondObject-Orie ntedProgramming”, Addison-Wesley,1998.