Author(s): Gillan, Charles; Kilpatrick, Peter; Spence, Ivor; Brown, T. John; Bashroush, Rabih; Gawley, Rachel;
Article title: Challenges in the Application of Feature Modelling in Fixed Line Telecommunications
Year of publication: 2007
Citation: Gillan, C. et al. (2007) 'Challenges in the Application of Feature Modelling in Fixed Line Telecommunications.' Proceedings of the First International Workshop on Variability Modelling of Software-intensive Systems (VaMoS2007), Lemrick, Ireland, Jan 16 -18, 2007.
Link to published version:
http://www.vamos-workshop.net/2007/files/VAMOS07_0007_Paper_16.pdf

Challenges in the Application of Feature Modelling in Fixed Line Telecommunications

C.J. Gillan, P. Kilpatrick, I. Spence, T.J. Brown, R. Bashroush and R. Gawley
The Institute for Electronics Communications and Information Technology (ECIT)
The Queen's University of Belfast (QUB), The Northern Ireland Science Park
Queen's Road, Queen's Island, Belfast, Northern Ireland BT3 9DT, UK
c.gillan,r.bashroush@ecit.qub.ac.uk
i.spence,tj.brown,p.kilpatrick,r.gawley@qub.ac.uk

Abstract

The global telephone system is a complex transmission network, the features of which are defined to a very high level by ITU-T standards. It is therefore a prime candidate at which to target the application of software product line techniques, and feature modelling in particular, in order to handle the inherent commonality of protocols and variability in equipment functionality. This paper reports on an experimental feature modelling notation and illustrates it with application to the modelling of embedded software for the core network elements. We look at three of the fundamental challenges facing the adoption of feature modelling in the field and explain how we have strived to address these within our tools set.

1. Introduction

The global telephone system is a technology space which is almost entirely defined by standards, mostly internationally agreed, and therefore should be a fruitful proving ground for the application of studies in variability management techniques. There are however relatively few such studies[1] in the literature.

As a market, the telecommunications sector is currently influenced by a number of factors for change meaning that there will continue to be an evolution of the inherent feature set. The global system continues to evolve in order to accommodate new service requirements, one example being the modifications such as real and virtual concatenation[2] introduced over the last decade to accommodate transport of data rates such as Gigabit Ethernet (GbE) and also Storage Area Networking technologies (SAN) such as Fibre Channel (FC) in addition to the traditional multiplexed hierarchies of 64 Kbit/s voice channels.

The next major evolution of the system is the incorporation of low cost Passive Optical Networks (PONs). A PON replaces the *final mile* copper network with an Optical Line Terminal (OLT) located at the local telephone exchange connected to an Optical Network Unit in the home (ONU) and for this reason is sometimes known as the Fibre To The Home (FTTH) solution. The roll-out of PON

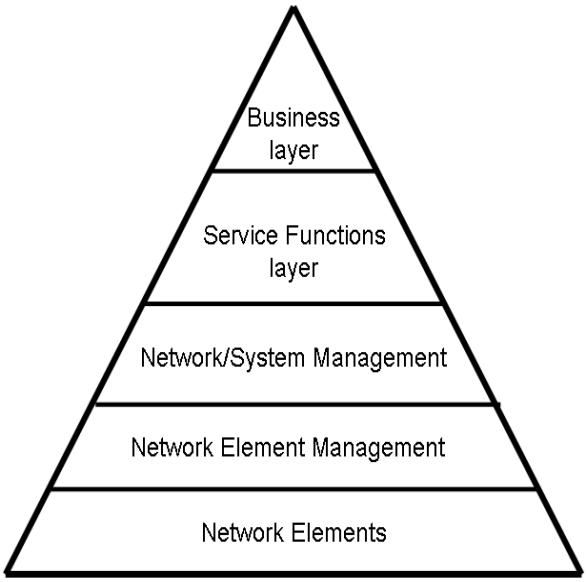

Figure 1. Representation of the TMN pyramid defined in ITU-T standard M.3010. Each higher layer functions as a client of each immediate lower layer which therefore takes on the role of a server. The FCAPS functional areas are perpendicular to each layer and therefore cross-cut each.

technology has so far mostly taken place in the Asia-Pacific region but is expected to generate a global market opportunity approaching $2.2 billion by the end of the current decade. It is worth noting that the financial case for PON role-out is based on the continuing update of novel broadband services as the revenue generating potential of traditional voice telephony approaches saturation, at least in the developed world. The addition of new services obviously introduces yet more variability into an already complex system. In turn, the System on Chip (SoC) revolution, which we briefly describe later in this paper, has lowered the cost of equipment manufacture, even for large multiplexers, thereby opening the vendor space to Small and Medium Enterprises (SMEs) whereas previously it was essentially the sole preserve of a small number of large multi-national companies with sufficient resources to fund ASIC developments. Of course, this also generates intense competition which translates to severe Time To Market (TTM) pressures on any company working in the space. At present there are estimated to be more than twenty five companies actively developing equipment in the PON sub-space.

Clearly, the telecommunications sector is critically dependent on software both at the embedded level, for device control and monitoring of the Network Elements (NEs), and in terms of the management, service and business level applications which are the tools with which a telecommunications operator leverages option value from their installed base of NEs. Although extreme programming is now being considered as viable alternative in some areas of embedded systems programming[3], legislative restrictions and the requirement for five nines reliability (99.999 percent) within the telecommunications industry is such that it will probably remain tightly bound to the traditional software development life cycle for some time to come. Telecommunications software engineering is by its nature a very conservative environment although somewhat less so than the aeronautical and nuclear industries. Related to this is the fact that ISO9001 certification is essentially mandatory for any telecommunications equipment manufacturer and this seems to generate a level of conservatism favouring the traditional waterfall life cycle of software development. Still, appropriate software engineering techniques applied to this domain can help ameliorate the increasing TTM pressures faced by all vendors in the telecommunications space and we believe that feature modelling is one such technique.

There are, as we see it, three main challenges facing the adoption of feature modelling in telecommunications development.

1. the number of variability points is excessively large, reflecting the network itself. In addition, as new technology, such as PON, arrives the feature set will grow rapidly. So, clearly we need some mechanism of standardizing our approach to domain modelling.
2. The second problem, driven by advances in hardware engineering, and is that SoC concepts drive more and more functionality out of the software into hardware within any one product but sometimes only well beyond the first release. This can be due to either added new functionality or sometimes to cost reduction activities providing an opportunity to migrate functionality on a board away from the processor onto an FPGA device or similar.
3. The third challenge facing the adoption of feature modelling in telecommunications software development is the fact that elucidation of a set of binary yes-no feature points is woefully inadequate to reflect the functionality of a device. One needs to encompass behaviour within the feature modelling in order to capture fully the behaviour of a software load.

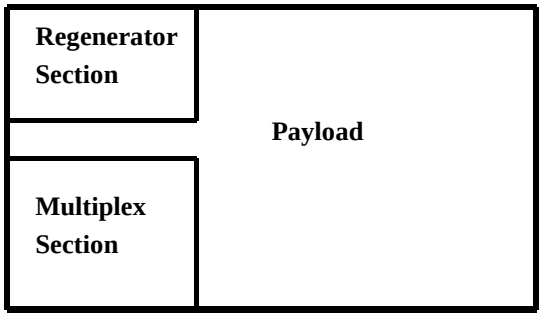

Figure 2. The generic physical view of the SDH frame structure. This needs to be contrasted with the logical layered view shown in figure 3.

In this paper, we report on the strategies that we have followed to begin to address each of these challenges and we illustrate these with examples from our research tools set. However, we stress at the outset that much work remains to be done to achieve a successful application of feature modelling to telecommunications and we comment at the end of our paper on what we believe to be the way forward. Telecommunications is in itself a diverse field presenting substantial challenges for variability management some of which are unique to the field itself. We have developed our strategies within the telecommunications field and have not as yet attempted to apply them in a wider context.

1.1. Focusing on the Core

Within this paper it is neither feasible nor appropriate to try to consider the entire telecommunications domain so we pick one subset of it for illustration. Specifically, we choose

the fibre optic backbone, at the heart of the global telecommunications system. Differences with the US and Canada at the protocol level notwithstanding, this implements the Synchronous Digital Hierarchy (SDH) protocol[4].

Our group has considerable experience in this sub-space through participation over several years in a program of industrial research, known as JigSAW, funded by Nortel Networks. So, for the rest of this paper we shall be applying feature modeling to SDH concepts. However, we stress that the points that we make are generally applicable in other telecommunications technology spaces.

2. Divide and Conquer: The ITU-T Standards and TMN

Perhaps the key enabler which facilitates global telephone systems, and that includes all technologies not just SDH, is that it is subject to strict standardization under the direction of the United Nations by the sub-division known as The International Telecommunications Union Standardization Sector (ITU-T). ITU-T is responsible for "studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis." In order to react to the changing telecommunications environment.

At the outset of the digital communication era, in the 1980s, the ITU defined the Telecommunications Management Network (TMN) model to specify a set of standard functions and interfaces for management of a multi-technology network thereby reflecting the functionality offered by a system. The TMN framework is typically viewed as a pyramid composed of layers enjoying a client-server relationship to each other as shown in figure 1. Perpendicular to these layers are a set of vertical functional areas:

- Fault
- Configuration
- Accounting
- Performance
- Security

known as the FCAPS model for short. This TMN based strategy, we therefore propose, is the one that should be followed in order to divide and conquer the problem of feature scalability and evolution in telecommunications. We have approached feature modelling therefore by classifying features in line with the FCAPS model. This also, we have found, lowers the barrier to adoption by telecommunications software engineers.

An alternative perspective is to take a hardware engineer's perspective and to view the features in terms of

1. the overhead bytes within the protocol definition (see figure 2)
2. the traffic rate layer within the SDH frame at which the overhead is found (see figure 3).

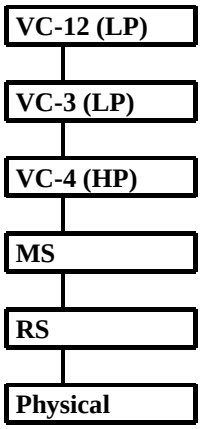

Figure 3. Some of the logical layers, from the standard ITU-T G.707, within an SDH signal. An STM-1 frame contains 1xRS, 1xMS, 1xVC-4 containers. The VC-4 container has 3xVC-3s and each VC-3 has 21xVC-12. VC-4 is known as a high path layer (HP), VC-3 and VC-12 as low path layers (LP)

In our experience, the latter approach alone generates a feature model with many more cross linkages, being therefore much more complex to maintain and manipulate, than starting with an FCAPS based model. Note, however, that within the FCAPS model, as shown in figures 4 and 5, generating using our tools set which is discussed later in the paper, there is still a need to take account of the traffic rates.

However, we have to add that this FCAPS distinction alone is not sufficient to guarantee a unique feature model, and this in itself is a problem. For example figures 4 and 5 are essentially representing the same thing, that is the collection of performance monitoring data on the overhead bytes of a telecommunications signal.

In figure 4,the sub-tree under *Performance Monitoring* has been developed to incorporate the logical layers of the signal which are shown in figure 3. This is why there is a dense mesh of features on the lower left-hand side of the screen. For example the overhead bytes B3 and G1 which are present in the High Path (HP) signals are shown as a sub-tree under the HP feature. The B3 byte is further expanded to encompass all of the properties pertaining to it such as *24 hour* and *15 minute* history bin collection.

In the case of figure 5, the explicit specification of the logical layers hierarchy has been omitted and it is only im-

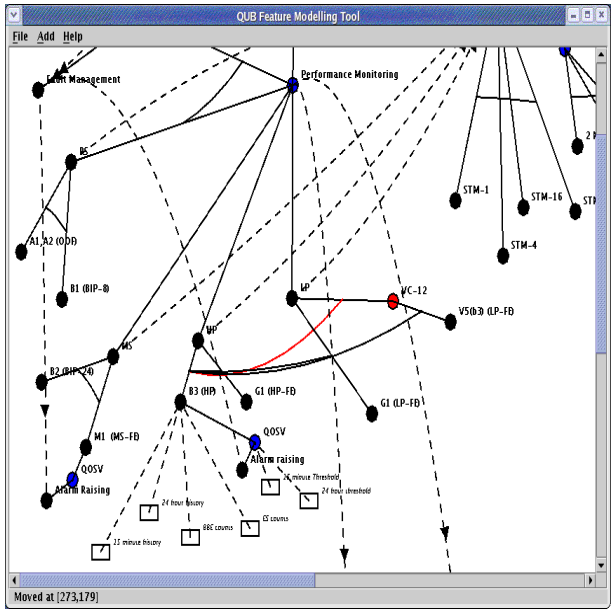

Figure 4. A small number of the variability points related to SDH performance monitoring and their coupling to the multiplexing hierarchy (dashed lines). The named bytes are described in ITU-T G.707. The amount of clutter is clear.

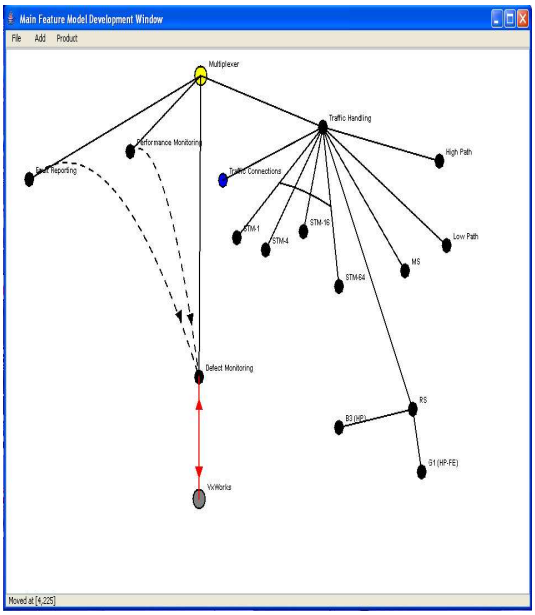

Figure 5. An alternative, but equally valid way to present the same set of features as in figure 4. This illustrates the non-unique nature of the feature modelling technique.

plicit contained within the *Traffic Handling* sub-tree on the right-hand side of the figure, a tree which is apparently decoupled from the *Defect Monitoring* feature. In this representation the *Defect Monitoring* feature incorporates all of the process issues surrounding collection of bit rate errors but does not distinguish the details for individual bytes. In general, we would probably also need to define a dependency between the Traffic Handling and Defect Monitoring feature. Indeed, this approach would be extensible to other functional areas of the system, such as the treatment of the features associated with the configuration of data for each port. This is because each port is ultimately involved in a connection at some particular data rate. However it would be just as valid to define a port in terms of logical layers.

3. Co-design feature models: handling the impact of FPGAs

System-on-a-chip (SoC) design, in its most general form, is about integrating all the components of an electronic system into a single chip, for example containing digital, analog, mixed-signal, and often radio-frequency functions all on one chip. SoC is therefore a key driver in decreasing the size of electronic boards. An SoC consists of this hardware but also of the software that controls the microprocessor and/or DSP cores, peripherals and interfaces. The design flow for an SoC aims to develop this hardware and software in parallel. For example, most SoCs are developed from pre-qualified blocks for the hardware elements (called IP cores), together with the software drivers that control their operation. The hardware blocks are put together using CAD tools; the software modules are integrated using a software development environment. The Field Programmable Gate Array (FPGA) is one technology that has facilitated the SoC revolution because, as its name implies, it is hardware which can be reprogrammed hardware. Often in the early life cycle of a product and FPGA will be used and later the hardware design will be migrated into an ASIC which remains the more cost effective solution for high volume production.

Clearly, the SoC concept can have a practical impact on any telecommunications board in the sense that one might migrate some of the functionality into a combined electronic device over the life cycle of manufacture of the product. Take as an example performance monitoring collection. In the simplest case one has a traffic carrying ASIC that records bit rate defects per channel over the course of one second. So, one must then schedule a one second interrupt on the programmable interval timer (PIT) of the board's microprocessor and within that piece of interrupt code collect the bit rate counts. Note the time constraint is important here. These are then passed, for example via a VxWorks message queue, to a lower priority process which stores the

counts into relevant data bins and performs various summations on the counts. Every fifteen minutes, as mandated by standards, we must roll over to a new bin but retain several of the *previous* bins, typically up to 96 of them. On top of this recording mechanism, at any instant, a management interface request to be given visibility of any bin, active or historical, may need to be handled. In essence, this is the set of features being recorded in figures 4 and 5; it is a significant part of the software implementation on any SDH equipment and scales in portion to the number of traffic channels carried. However, today it is possible to buy COTS ASICs that perform essentially all of this activity and thus render the entire software module for performance collection obsolete in any product variant which adopts such an ASIC. All that is required is to interface the management application to the registers holding history data.

So, we need to able, in our feature models to handle the situation whereby we have this migration of a feature into hardware. We have previously introduced the concept of bi-directional feature modelling[5] to handle this situation. We illustrate its role in figure 6 in which we present a drastically simplified feature model for the situation. This figure

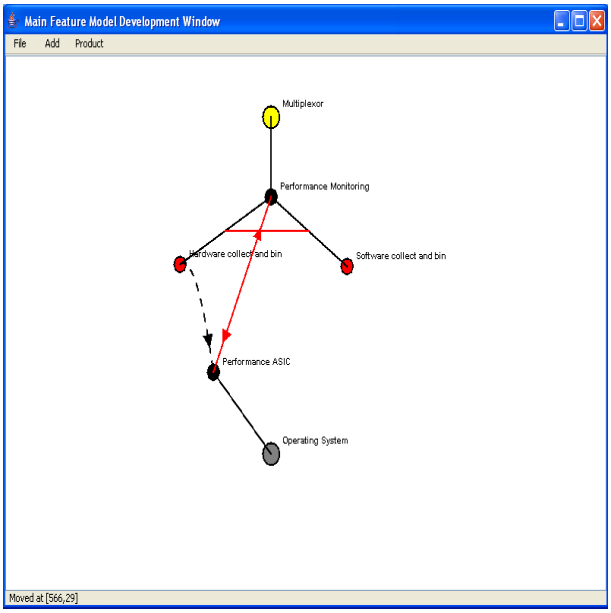

Figure 6. This figure illustrates the role of the platform node in our bi-directional feature modelling notation. In the potential migration of the PM collect-and-bin function from software to hardware is shown.

has been created by condensing the features in figure 5 into subsets and then showing the relationship between the subsets. Being much less cluttered, figure 6 is also able to show the link between functional features and the ASIC and Operating System.

4. Adding Behaviour

Feature modelling, at least in its original form [6] is simply a set of binary points for which any given product adds the values yes/no. This alone is not sufficient to classify the behaviour of a telecommunications product as we show in figures 7 and 8 and discuss below.

Consider first the case of a feature point with two mandatory child features and one alternative child feature, such as the one shown in figure 7. The collection, storage and reporting of performance monitoring data is one example of this. The two mandatory child features are

1. collection
2. sorting into bins.

The alternative child feature relates to the way in which the data is reported. In older systems, where there are relatively few monitoring points and perhaps few history bins, it is viable to report performance monitoring data at the command line interface (CLI) without otherwise impacting the NE itself or indeed the management network overhead channels (the 192 Kbit/s D1-D3 DCC bytes defined in G.707) that

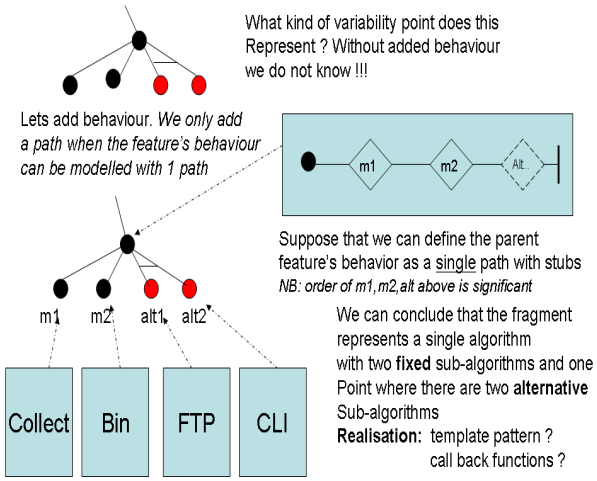

Figure 7. Example of a feature for which the three child features can be assigned to a specific behaviour pattern. The graphics are extracted from our feature GUI and we added some additional text, in this figure, to help explain these.

link the management monitoring workstation to the NE being investigated. However, in larger NE systems this puts a heavy load on both the processor and the DCC channels, so it is preferable to perform bulk FTP transfers of compressed files at set intervals. They key point here is that the behaviour of the parent feature can be classified as execution of each child in succession from left to right; that is the behaviour pattern is always the same. We have augmented out feature modelling GUI to include the Use Case Maps (UCM) notation to capture behaviour. Actually this is line with totally independent recommendations of the ITU where UCM has been adopted and renamed URN. The ITU-T Z Series, *Languages and General Software Aspects for Telecommunications Systems*, has two standards which are of particular relevance:

Z.150 User Requirements Notation (URN) - language requirements and framework

Z.152 ITU-T Study Group 17, Progress work on URN - Use Case Map Notation (Z.152). This is not yet formally adopted as a standard but it is expected that it soon will be adopted.

Of course not all features will have a unique behaviour assigned to them. Consider for example the feature shown in figure 8. At first sight this looks the same as the feature in figure 7 with two mandatory child features and one alternative child feature. However, in this case the feature represents the presence of a variety of management interfaces.

All products require a serial port interface, which is typically based on an Intel NS16550 chip, so that they can be commissioned in the field. This interface can be connected to the serial port on laptop PC using an appropriate serial port cable. As serial ports are readily supported, for example by the Microsoft Windows HyperTerminal program, this interface provides the simplest type of physical connection to the NE obviating the need for the presence of a network of any kind. Of course, there needs to be a layer of software to handle commands and to provide responses to those commands at the interface. In summary, the serial port interface is the primary route by which an NE is commissioned onto the network and brought into service making it a mandatory feature. Once in this state, network based management interfaces are clearly more suitable and at least one such networked interface will always be provided thereby enabling the capacity to remotely manage the NE from a Network Operations Centre (NOC) which could, in principle, be anywhere in the world.

Almost all products now incorporate an SNMP interface as that is a very popular management interface in the enterprise field. SNMP standards for Simple Network Management Protocol (SNMP) which is a part of the Internet Protocol (IP) suite as defined by the Internet Engineering Task Force (IETF). With reference to the ISO seven layer model, SNMP is considered as an application layer protocol and in turn it uses UDP, at layer four, as a server layer. Therefore, SNMP is used by network management systems to monitor network-attached devices particularly for alarm conditions requiring administrative attention.

There will additionally be at least one kind of Command Line Interface (CLI) for the system. CLIs are generally preferred by telecommunications operators as opposed to the use of SNMP by enterprise operators or even for Customer Premises Equipment (CPE). A key advantage of any CLI

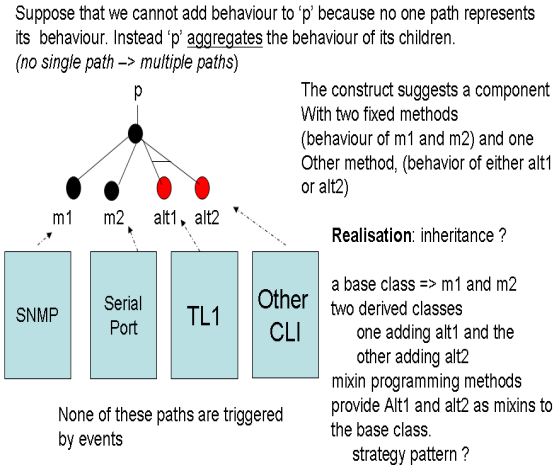

Figure 8. For this parent the child features do not have an associated unique behaviour pattern and so represent an entirely different type of feature than the one in figure 7.

is that one can write scripts, for example in Perl, to automate any amalgam of required command line operations. By contrast, when using SNMP, a layer of client software is required.

In the US and Canada provision of the CLI known as TL1 is essentially mandatory for transmission equipment because of the widespread use of Telcordia network management software, whereas in the rest of the world a different kind of CLI can be just as acceptable. TL1, which is in fact based on the ITU-T Z.300 series Man Machine Language (MML) standards, was developed by Bellcore (which has subsequently been renamed Telcordia Technologies) around 1984 for the Regional Bell Operating Companies (RBOCs) in North America as a language with which to manage NEs. TL1 is full standardized in publication GR-831 by Bellcore. Telcordia Operation Support Systems

(OSS) such as Network Monitoring and Analysis (NMA) uses TL1 as the Element Management System (EMS) protocol for example and therefore drove NE vendors to implement TL1 in their devices. Generally though in the case of embedded systems, each vendor will also define their own CLI and a proprietary set of rules to which all commands within their CLI conform. One such example is Cisco IOS. This diversity of CLIs is represented by the features *alt1* and *alt2* in figure 8.

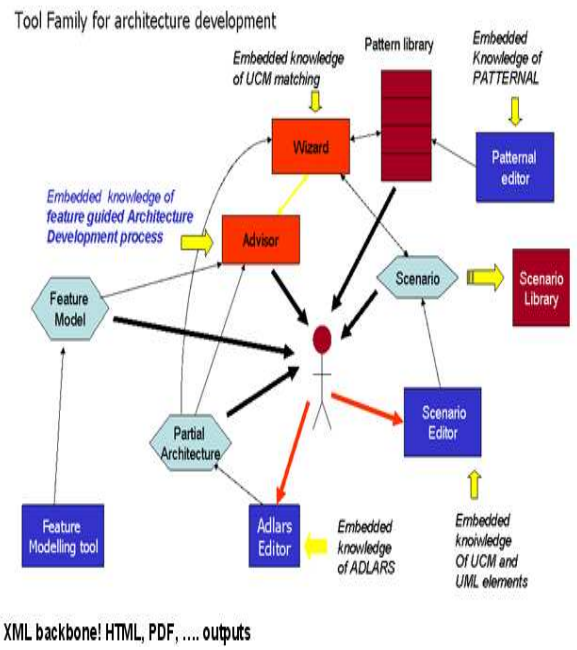

Figure 9. Overview of the QUB/ECIT tools set for feature guided architecture development which is explained further in the text.

We cannot assign a unique behaviour path to the parent feature here. So, clearly, the parent feature in figure 8 is an entirely different one than that depicted in figure 7. So, it is only by adding a notation for behaviour to the features that permits us to make a distinction between them. We have found UCM to be well suited to this. We have also found, by distinguishing features in this way, that we can gain information on possible implementation strategies. Figures 7 and 8 include comments on the possible relevant object oriented patterns for implementation in each case.

5. Beyond Feature Modelling

To conclude the paper we point out that we do not see our work on feature modelling in isolation from other efforts that must be undertaken to deliver a working software system[7, 8, 9]. Our feature modelling tool, which has been illustrated earlier in this paper, is part of a wider research effort to generate feature architecture development tools. This is reflected in the grand view of our QUB/ECIT tools set in figure 9.

If we consider the actor in the centre of figure 9 as the end goal of creating a running software system, we can identify several steps to achieving this by working inwards from the periphery of the figure. On the left-hand side we have the feature modelling tool which we have illustrated in this paper. The output from this tool feeds directly into the architecture editor (ADLARS) tool within which we can add event driven behaviour, this being a well known technique for construction of embedded systems. The ADLARS tool can create code although currently only Java. The PATTERNAL tool, which is still being actively researched is used to represent object-oriented patterns in visual notation.

5.1. Addressing the documentation requirements

It is our observation that in the telecommunications software industry today the process of preparation of ISO9000 compliant documentation is largely divorced from the creation of the core asset code base thereby leading to substantial duplication of effort. Yet there are now many techniques, driven largely by Web technologies, that would facilitate substantial auto-generation of documentation. In addition, one then removes the problem of keeping document and code in synchronization, a working practice that is particularly difficult to maintain under TTM pressures.

We have found that it is actually quite straightforward to incorporate the aforementioned Web technologies into our tools set therefore overcoming, we suggest, the aforementioned disconnection between documentation and implementation. To this end, we use an XML representation *behind the scenes* of our GUIs and combine this with XSL transforms to generate documentation from our feature models in such formats as PDF, Word, Excel, HTML, etc. The Word for PDF formats remain, in our experience, by far the most popular for ISO 9000 review process in most companies.

We should point out that this idea for documenting software is in itself not particularly new but is an evolution of Knuth's literate programming idea. In the eighties, Donald Knuth created a language known as *The Web language*, and later CWeb, to allow the preparation of a single document containing all the information that is needed both to produce a compilable Pascal program, and later C, and to produce a well-formatted document describing the program in as much detail as the writer may desire[10, 11].

6. Summary and Conclusions

In this work, we have addressed some of the main challenges facing the adoption of feature modelling within the telecommunications software industry and have focused on the role of performance monitoring within SDH network elements to illustrate out ideas.

We propose firstly that, in order to overcome the sheer size of the variability management problem, the analysis of features should be guided by existing domain specific partitioning ideas, specifically the FCAPS model. This will lower the barriers to adoption of the techniques in what is quite a conservative industry. Next, we have explained how the role of SoC forces us to integrate hardware co-design features at least into the software engineering feature model. We have then argued the case for the need to add behaviour to a feature model in order to clarify its meaning. Finally, we have explained how we are incorporating behaviour guided feature modelling into out tools set for architecture development.

As we have shown, there is no unique way to express the feature model for a telecommunications system and we believe that overcoming this hurdle would expedite the acceptance of the technology in the field. Clearly, there is a requirement for more and deeper case studies in the area.

7. Acknowledgements

The Institute for Electronics Communications and Information Technology (ECIT) is jointly funded by The Queen's University of Belfast, Invest Northern Ireland and The Department of Employment and Learning for Northern Ireland.

References

[1] K Lee, K C Kang, S Kim and J Lee, *Feature-Oriented Engineering of PBX Software* Proceedings of the Sixth Asia-Pacific Software Engineering Conference (APSEC'99) page 394, pub: IEEE Computer Society.

[2] See ITU-T standards: G.707 Virtual Concatenation (VCAT), G.7041 Generic Framing Procedure (GFP) and G.7042 Link Capacity Adjustment Scheme (LCAS).

[3] P Manhart and K Schneider, *Breaking the Ice for Agile Development of Embedded Software: An Industry Experience Report*, Proceedings of the 26$^{\text{th}}$ International Conference on Software Engineering (ICSE'04), pages 378-386, pub: IEEE Computer Society

[4] See the ITU-T section at the web address http://www.itu.int.

[5] T J Brown, R Gawley, R Bashroush, I Spence, P Kilpatrick and C J Gillan, *Weaving Behavior into Feature Models for Embedded System Families*, Proceedings 10th International Software Product Line Conference, Baltimore Maryland, IEEE Computer Society, pp52-64, August 2006, ISBN 0-7695-2599-7.

[6] K C Kang, S G Cohen, J A Hess, W E Novak and A S Patterson, *Feature Oriented Domain Analysis (FODA) feasibility study.* Technical Report CMU/SEI-90-TR-21, available from the Software Engineering Institute, Carnegie-Mellon University 1990.

[7] T J Brown, I Spence and P Kilpatrick, *A Relational Architecture Description Language for Software Families* in Proceedings of the 5th International Workshop on Product Family Engineering, Sienna, Italy 2003.

[8] T J Brown, R Bashroush, I Spence and P Kilpatrick, *Feature-guided architecture development for embedded system families*, in Proceedings IEEE/IFIP Working International Conference on Software Engineering (WICSA '05), pp. 223-226, Pittsburgh, 2005.

[9] R Bashroush, T J Brown, I Spence and P Kilpatrick, *ADLARS: An Architecture Description Language for Software Product Lines*, Proceedings of the 29th IEEE/NASA workshop on software architecture, Nov. 2005.

[10] D. Knuth, *Literate Programming*, 1992, *pub: Center for the Study of Language and Information - Lecture Notes*, ISBN 0-937-07380-6

[11] D. Knuth, *The CWEB System of Structured Documentation, Version 3.0*, 1993, *pub: Addison-Wesley Professional* ISBN: 0-201-57569-8