ISSN 1744-1986

Technical Report N° 2006/ 18

Towards Context-Aware Product-Family Architectures

Mohammed Salifu
Bashar Nuseibeh
Lucia Rapanotti

30th December, 2006

Department of Computing
Faculty of Mathematics, Computing and Technology
The Open University

Walton Hall, Milton Keynes, MK7 6AA
United Kingdom

http://computing.open.ac.uk

Department of Computing
The Open University
Walton Hall
Milton Keynes, MK7 6AA

Pilot Study

Device Mobility Management Problems

Towards Context-Aware Product-Family Architectures

Mohammed Salifu

Supervised by Prof. Bashar Nuseibeh & Dr. Lucia Rapanotti

December 2006

Content **Page**

Towards Context-Aware Product-Family Architectures

1. Introduction

This pilot study examines a sample of the Nokia phone product-line to establish sources of its variability points, types of variability points, types of dependencies existing between these points and how variability points are represented. The number of variants in a product-line can be large; and in the case of the Nokia phone product-family it has been said to be very large [5] due to the wide range of phones with over 1000 different handsets operating in all continents. In the study, an initial assessment will be made of the impact of transforming the current (static) product-line approach which designs mobile phones for targeted fixed operating context into one that develops phones for variable operating contexts. By context-aware phones, we mean phones that adapt or adjust their behaviour in response to changes in their operating environment such as changes in location and/or network resources. It is agreeable to say that current mobile phones transmit and receive signals using the nearest available masks and can therefore be said to be location sensitive. However, in terms of software application development, this form of adaptation does not present any challenge in terms of variability points management as all phones irrespective of its targeted segment is expected to undergo this type of adaptation. Hence, this form of adaptation therefore represents a common core asset (or property) of all mobile phones. Our notion of context requires different phones to undergo different adaptations for multiple reasons such as satisfying different market segments (or enable the user to play a different role) and not just to maintain continual signal presence. This therefore requires the explicit representation of all possible operating contexts and their associated requirements, and using such representations in establishing the trade-offs of making a variability point static or dynamic and in choosing the required variants at such points in the generic architecture.

2. Why the Nokia phone product-family?

The Nokia phone product-family is chosen for the pilot study due its wide range of phones; between 500-600 million Nokia phones are estimated to be used globally [2]. This provides a rich source of varied data covering a range of variability sources suitable for the areas to be investigated in our research. Some sources of variability in producing the Nokia mobile phones are discussed next.

2.1 Source of variation in the Nokia Product-Family

Diversifying phones may provide opportunities and access to new market segments, but doing so presents new challenges by way of managing and keeping track of the variabilities across phones. Some sources of these variabilities [6], which are briefly discussed next are:

Physical handset categories

This refers to differences in characteristics such as memory size, processor speeds, weight, or phone case colours. Not all of these, e.g. case colours, are relevant to the underlying software system.

Communication standards (protocols)

This refers to differences in communication protocols, each enabling specific type of network connectivity. Examples are Code-Division Multiple Access (CDMA), Time-Division Multiple Access (TDMA), and Wideband CDMA (WCDMA) [9]. Phones with dual protocols capability can host two of these at a time thereby widening the area of coverage.

User interface

This refers to differences in areas such as screen display dimension, screen colour co-ordinations, number of keys on the keypad and input methods.

Operating system (platform)

Since each phone is (currently) capable of running only one operating platform, all phones are derived based on one of the four main types of platforms. These are Series30, Series40, Series60 and Series80, each of which will be discussed in the next section of this document.

Country customisation

This is largely based on the need to customise phones for the language and (available) communication protocols of the country in which it is to be used. For instance, Nokia currently support over 60 different languages across the globe and the number is expected to rise [5]. Some of these languages require different input methods. For instance, most Asian languages requires special combination of input keys presses to generate a single symbol, while most western languages have standard input methods (suck as key strokes).

3. Categorisation of the Nokia phone product-line

Nokia uses three main types of classifications to describe its phones. The first is based on the operating platforms of the phones, the second on the nature of the application domain and the third the functionalities or feature sets of the mobile hardware device (and the associated software components). In this context, an application domain refers to the categorisation of the usage context such as home usage (i.e. home domain) or business usage (i.e. enterprise domain). Examples of device functionalities (or feature set) based classification are entry-level (i.e. mobile phones with minimal functionalities such as making a phone call and sending basic text messages) and high-end or smart phones (i.e. phone with added functionalities beyond the basic and could include features such as cameras, FM radio, etc). Each of these categorisations is briefly discussed next.

3.1 Categorising phones by the operating platform (Design oriented)[1]

In this case, Nokia uses four main categories corresponding to operating system platforms S30, S40, S60, and S80 [7] (See appendix B for further details). Each platform may have several versions with each version offering different application libraries for system developers [8] (See appendix G for further details). Each of these groups is briefly discussed next.

Series 30: Series 30 is based on Nokia's proprietary Operating System, designed for entry-level Nokia phones. The platform is optimised for easy-to-use products that target first time users in, for example, new growth markets. As Series 30 has a significant memory cost benefit compared to fully featured platforms, it is highly cost-effective and caters for customers seeking lower price points. Due to being both affordable and easy-to-use, Series 30 is an ideal platform for enabling millions of first time users to go mobile and gain the socio-economic benefits of mobility.

Series 30 serves best for voice-centric products with simple text and picture messaging functionality - both in GSM and CDMA markets. Series 30 has an intuitive, visually attractive user interface, with graphics enhancements and selected value-adding features like built-in games.

Series 40: Series 40 is based on Nokia's proprietary Operating System designed for Nokia phones. Series 40 is a highly configurable and flexible software platform that enables a wide variety of different user interface styles and displays, product concepts and feature configurations. This enables

[1] [1] The discussion on each of the platforms is a summary of the articles published at Nokia Forum, 2005, Platforms; http://europe.nokia.com/nokia/0,,62626,00.html

the provision of distinctive, innovative products for different product categories, target users, and markets. For example, the Series 40 platform allows for the customization of devices for Nokia's different operator customers with distinctive operator branded soft-keys, operator logos and ring tones, colour schemes and wallpapers, wake up graphics, bookmark & link delegates, and data settings.

Java, XHTML Browser, and Multimedia Messaging Service (MMS) open the platform to the developers' community.

Series 60: S60 platform offers a feature rich software base for phones with advanced data capabilities. Optimized for the Symbian operating system and available for OEM licensing, S60 platform is a source code product that customers can port and integrate into their own hardware designs in order to produce advanced data capable handsets. S60 platform supports multitasking between applications and provides the user interfaces for switching between running applications and starting new ones while others are running.
S60 platform provides an extensive set of enabling technologies, such as GSM/GPRS/EDGE and WCDMA, Multimedia Messaging Service (MMS), Email, HTML and XHTML browsing, streaming and certain Java™ libraries, that facilitate interoperability between terminals and foster the use of ARPU generating mobile services.

Series 80: The Series 80 platform is a high-end platform for enterprise Communicators and smart-phones. It is based on the same Symbian OS as S60 platform but implements a two-hand-operated feature platform with full QWERTY keyboard, enabling richer user interface with widescreen and multiple applications available instantly. The Series 80 platform has been on the market since 1999 with the introduction of the first Symbian based Communicator, the Nokia 9210 Communicator. Further on, it has been implemented for example in the Nokia Communicator, Nokia 9290 Communicator, Nokia 9500 Communicator and Nokia 9300 smart-phone.

The Series 80 platform has been optimized for enterprise productivity applications like email, calendar, contacts/phonebook, printing and Microsoft Office™ compatible applications for Word document, spreadsheet, and presentation creation and editing.

3.2 Categorising phones by the application domain (Requirements Oriented)[2]

Four types of usage domains (or contexts) have been promoted by Nokia. These are, Mobile, Enterprise, Home and Internet domains. Each of these is briefly discussed next.

Mobile Domain: The Mobile domain provides person-to-person communication and data transport services. Mobility management, authentication, charging, and other value adding functionalities supported by the cellular network infrastructure offer a basis for a wide range of services throughout the Nokia domains. The mobile domain aims to provide richer communications through various media; voice, text images, streaming video and combinations thereof.

Enterprise domain: In the Enterprise domain, the primary user – usually an employee - connects to IT services, business applications, other employees, and customers. Employees need various types of enterprise applications, like corporate Personal Information Management (PIM) and communication services (such as instant messaging and email), databases, business services, and collaboration tools,

[2] [2] The discussion on each of the domains is a summary of the articles published at Nokia Forum, 2005, Domains; http://europe.nokia.com/nokia/0,,62626,00.html

as well as Internet services. Access to these services can take place in various environments - e.g. at home, at the office, or at the airport - using various devices and access channels, such as wireless networks, Wireless Local Area Networks (WLANs), and fixed broadband access.

Home domain: The Home domain includes communications, collaboration, and content sharing within the physical boundaries of the home as well as within the "extended home". Family- or home-related activities happening outside the home are considered part of the extended home. The extended home environment is becoming more significant as consumers want to access information and entertainment, view and share self-created content, and exchange messages with family members regardless of time or place. Nokia aims to provide the control core. In a digital home, users can potentially create and share media, control the entire home AV (audio- visual) system and even do virtual visits to remote locations.

Internet Domain: In the Internet domain, all the services are provided by a multitude of companies, organizations and even individual users. The evident driving force in the Internet domain is the increasing consumption of media in addition to traditional e-mail, WWW services and instant messaging. One is able to publish his/her own content and interact with friends and communities regardless of time, place or connection type.

3.3 Categorising phones by device specification for different Segments (Requirements Oriented)[3]

The following gives a brief discussion of five different segments under this classification:

Entry level: This refers to phone with basic features such as the ability to make a phone call and to send a text message. This category of phones also has limited gaming and basic user interface features. Examples of Nokia entry phones are 3310 and 1100.

High End: This category is made up of phones of advanced features such as the ability to run third party application such as Microsoft Word applications and advanced user interface features such as voice messaging. Examples of phones of this category are 6600 and 6610.

Enterprise: This is primarily targeted at the co-operate world aimed at providing continual business process while on the move. Therefore, in addition to having the ability to run third party application such as MS Office applications, they are also equipped to secure their communication channels with facilities such as the use of virtual private networks (VPN). Examples of phones in this category are 6800 and 9500.

Fashion: The features that uniquely identify this category are largely user interface driven designed to appeal to users who pay special attention to the external observable features of a phone such as cover case colour and keypad layout. Examples of phone in this category are Nokia 3650 and 7250.

Rough: This is designed for people who are mostly outdoors or put phones under physicals strain. Most of the specialised features are hardware based and cover areas such as shock and water-resistant covers and scratch-proof display. However software also plays some part in allowing the phone to be used for other services such as stopwatch and as a camera. An example of this is 5410.

[3] [3] The discussion on each of the domains is a summary of the articles published at Nokia Forum, 2005, Domains; http://europe.nokia.com/nokia/0,,62626,00.html

4. Dependency relations between platforms and segments

This section briefly discusses sample features across the current Nokia operating platforms (solution space) and uses them to derive sample phones belonging to different market segments (requirements space). A feature in this context refers to an observable functional unit such as an FM radio or camera built into the phone. Also, for simplicity, an assumption is made that, each feature is implemented in a single component. This allows us to simplify and surface some dependency relations between platforms and segments and within segments (*Appendixes C and D provide sample lists of phones for the four main categories or sub-families of Nokia phones along with their feature and general device specifications*).

The following classes of features have been identified in the pilot study and will be used to facilitate the discussion of dependency relation between platforms and segments.

Type of features:

1. Mandatory: These features must be present in every released product such as being able to make calls and send texts.
2. Optional: These features may be included in released products but no obligation to do so. E.g. FM radio and Camera.
3. Single Alternative: Choosing a feature from two alternative features but only one must be chosen at a time. E.g. choosing to have either an emergency or a conference call button but not both.
4. Multiple Alternatives: Choosing a sub-set of features from a larger set such as choosing to have (GSM 1800 AND GSM 1900) or (GSM 900 AND GSM 1800). However, a single feature such as GSM 1800 could be selected but at least one must be selected.

Note: alternative (single or multiple) could be optional or mandatory depending on the choice of other features. E.g. both the FM radio and RealPlayer features are optional but only one of them can be chosen in a given phone as they are currently available only in different platforms.

Key

Colour Code	Feature
	Mandatory
	Single Alternative-Mandatory
	Optional Feature
	Optional single alternative
	Multiple alternative

4.1 Nokia Phone Product-Family Features by Segments

Segments vs. Features	SMS	MMS	call waiting	Call Forwarding	Conference calls	Emergency calls	Extendable memory capability	Wireless LAN Access	FM Radio	Real-One Player	Camera	Video Recorder	Hands Free	Multiple Connectivity options	Multiple Band/Dual Band	Java Applications	Email, Personal Information Management (PIM), HTML 4.01 / xHTML browser	processor, spreadsheet, and presentation editor, and viewer (Microsoft compatible, MS Office 97	Security solutions: SSL/TLS, VPN, Ipsec, WPA
Enterprise	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X
Entry	X		X	X	X	X									X				
Fashion	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
High End	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			

Table 1 Nokia Phone Product-Family Features by (Market) Segments
X = this feature is available for segment

4.2 Nokia Phone Product-Family Features by platform

Platform vs. Features	SMS	MMS	call waiting	Call Forwarding	Conference calls	Emergency calls	Extendable memory capability	Wireless LAN Access	FM Radio	RealOne Player	Camera	Video Recorder	Hands Free	Multiple Connectivity options	Multiple Band/Dual Band	Java Applications	Email, Personal Information Management (PIM), HTML 4.01 / xHTML browser	processor, spreadsheet, and presentation editor, and viewer (Microsoft compatible, MS Office 97	Security solutions: SSL/TLS, VPN, Ipsec, WPA
Series 30	X		X	X	X	X									X				
Series 40	X	X	X	X		X		X	X		X		X	X	X	X	X		
Series 60	X	X	X		X		X	X		X	X	X	X	X	X	X	X		
Series 80	X	X	X		X		X	X		X	X		X	X	X	X	X	X	X

Table 2 Nokia Phone Product-Family Features by Platform
Key: X = this feature is available in this platform

Note that the entire features set of a product must be realisable from a single platform as a hardware device (phone) is capable of accommodating or hosting only one platform at a time. Taking this into the account and the available features in each platform and in each segment, as well as their dependency (i.e. mandatory or alternative etc), table 3 gives a list of sample products (phones) as currently available in the markets.

4.3 Nokia Phone Product-Family Features- Selected product (phones)

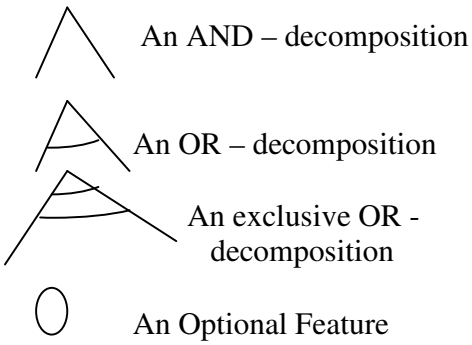
Products vs. Features	SMS	MMS	call waiting	Call Forwarding	Conference calls	Emergency calls	Extendable memory capability	Wireless LAN Access	FM Radio	Real-One Player	Camera	Video Recorder	Hands Free	Multiple Connectivity options	Multiple Band/Dual Band	Java Applications	Email, Personal Information Management (PIM), HTML 4.01 / xHTML browser	processor, spreadsheet, and presentation editor, and viewer (Microsoft compatible, MS Office	Security solutions: SSL/TLS, VPN, Ipsec, WPA
Enterprise - 6800	X	X	X	X		X		X	X				X	X	X	X	X		
Enterprise - 9500	X	X	X		X		X	X		X	X		X	X	X	X	X	X	X
Entry - 1100	X		X	X	X										X				
Entry-3310	X		X			X													
Fashion - 3650	X	X	X		X		X	X		X	X	X		X	X	X	X		
Fashion - 7250	X	X	X	X		X		X	X		X		X	X	X	X			
High End - 6600	X	X			X		X	X		X	X	X	X	X	X	X			
High End - 6610	X	X	X	X		X		X	X				X	X	X	X			

Table 3 Nokia Phone Product-Family Features- Selected product (phones)

5. Feature -Diagram Representation of the Nokia Product-Family Features (Problem Space)

In this section we illustrate the current (predominant) use of feature diagrams to represent variation and variability points in software product-families by using the Feature Oriented Domain Analysis (FODA) [10] notation to model the Enterprise segment from table 1. FODA is the pioneer language in feature oriented system development and have been the basis for numerous other feature representation notations [10]. This representation is a requirement (or problem space oriented) and variation points identified in feature diagrams do not necessarily translate to variants in the generic architecture. This is largely due the fact that a given feature may be implemented in more than one component and that single component could implement (or represent) more than one feature. Further discussion of this will be made in section 9.

Key:



FODA representation of features of the Enterprise segment (details in table 1 section 4.1):

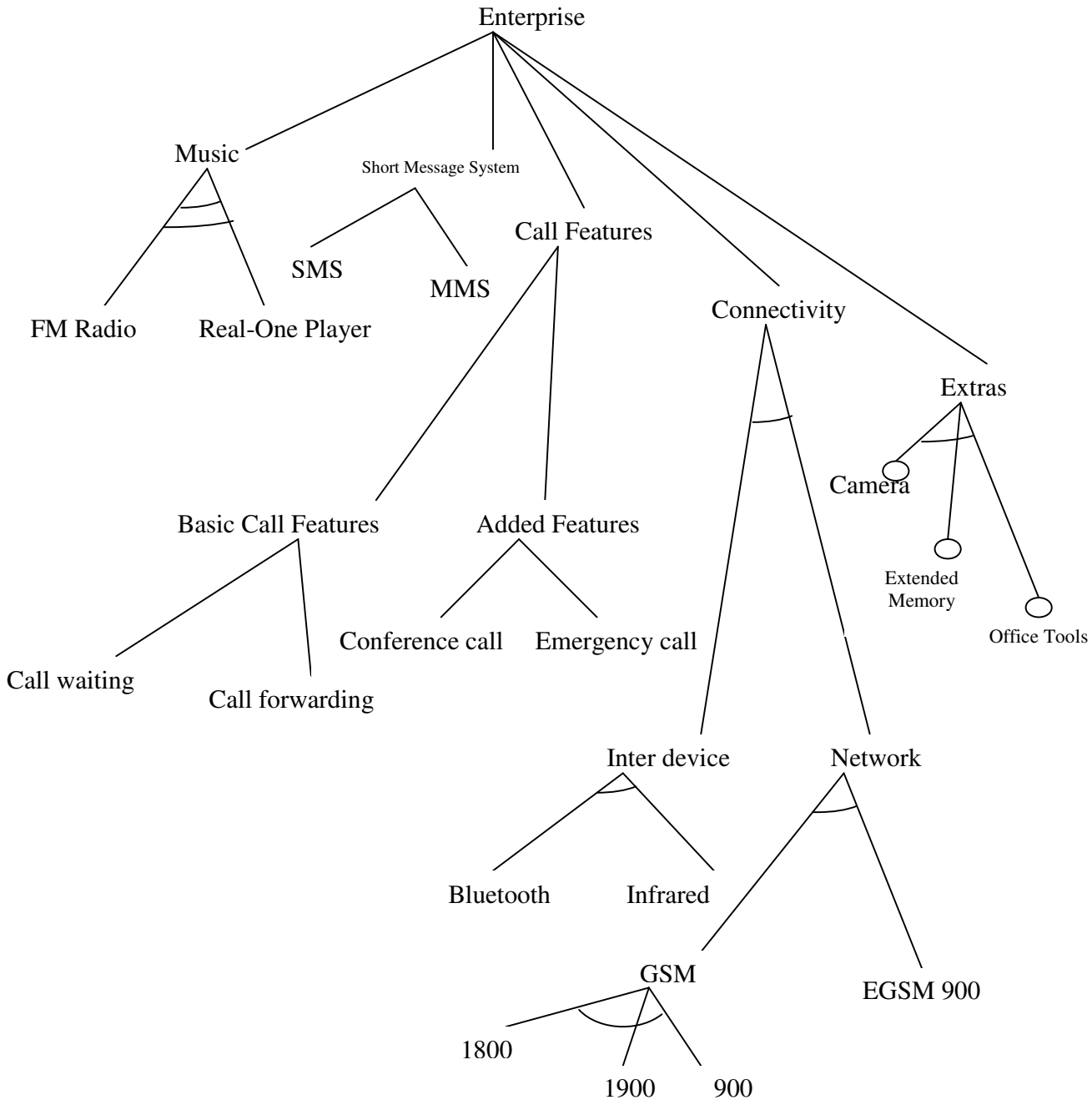


Figure 1, FODA representation of features of the Enterprise segment

6. Architectural Level Representation of the Nokia Product-Family Features (Solution Space)

This section briefly represents some static variants and variability points using the architectural notation in [1]. The sample of features represented here are such that the differences in them are visible to the consumer such as the difference between FM Radio and Camera. However, much as a text-editor feature may appear the same to different consumers using different phones based on different platforms, their implementation may well be different. This type of differences (in features) is not represented in here but will be considered as part main research study. Further discussion of this will be made in section 9.

The following types of variants are briefly discussed (a brief discussion of the notations used can be found in Appendix A). For simplicity, an assumption is made that every feature is implemented in exactly one component or that sub-components are abstracted away.

Mandatory: This is represented using MMS (multimedia messaging system) for architecture 1 and Camera & MMS features for architecture 2. Architecture 1 could be used to send messages downloaded to the phone while architecture two allows the phone to be used to take pictures which are subsequently sent as part of MMS.

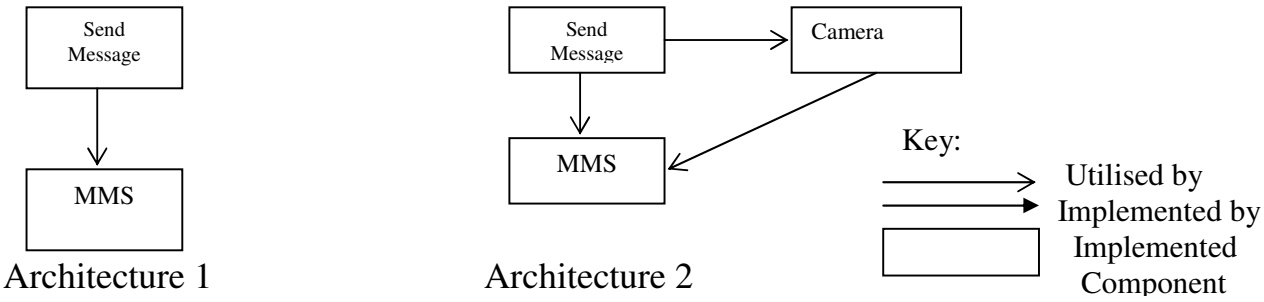
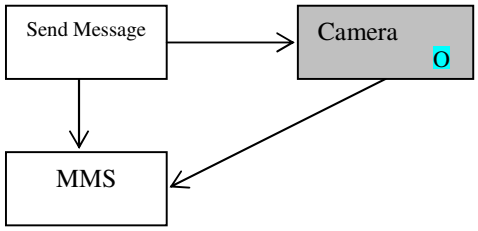


Figure 2, (mandatory components) architectures

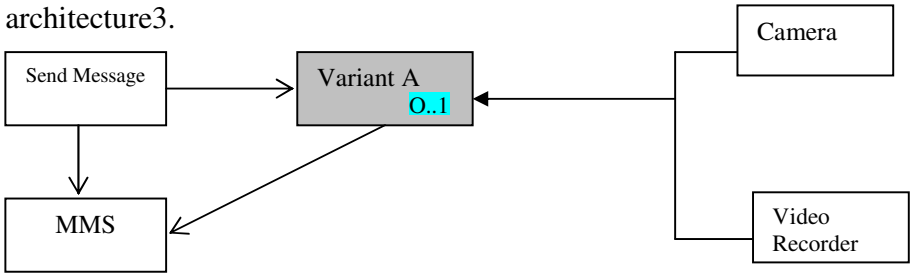
Optional: This is represented by combining the two architectures in figure 2.



Architecture 3, a camera feature may be included.

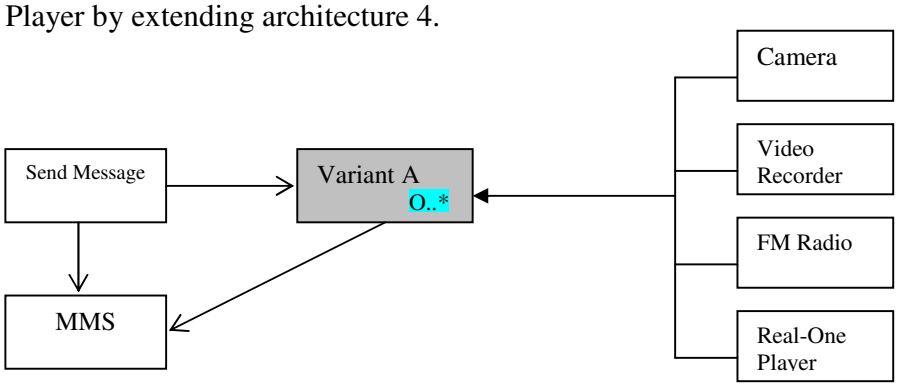
Figure 3, optional variant architecture

Optional alternative: This is represented as a camera and a video recorder by extending architecture3.



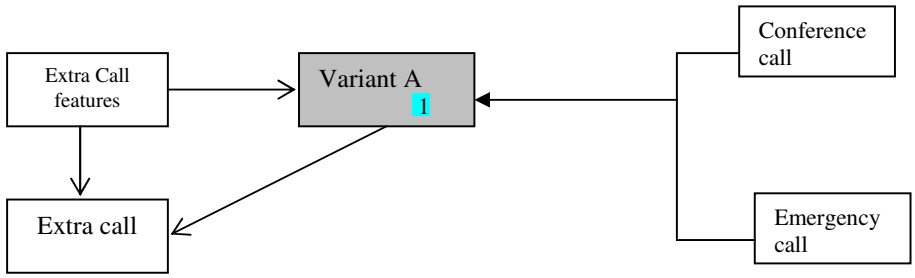
Architecture 4, either a camera or video recorder is included but not both.
Figure 4, optional alternative variant architecture.

Optional alternatives: This is represented as a camera, a video recorder, a FM Radio and Real-One Player by extending architecture 4.



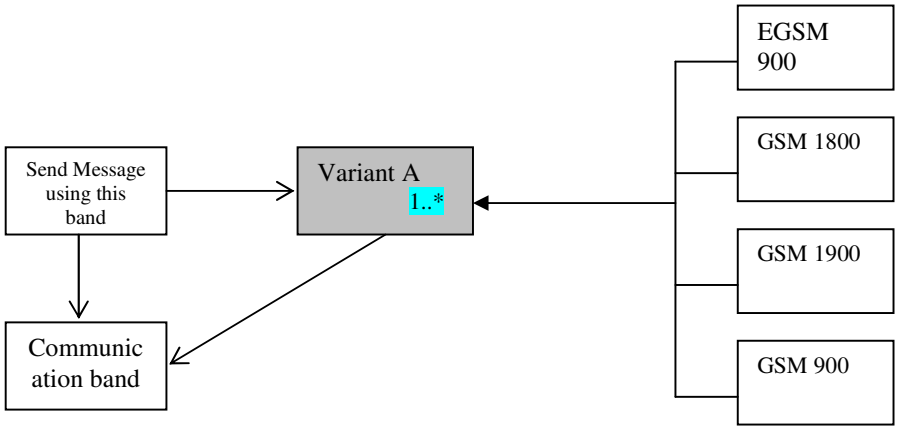
Architecture 5, more than one feature may be included; i.e. consumer could have both a camera and video recorder in this phone.
Figure 5, optional alternatives architecture

Single alternative: This is represented conference and emergency calls as extra features.



Architecture 6, either a conference call or an emergency call feature must be included but not both.
Figure 6, single alternative architecture

Multiple alternatives: This is represented different type of network connectivity features.



Architecture 7, more than one communication band could be selected but at least one must be selected.

Figure 7, multiple alternatives architecture.

7. Variability points dependency relations of the current Nokia Product-Family

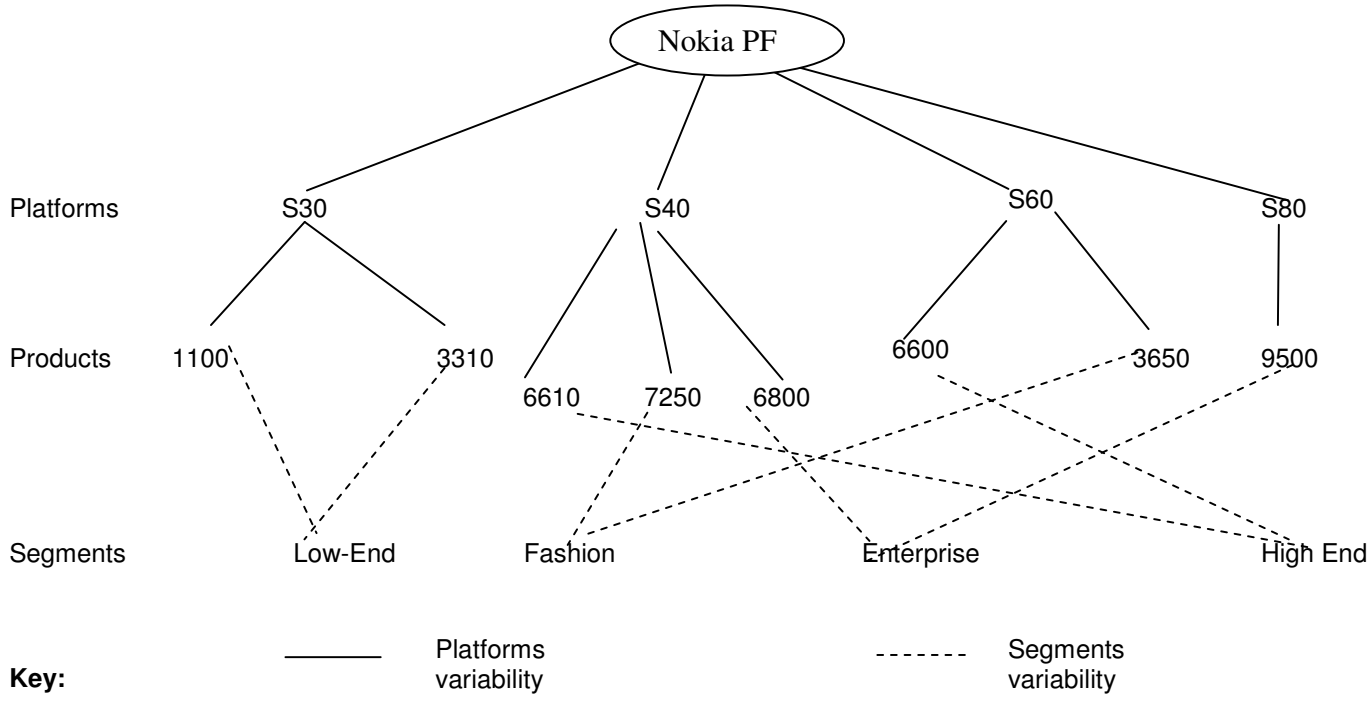


Figure 8, variability points in segments (requirements) and platforms (architecture)

From the above (tree like) representation of the Nokia product-family in terms of both problem and solution spaces, it can be seen that the classification based on the available platform does not mirror that of the segments. For instance, phones 7250 and 3650 are both classified at the requirements level as (belonging to segment) Fashion even though 7250 is based on the series 40 platform whereas 3650 on series 60. Also, we can conclude that consumers can only select a set of features if the entire members of the set come (or can be realised) from a single operating platform such as series 40 or

Series 60. This is due to the (limited) ability of current mobile phones to run or host only a single operating platform at a time.
Since we do not have access to the implementation (or architectural) details of each of the implemented features in the operating platforms, we are unable to conclude that a text-editor feature in different operating platforms are implemented differently. However, by looking at the diagram together with tables 1-3, gives the impression that, a text-editor feature in a low-end phone (which appears to be entirely implemented on only series 30 platforms) is somehow different from one implemented in series 40 and 60. Also, if we are to take the classification of requirements (or features) into segments as representing members of equivalent status within a segment (i.e. to say that all text-editor components in all Fashion segment phones are the same), then we could conclude that, a text-editor feature implemented in series 40 and 60 are equivalent as different members within the Fashion category are implemented in both platforms.
In summary therefore, we can conclude that, much as the representation of features as classes of segments (requirements) and platforms (architectures) shows possible differences and dependencies between these two classifications, further details are required to establish, if variabilities or difference in requirements classifications necessarily translate to differences in the generic architecture. For now and without any conclusive evidence, it is safe to say that differences in features classifications may or may not result in differences in architectural elements such as components, connectors, etc.
Also, the difference in the classification of features by segments and platforms induces dependency relations between such classifications which in this case constraints the choice of consumers of the product (i.e. phones). For instance, even though fashion (segment) phones could be implemented in both platforms 40 and 60, the set of features for any given fashion category phone must entirely be realisable in a single platform. This fact (or indicative property) needs to be made explicit when the context(s) in which the product is to be used is considered (during requirements analysis) even though it is architectural in nature. Failure to make it explicit will lead to requirement specification or architectural design that cannot be implemented due to this immutable constraint of the solution space. This argues for the explicit capture of all possible operating contexts and existing immutable architectural elements such as components and connectors or constraints in determining and assessing the trade-offs of making variability points in the architecture static or dynamic. This is one of the issues to be considered in our research.

8. Summary of some challenges in transforming the Nokia Product-Family into context-aware Product-families

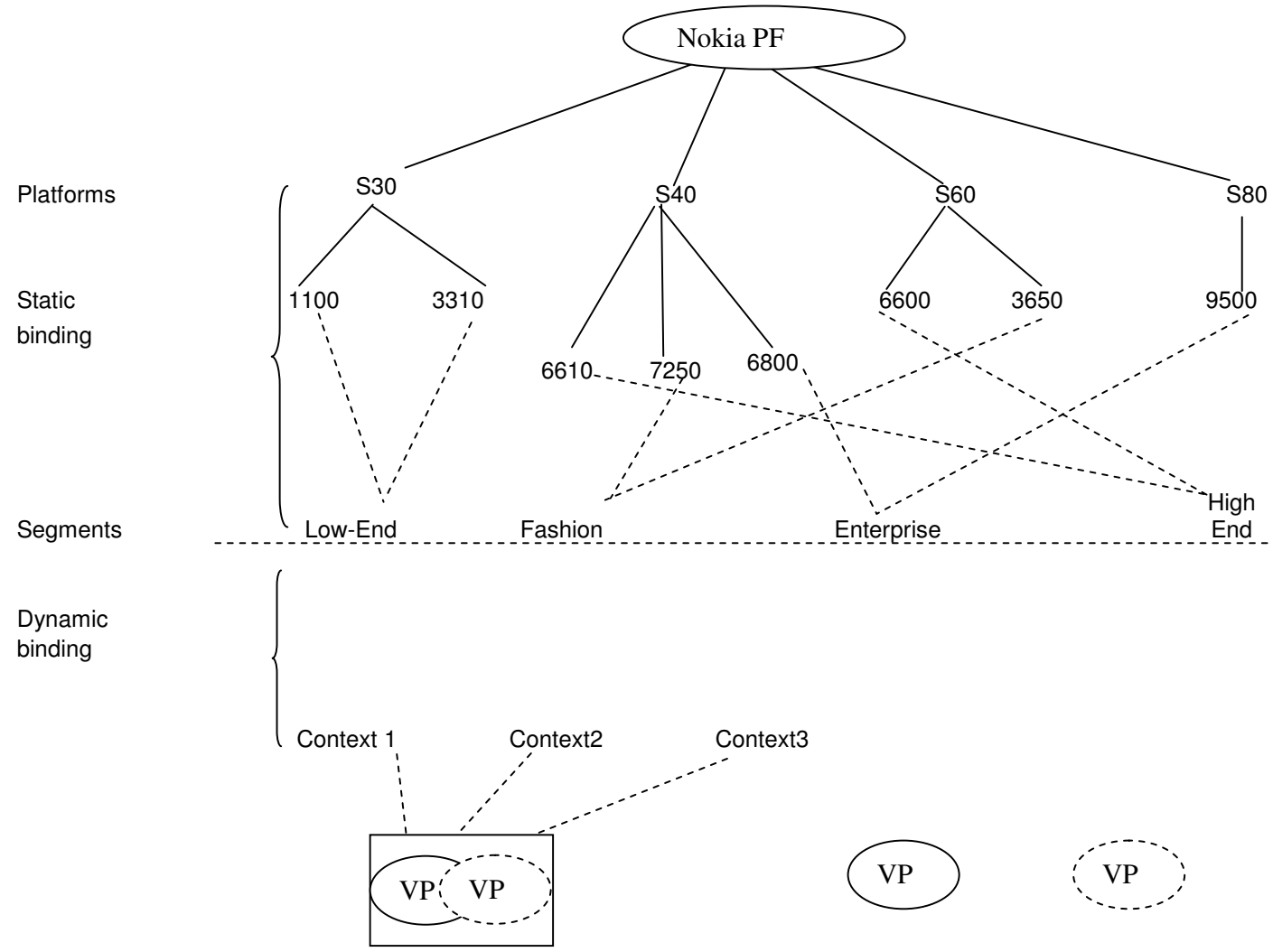


Figure 9, static and dynamic binding

Considering the discussion in section 6, which largely seeks to highlight differences in the classification of features into segments and platforms and the dependency relations it has created, the obvious question is how one assesses the impact of multiple classifications on variability points design and their trade-off assessments. Therefore, part of the issues to be addressed is to establish the extent to which one can capture and reason about the contextual information such as requirements of segments, contexts of operations and existing immutable architectural elements and/or constraints in

these contexts for different products in a given segment. This will enable us to establish if classifications in existing architectural elements has any impact in assessing and designing variability points (static and dynamic) and the extent of their influence if any or that it is only requirements (or segments) classifications that is of relevant in doing so.

Summary of challenges:

1. Variability points (static or dynamic) could be optional, single and multiple alternatives, how does one represent these?
2. How does one represent contexts and the associated requirements?
3. How does one assess the trade-offs among different contexts at variability points?
4. How does one choose to make a variability points static or dynamic based on 3?

9. What is meant by N-Dimensional & Hierarchical Product-Families?

The dependency relations and the resulting constraints resulting from difference in classifications of features (in the case of Nokia phones by segments and platforms as well as by domains) has been reported by Thompson and Heimdahl [11] to have been observed earlier by [3, 4]. They claim this is largely driven by differences in the objectives and purposes for which such classifications are meant to serve. The observation in this pilot study appears to confirm this. For instance, the classification of features into segments such as fashion and enterprise appears to be marketing oriented and largely driven by the total cost of the product. Similar argument could be made of the classification of features into domains such as home and internet, which appears to be usage contexts driven and also serve as medium of communication to potential consumers of the products. Likewise, the classification of features into platforms is technologically driven and does not totally reflect the other classifications. However, it will be naïve to assume that these other classifications have no bearing in how the platform based classification is structured.

As a result of this multiplicity of classification and the resulting dependencies, Thompson and Heimdahl have proposed the concepts of n-dimensional and hierarchical product-families. By hierarchical, they mean, instead of a single all encompassing generic product-family architecture, the architecture is broken into a collection of sub-families which together make up the generic architecture of the entire family. Under this concept, each platform (i.e. S30, S40, S60 and S80) will represent a sub-family of the Nokia generic family architecture. This by definition creates levels of variabilities and commonalities. That is, in choosing a platform, S30 to S80 are seen as variants of the variability point (platform). However, all products in each platform irrespective of their segment's classification will consider the platform as common and not variable. This highlights the complexity involvde in determining what is to be made static or dynamic variability point and the depth (i.e. extent) of the variability in the overall product-family. By n-dimension, they refer to the multiplicity of relations or dependencies (i.e. factors to consider) in determining sub-families of the entire generic architecture.

In summary, we argue that the difficulty is in how to capture the relevant classifications and the possible operating contexts and use them to design variability points and in assessing the trade-offs in doing so. The formulation of the appropriate architectural level notation, the capture of operation contexts and their associated requirements and the development of conceptual tools for representing and reasoning about variability points are needed for effective and efficient variability points design and management. Therefore, the overall aim of our research is to extend current static product-family architectures into context-aware, capable of handling both static and dynamic variability points.

10. Summary and conclusion

Going through the pilot study has enhanced our understanding of the practical application of the product-family paradigm in the mobile phone sector. Even though it was centred on the Nokia product-family, the understanding gained will be useful in investigating other product-families in different application domains.

To facilitate our reasoning of the types of variability points and their dependencies in the Nokia product-family, we used the Feature Oriented Domain Analysis (FODA) notation to model the Enterprise (market) segment of the (sampled) Nokia phone features. This gave us a requirements (i.e. problem space oriented) view of variability points design and management. We also used the architectural notation proposed by Bachmann and Bass to represent architectural level variability points for the Nokia product-family. As this latter representation is architectural, it is said to be solution space oriented.

The key observation and lesson learnt is that, multiple factors such as cost and usage domain are as important as technological factors in classifying features of products. In the case of Nokia product-family, this has resulted in three classifications. These are by the operating platforms, the device specifications (i.e. by market segments) and the usage domains each of which has some degree of influence in how the others are structured. This in turn has resulted in increased complexity in the design and representation of variability points and in determining the depth of such variabilities. It is this multiplicity of issues which has lead to the proposed concept of hierarchy of families by Thompson and Heimdahl (see section 8 for further details). That is, the partitioning of product-family into sub-families. We argue that, these two issues together (i.e. multiple classifications and hierarchy of families and the resultant dependencies) add (further) complexity to the design and representation variability points. Extending such product-families architectures to be capable of handling both static and dynamic variability points thereby making them suitable for the design of context-aware architectures will further increase the complexity of variability design, representation and management

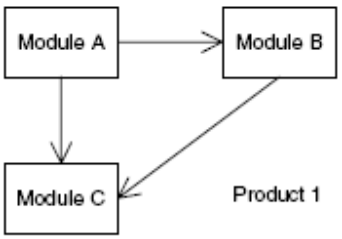
In conclusion, we claim that ability to design and represent variability points for context-aware product-family architectures is centred on the existence of architectural conceptual tools in doing so and reasoning about the trade-offs in determining what to make static or dynamic and the depth of such variabilities in the family or sub-families.

References

1. Bachmann, F. and L. Bass. *Managing variability in software architectures*. in *SSR'01*. 2001. Toronto, Ontario, Canada: ACM Press.
2. Bosch, J. *Software product families at nokia*. in *SPLC05*. 2005.
3. Brownsword, L. and P. Clements., *A case study in successful product line development*, in *Report CMU/SEI 96-TR-016*. 1996, Carnegie Mellon University, Software Engineering Institute Technical.
4. Lutz, R.R., *Extending the product family approach to support safe reuse.* Journal of Systems and Software, 2000. **53**: p. 207-217.
5. Maccari, A. and A. Heie, *Managing infinite variability in mobile terminal software.* SOFTWARE—PRACTICE AND EXPERIENCE, 2005. **35**(6): p. 513–53.
6. Maccari, A. and A.-P. Tuovinen. *System family architectures: Current challenges at nokia*. in *Software Architectures for Product Families: International Workshop IW-SAPF-3*. 2000. Las Palmas de Gran Canaria, Spain, March 15-17, 2000. Proceedings: Lecture Notes in Computer Science,Publisher: Springer Berlin / Heidelberg.

7. Nokia, *Nokia architecture - platform architectures*. 2005, Nokia - http://europe.nokia.com/nokia/0,,62611,00.html: Online. p. 1-2.
8. Nokia, F., *Enterprise: Developing end-to-end systems*. 2006, Nokia Forum: Online. p. 1-54.
9. Nokia, F., *Series 80 developer platform: Porting from series 60*. 2005, Nokia Forum: http://www.forum.nokia.com/porting. p. March 14.
10. Schobbens, P.-Y., P. Heymans, and J.-C. Trigaux, *Feature diagrams: A survey and a formal semantics.* To appear: REquirements Engineering '06, 2006: p. 1-10.
11. Thompson, J.M. and M.P.E. Heimdahl, *Extending the product family approach to support n-dimensional and hierarchical product lines**. Fifth IEEE International Symposium on Requirements Engineering (RE'01), 2001: p. 56--66.

Appendix A: Bachmann and Bass architectural notations for variability points [1]



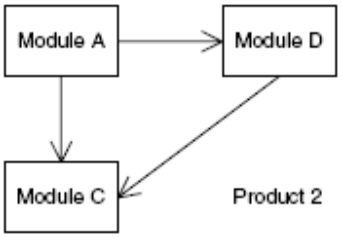


Figure 1 Alternative architectures for two products

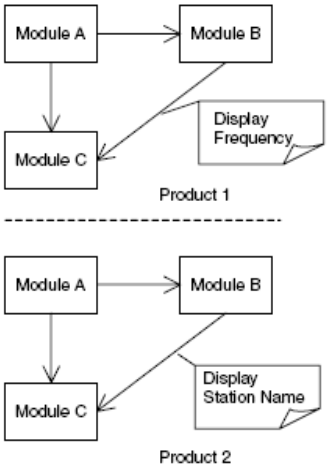


Figure 4 Variation of relation

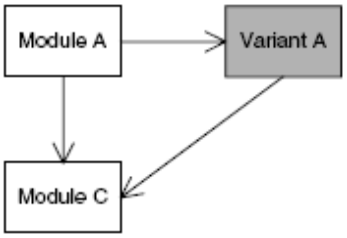


Figure 2 Variation in architecture

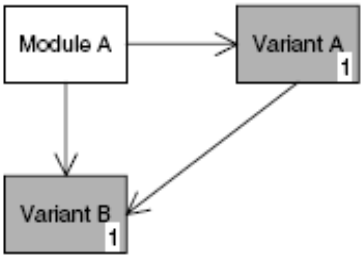


Figure 5 Variation of relation as variation of components

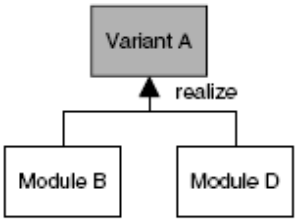


Figure 3 Alternatives for Variant A

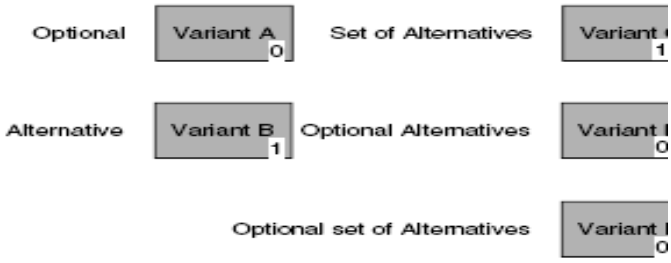


Figure 8 Notation for types of variants

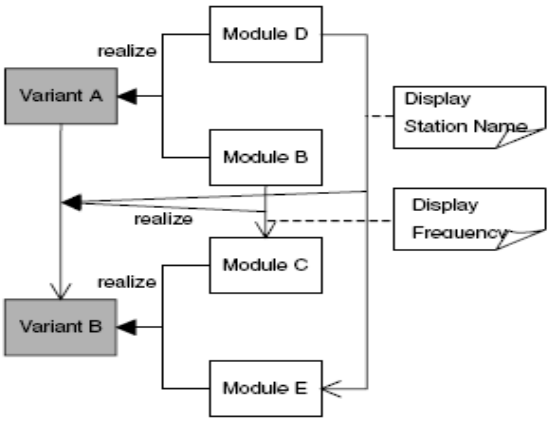


Figure 6 Constraints between alternatives of variants

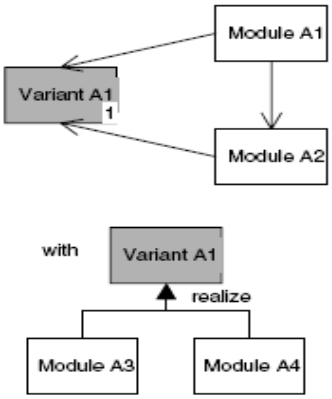
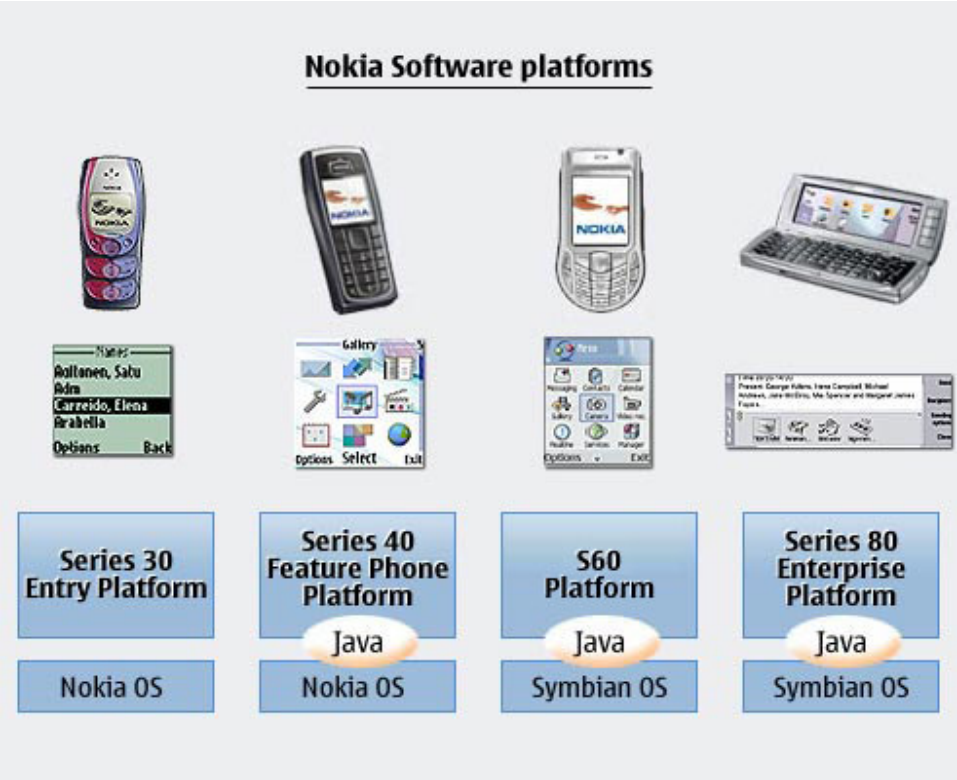


Figure 7 Decomposition of Variant A

Appendix B – Nokia Software Platforms



Series 30

Series 30 is based on Nokia's proprietary Operating System, designed for entry-level Nokia phones. The platform is optimised for easy-to-use products that target first time users in, for example, new growth markets. As Series 30 has a significant memory cost benefit compared to fully featured platforms, it is highly cost-effective and caters for customers seeking lower price points. Due to being both affordable and easy-to-use, Series 30 is an ideal platform for enabling millions of first time users to go mobile and gain the socio-economic benefits of mobility.

Series 30 serves best for voice-centric products with simple text and picture messaging functionality - both in GSM and CDMA markets. Series 30 has an intuitive, visually attractive user interface, with graphics enhancements and selected value-adding features like built-in games.

Series 40

Series 40 is based on Nokia's proprietary Operating System designed for Nokia phones. Series 40 is a highly configurable and flexible software platform that enables a wide variety of different user interface styles and displays, product concepts and feature configurations. This enables the provision of distinctive, innovative products for different product categories, target users, and markets. For example, the Series 40 platform allows for the customization of devices for Nokia's different operator customers with distinctive operator branded softkeys, operator logos and ring tones, color schemes and wallpapers, wake up graphics, bookmark & link delegates, and data settings.

Java, XHTML Browser, and Multimedia Messaging Service (MMS) open the platform to the developers' community.

Series 60

S60 platform offers a feature rich software base for phones with advanced data capabilities. Optimized for the Symbian operating system and available for OEM licensing, S60 platform is a source code product that customers can port and integrate into their own hardware designs in order to produce advanced data capable handsets. S60 platform supports multitasking between applications and provides the user interfaces for switching between running applications and starting new ones while others are running.
S60 platform provides an extensive set of enabling technologies, such as GSM/GPRS/EDGE and WCDMA, Multimedia Messaging Service (MMS), Email, HTML and XHTML browsing, streaming and certain Java™ libraries, that facilitate interoperability between terminals and foster the use of ARPU generating mobile services.

Series 80

The Series 80 platform is a high-end platform for enterprise Communicators and smartphones. It is based on the same Symbian OS as S60 platform but implements a two-hand-operated feature platform with full QWERTY keyboard, enabling richer user interface with widescreen and multiple applications available instantly. The Series 80 platform has been on the market since 1999 with the introduction of the first Symbian based Communicator, the Nokia 9210 Communicator. Further on, it has been implemented for example in the Nokia Communicator, Nokia 9290 Communicator, Nokia 9500 Communicator and Nokia 9300 smartphone.

The Series 80 platform has been optimized for enterprise productivity applications like email, calendar, contacts/phonebook, printing and Microsoft Office™ compatible applications for Word document, spreadsheet, and presentation creation and editing.

Appendix C – Sample Phones for Nokia platforms

Sample Series 40 Phones

		Display	Java	Developer Platform	Messaging	Browser
	Nokia 6131	240x320	CLDC 1.1 MIDP 2.0	S40 3rd Ed	MMS+SMIL SMS	HTML WAP 2.0 XHTML
	Nokia 6136	128x160	CLDC 1.1 MIDP 2.0	S40 3rd Ed	MMS+SMIL SMS	HTML WAP 2.0 XHTML
	Nokia 6282	240x320	CLDC 1.1 MIDP 2.0	S40 3rd Ed	MMS+SMIL SMS	WAP 2.0 XHTML
	Nokia 6233	240x320	CLDC 1.1 MIDP 2.0	S40 3rd Ed	IM MMS+SMIL SMS	WAP 2.0 XHTML
	Nokia 6234	240x320	CLDC 1.1 MIDP 2.0	S40 3rd Ed	IM MMS+SMIL SMS	WAP 2.0 XHTML
	Nokia 6165	128x160	CLDC 1.1 MIDP 2.0	S40 3rd Ed	MMS+SMIL SMS	Openwave WAP 2.0 XHTML
	Nokia 7370	240x320	CLDC 1.1 MIDP 2.0	S40 3rd Ed	MMS+SMIL SMS	WAP 2.0 XHTML
	Nokia 6280	240x320	CLDC 1.1 MIDP 2.0	S40 3rd Ed	MMS+SMIL SMS	WAP 2.0 XHTML
	Nokia 6265	240x320	CLDC 1.1 MIDP 2.0	S40 3rd Ed	MMS+SMIL SMS	Openwave WAP 2.0 XHTML
	Nokia 6265i	240x320	CLDC 1.1 MIDP 2.0	S40 3rd Ed	MMS+SMIL SMS	Openwave WAP 2.0 XHTML

Sample Series 60 Phones

		Display	Java	Developer Platform	Messaging	Browser
	Nokia N71	240x320	CLDC 1.1 MIDP 2.0	S60 3rd Ed	MMS+SMIL SMS	HTML WAP 2.0 XHTML
	Nokia N80	352x416	CLDC 1.1 MIDP 2.0	S60 3rd Ed	MMS+SMIL SMS	HTML 4.0 WAP 2.0 XHTML
	Nokia N92	240x320	CLDC 1.1 MIDP 2.0	S60 3rd Ed	MMS+SMIL SMS	HTML 4.0 WAP 2.0 XHTML
	Nokia E60	352x416	CLDC 1.1 MIDP 2.0	S60 3rd Ed	MMS+SMIL SMS	HTML 4.0 WAP 2.0 XHTML
	Nokia E61	320x240	CLDC 1.1 MIDP 2.0	S60 3rd Ed	MMS+SMIL SMS	HTML 4.0 WAP 2.0 XHTML
	Nokia E70	352x416	CLDC 1.1 MIDP 2.0	S60 3rd Ed	MMS+SMIL SMS	HTML WAP 2.0 XHTML
	Nokia 3250	176x208	CLDC 1.1 MIDP 2.0	S60 3rd Ed	MMS+SMIL SMS	HTML WAP 2.0 XHTML
	Nokia N90	352x416	CLDC 1.1 MIDP 2.0	S60 2nd Ed	MMS+SMIL SMS	HTML WAP 2.0 XHTML
	Nokia N70	176x208	CLDC 1.1 MIDP 2.0	S60 2nd Ed	MMS+SMIL SMS	HTML WAP 2.0 XHTML
	Nokia N91	176x208	CLDC 1.1 MIDP 2.0	S60 3rd Ed	MMS+SMIL SMS	HTML WAP 2.0 XHTML

Sample Series 80 phones

		Display	Java	Developer Platform	Messaging	Browser
	Nokia 9500	640x200	CLDC 1.1 MIDP 2.0	S80 2nd Ed	MMS+SMIL SMS	HTML XHTML

Appendix D – Sample phones specification and feature comparison

Sample Series 40 Phone Specification Comparison

Nokia 3530 Technical Specs	Nokia 6131 Technical Specs	Nokia 7250i Technical Specs	Nokia 6800 Technical Specs
Operating System:	**Operating System:**	**Operating System:**	**Operating System:**
Nokia OS	Nokia OS	Nokia OS	Nokia OS
Developer Platform:	**Developer Platform:**	**Developer Platform:**	**Developer Platform:**
Series 40 Developer Platform 1.0	Series 40 3rd Edition	Series 40 Developer Platform 1.0	Series 40 Developer Platform 1.0
Java Technology:	**Java Technology:**	**Java Technology:**	**Java Technology:**
CLDC 1.0 MIDP 1.0 Nokia UI API	CLDC 1.1 Wireless Messaging API (JSR-120) Mobile Media API (JSR-135) Mobile 3D Graphics API (JSR-184) JTWI (JSR-185) FileConnection and PIM API (JSR-75) Bluetooth API (JSR-82) MIDP 2.0 Nokia UI API	CLDC 1.0 Wireless Messaging API (JSR-120) MIDP 1.0 Nokia UI API	CLDC 1.0 MIDP 1.0 Nokia UI API
Browser:	**Browser:**	**Browser:**	**Browser:**
WAP 1.2.1	HTML over TCP/IP WAP 2.0 XHTML over TCP/IP	WAP 1.2.1 XHTML	WAP 1.2.1
Messaging:	**Messaging:**	**Messaging:**	**Messaging:**
MMS SMS	MMS+SMIL SMS	MMS SMS	MMS SMS
Sound Formats:	**Digital Rights Management:**	**UAProfile:**	**UAProfile:**
MIDI Tones (poly 4)	OMA DRM v1.0	Profile 1	Profile 1
Functionality:	**Delivery Method:**	**Digital Rights Management:**	**Sound Formats:**
GSM 1800 GSM 900	MMS WAP Download	OMA DRM Forward Lock	MIDI Tones (poly 4)
Regional Availability:	**Functionality:**	**Delivery Method:**	**Functionality:**
Asia-Pacific	GSM 1800 GSM 1900 GSM 850 GSM 900	MMS WAP Download	GSM 1800 GSM 900
Screen Display:	**Regional Availability:**	**Sound Formats:**	**Regional Availability:**
Color Depth: 12 bit Resolution: 96 x 65	Africa Americas Asia-Pacific China Europe	MIDI Tones (poly 4)	Africa Asia-Pacific Europe
Physical Descriptions:	**Screen Display:**	**Functionality:**	**Screen Display:**
Dimensions: 118 x 50 x 17 mm Weight: 106 g	Color Depth: 24 bit Color Depth: 18 bit Resolution: 240 x 320 Resolution: 128 x 160	GSM 1800 GSM 1900 GSM 900	Color Depth: 12 bit Resolution: 128 x 128
Memory:	**Physical Descriptions:**	**Regional Availability:**	**Physical Descriptions:**
Heap size: 195 KB Shared Memory for Storage: 176 KB Max JAR Size: 63 KB	Dimensions: 92 x 48 x 20 mm Weight: 112 g	Africa Asia-Pacific Canada Europe Latin America	Dimensions: 119 x 55 x 23 mm Weight: 122 g
Keypad Descriptions:	**Memory:**	**Screen Display:**	**Memory:**
2 Labeled Soft Keys 2-way Scrolling Grid Key Mat	Shared Memory for Storage: 11 MB Max JAR Size: 1 MB	Color Depth: 12 bit Resolution: 128 x 128	Heap size: 195 KB Shared Memory for Storage: 5 MB Max JAR Size: 63 KB
Network Data Support:	**Keypad Descriptions:**	**Physical Descriptions:**	**Keypad Descriptions:**
CSD GPRS	3 Labeled Soft Keys 5-way Scrolling Grid Key Mat	Dimensions: 105 x 44 x 19 mm Weight: 92 g	2 Labeled Soft Keys 4-way Scrolling Grid Key Mat Qwerty Key Mat
Extra Features:	**Network Data Support:**	**Memory:**	**Network Data Support:**
Wallet	CSD EGPRS GPRS HSCSD	Heap size: 195 KB Shared Memory for Storage: 4 MB Max JAR Size: 63 KB	GPRS HSCSD
	PC Connectivity:	**Keypad Descriptions:**	**PC Connectivity:**
	Bluetooth Infrared USB	2 Labeled Soft Keys 4-way Scrolling Grid Key Mat	Infrared USB
	Extra Features:	**Network Data Support:**	**Extra Features:**
	Handsfree Speaker **MP3 Ringtones** **PoC (Push-to-talk over Cellular)** **Presence** **Stereo FM Radio** **SyncML** **Themes**	GPRS HSCSD	**Fold-out QWERTY Keyboard** **IMAP4 email** **POP3 email** **SMTP email** **Stereo FM Radio** **SyncML** **Wallet**
		PC Connectivity:	
		Infrared USB	
		Extra Features:	
		CIF camera **Handsfree Speaker** **Stereo FM Radio** **SyncML** **Wallet**	

Sample Series 60 Phone Specification Comparison

Nokia N71 Technical Specs
Operating System:
Symbian OS v9.1
Developer Platform:
S60 3rd Edition
Java Technology:
CLDC 1.1
Mobile Media API (JSR-135)
Web Services API (JSR-172)
Security and Trust Services API (JSR-177)
Location API (JSR-179)
SIP API (JSR-180)
Mobile 3D Graphics API (JSR-184)
JTWI (JSR-185)
Wireless Messaging API (JSR-205)
FileConnection and PIM API (JSR-75)
Bluetooth API (JSR-82)
MIDP 2.0
Nokia UI API
Browser:
HTML over TCP/IP
WAP 2.0
XHTML over TCP/IP
Messaging:
MMS+SMIL
SMS
Digital Rights Management:
OMA DRM v1.0
Delivery Method:
HTTP Download
MMS
Sound Formats:
AAC
AAC+
AMR (NB-AMR)
eAAC+
M4A
MIDI Tones (poly 64)
MP3
MP4
RealAudio
SP-MIDI
Video/Audio Streaming
True Tones (WB-AMR)
WAV
WMA
Functionality:
GSM 1800
GSM 1900
GSM 900
WCDMA 2100
Regional Availability:
Africa
Asia-Pacific
China
Europe
Screen Display:
Color Depth: 18 bit
Color Depth: 16 bit
Resolution: 240 x 320
Resolution: 96 x 68
Physical Descriptions:
Dimensions: 98 x 51 x 23 mm
Weight: 139 g
Memory:
Unlimited Heap size
Shared Memory for Storage: 10 MB
Unlimited Jar size
Keypad Descriptions:
2 Labeled Soft Keys
5-way Scrolling
Grid Key Mat
Video Support:
3GPP formats (H.263)
MPEG-4
RealVideo
Network Data Support:
CSD
EGPRS
GPRS
HSCSD
WCDMA
PC Connectivity:
Bluetooth
Infrared
USB
Extra Features:
2 megapixel camera with digital zoom, IMAP4 email
Mini SD, MP3/AAC player
POP3 email, Push-to-Talk
SMTP email, SyncML
Themes, VGA camera w/ 2x digital zoom
Video call
Video player
Video recorder
Video sharing
Visual Radio
Word/Excel/Powerpoint Compatibility

Nokia N80 Technical Specs
Operating System:
Symbian OS v9.1
Developer Platform:
S60 3rd Edition
Java Technology:
CLDC 1.1
Mobile Media API (JSR-135)
Web Services API (JSR-172)
Security and Trust Services API (JSR-177)
Location API (JSR-179)
SIP API (JSR-180)
Mobile 3D Graphics API (JSR-184)
JTWI (JSR-185)
Wireless Messaging API (JSR-205)
FileConnection and PIM API (JSR-75)
Bluetooth API (JSR-82)
MIDP 2.0
Nokia UI API
Browser:
HTML 4.0 (XHTML 1.1)
WAP 2.0
XHTML over TCP/IP
Messaging:
MMS+SMIL
SMS
Digital Rights Management:
OMA DRM v1.0
Delivery Method:
HTTP Download
MMS
Sound Formats:
AAC
AAC+
AMR (NB-AMR)
eAAC+
MIDI Tones (poly 48)
MP3
RealAudio
SP-MIDI
Video/Audio Streaming
True Tones (WB-AMR)
WAV
WMA
Functionality:
GSM 1800
GSM 1900
GSM 850
GSM 900
WCDMA 1900
WCDMA 2100
Regional Availability:
Africa
Americas
Asia-Pacific
China
Europe
Screen Display:
Color Depth: 18 bit
Resolution: 352 x 416
Physical Descriptions:
Dimensions: 95 x 50 x 23 mm
Weight: 134 g
Memory:
Unlimited Heap size
Shared Memory for Storage: 60 MB
Unlimited Jar size
Keypad Descriptions:
2 Labeled Soft Keys
5-way Scrolling
Grid Key Mat
Video Support:
3GPP formats (H.263)
MPEG-4
RealVideo
Network Data Support:
CSD
EGPRS
GPRS
HSCSD
WCDMA
PC Connectivity:
Bluetooth
Infrared
USB
WLAN
Extra Features:
Instant Messaging
IMAP4 email, Mini SD
POP3 email, Presence
Push-to-Talk, RealOne player
SMTP email, Stereo FM Radio
SyncML, Themes
Video call
Video player
Video recorder
Video sharing
Visual Radio
Word/Excel/Powerpoint Compatibility

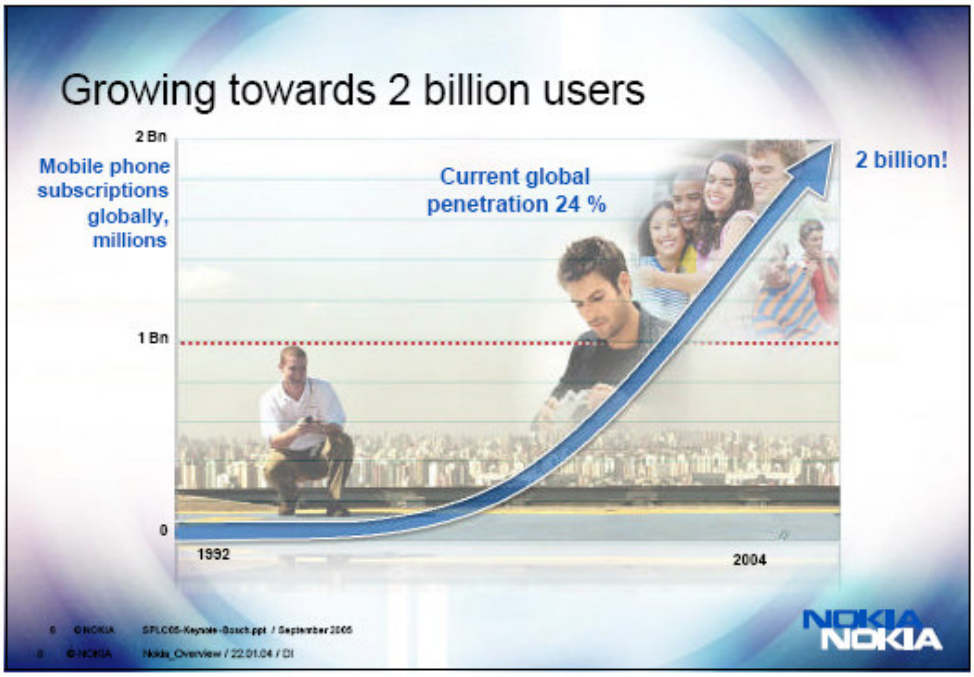
Nokia N92 Technical Specs
Operating System:
Symbian OS v9.1
Developer Platform:
S60 3rd Edition
Java Technology:
CLDC 1.1
Mobile Media API (JSR-135)
Web Services API (JSR-172)
Security and Trust Services API (JSR-177)
Location API (JSR-179)
SIP API (JSR-180)
Mobile 3D Graphics API (JSR-184)
JTWI (JSR-185)
Wireless Messaging API (JSR-205)
FileConnection and PIM API (JSR-75)
Bluetooth API (JSR-82)
MIDP 2.0
Nokia UI API
Browser:
HTML 4.0 (XHTML 1.1)
WAP 2.0
XHTML over TCP/IP
Messaging:
MMS+SMIL
SMS
Digital Rights Management:
OMA DRM v1.0
Delivery Method:
HTTP Download
MMS
Sound Formats:
AAC
AAC+
AMR (NB-AMR)
eAAC+
M4A
MIDI Tones (poly 64)
MP3
RealAudio
SP-MIDI
Video/Audio Streaming
True Tones (WB-AMR)
WAV
WMA
Functionality:
GSM 1800
GSM 1900
GSM 900
WCDMA 2100
Regional Availability:
Africa
Asia-Pacific
Europe
Screen Display:
Color Depth: 24 bit
Color Depth: 16 bit
Resolution: 240 x 320
Resolution: 128 x 36
Physical Descriptions:
Dimensions: 107 x 58 x 24 mm
Weight: 191 g
Memory:
Unlimited Heap size
Unlimited Jar size
Keypad Descriptions:
2 Labeled Soft Keys
5-way Scrolling
Grid Key Mat
Video Support:
3GPP formats (H.263)
MPEG-4
RealVideo
Network Data Support:
CSD
EGPRS
GPRS
HSCSD
WCDMA
PC Connectivity:
Bluetooth
Infrared
USB
WLAN
Extra Features:
2 megapixel camera with digital zoom, CIF camera
Handsfree Speaker
Instant Messaging
IMAP4 email
POP3 email, Presence
Push-to-Talk, SMTP email
Still Image Editor
SyncML
Themes
Video call
Video player
Video recorder
Video sharing
Visual Radio

Nokia N-Gage Technical Specs
Operating System:
Symbian OS v6.1
Developer Platform:
Series 60 Developer Platform 1.0
Java Technology:
CLDC 1.0
Wireless Messaging API (JSR-120)
Mobile Media API (JSR-135)
MIDP 1.0
Nokia UI API
Browser:
WAP 1.2.1
XHTML
Messaging:
MMS
SMS
UAProfile:
Profile 1
Digital Rights Management:
OMA DRM Forward Lock
Delivery Method:
WAP Download
Sound Formats:
AAC
AMR (NB-AMR)
MIDI Tones (poly 24)
MP3
WAV
Functionality:
GSM 1800/1900
GSM 900
Regional Availability:
Africa
Americas
Asia-Pacific
Europe
Screen Display:
Color Depth: 12 bit
Resolution: 176 x 208
Physical Descriptions:
Dimensions: 134 x 70 x 20 mm
Weight: 137 g
Memory:
Unlimited Heap size
Shared Memory for Storage: 3 MB
Unlimited Jar size
Keypad Descriptions:
2 Labeled Soft Keys
5-way Scrolling
8-way Scrolling in Game
Grid Key Mat
Video Support:
3GPP formats (H.263)
Network Data Support:
GPRS
HSCSD
PC Connectivity:
Bluetooth
USB
Extra Features:
Handsfree Speaker
IMAP4 email
MP3/AAC player
MP3/AAC recorder
POP3 email
SMTP email
Stereo FM Radio
Video player

Sample Series 80 Phone Comparison:

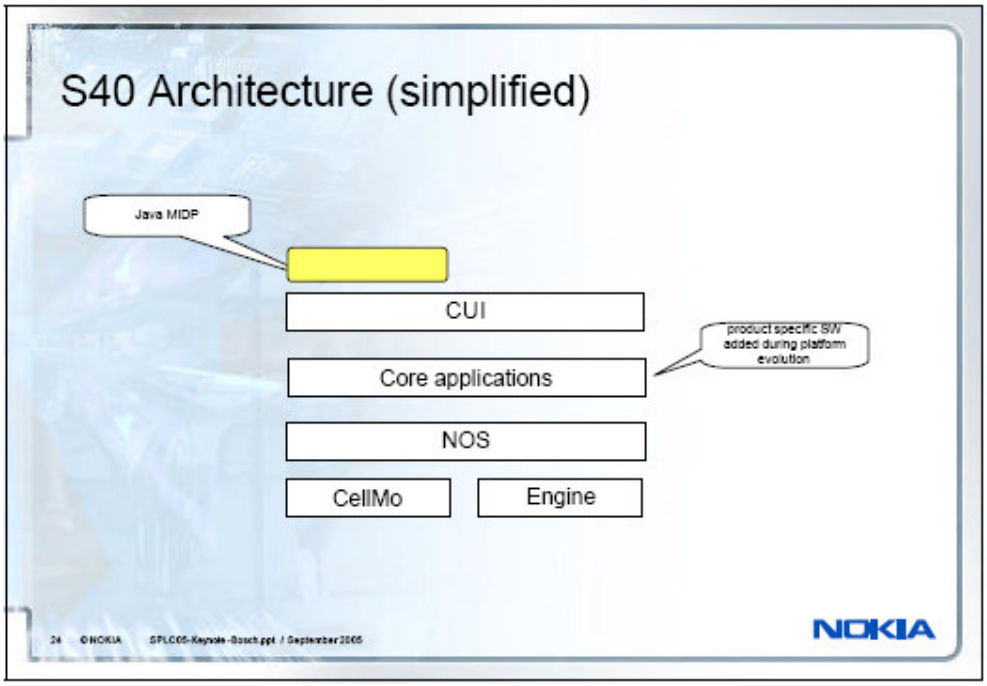
Nokia 9500 Technical Specs	Nokia 9290 Technical Specs	Nokia 9300 Technical Specs	Nokia 9210c Technical Specs
Operating System: Symbian OS v7.0s	**Operating System:** Symbian OS v6.0	**Operating System:** Symbian OS v7.0s	**Operating System:** Symbian OS v6.0
Developer Platform: Series 80 Developer Platform 2.0	**Java Technology:** JavaPhone API Personal Java	**Developer Platform:** Series 80 2nd Edition	**Java Technology:** JavaPhone API Personal Java
Java Technology: CLDC 1.1 Wireless Messaging API (JSR-120) Mobile Media API (JSR-135) CDC (JSR-36) Foundation Profile (JSR-46) Personal Profile (JSR-62) FileConnection and PIM API (JSR-75) Bluetooth API (JSR-82 No OBEX) MIDP 2.0 Nokia UI API	**Browser:** HTML 4.0 (XHTML 1.1) WAP 1.1	**Java Technology:** CLDC 1.1 Wireless Messaging API (JSR-120) Mobile Media API (JSR-135) CDC (JSR-36) Foundation Profile (JSR-46) Personal Profile (JSR-62) FileConnection and PIM API (JSR-75) Bluetooth API (JSR-82 No OBEX) MIDP 2.0 Nokia UI API	**Browser:** HTML 3.2 WAP 1.1
Browser: HTML over TCP/IP XHTML over TCP/IP	**Messaging:** SMS	**Browser:** HTML over TCP/IP XHTML over TCP/IP	**Messaging:** SMS
Messaging: MMS+SMIL SMS	**Sound Formats:** Video/Audio Streaming WAV	**Messaging:** MMS+SMIL SMS	**Sound Formats:** Video/Audio Streaming WAV
UAProfile: Profile 1	**Functionality:** GSM 1900	**UAProfile:** Profile 1	**Functionality:** GSM 1800 GSM 900
Digital Rights Management: OMA DRM Forward Lock	**Regional Availability:** Americas	**Digital Rights Management:** OMA DRM Forward Lock	**Regional Availability:** China
Delivery Method: HTTP Download MMS	**Screen Display:** Color Depth: 12 bit Resolution: 640 x 200	**Delivery Method:** MMS WAP Download	**Screen Display:** Color Depth: 12 bit Resolution: 640 x 200
Sound Formats: AAC AMR (NB-AMR) MIDI Tones (poly 24) MP3 WAV	**Physical Descriptions:** Dimensions: 158 x 56 x 27 mm Weight: 244 g	**Sound Formats:** AAC AMR (NB-AMR) MIDI Tones (poly 24) MP3 WAV	**Physical Descriptions:** Dimensions: 158 x 56 x 27 mm Weight: 244 g
Functionality: GSM 1800 GSM 1900 GSM 850/900	**Memory:** Unlimited Heap size Shared Memory for Storage: 2 MB Unlimited Jar size	**Functionality:** GSM 1800 GSM 1900 GSM 850/900	**Memory:** Unlimited Heap size Shared Memory for Storage: 2 MB Unlimited Jar size
Regional Availability: Africa Americas Asia-Pacific Europe	**Keypad Descriptions:** 2 Labeled Soft Keys 2-way Scrolling Grid Key Mat	**Regional Availability:** Africa Americas Asia-Pacific Europe	**Keypad Descriptions:** 2 Labeled Soft Keys 2-way Scrolling Grid Key Mat
Screen Display: Color Depth: 16 bit Resolution: 640 x 200 Resolution: 128 x 128	**Network Data Support:** HSCSD	**Screen Display:** Color Depth: 16 bit Resolution: 640 x 200 Resolution: 128 x 128	**Network Data Support:** HSCSD
Physical Descriptions: Dimensions: 148 x 56 x 23.6 mm Weight: 222 g	**PC Connectivity:** Infrared Serial Cable	**Physical Descriptions:** Dimensions: 132.0 x 51.0 x 21.0 mm Weight: 167.0 g	**PC Connectivity:** Infrared Serial Cable
Memory: Unlimited Heap size Shared Memory for Storage: 76 MB Unlimited Jar size	**Extra Features:** Fold-out QWERTY Keyboard Handsfree Speaker IMAP4 email POP3 email SMTP email SyncML	**Memory:** Unlimited Heap size Shared Memory for Storage: 80 MB Unlimited Jar size	**Extra Features:** Fold-out QWERTY Keyboard IMAP4 email POP3 email SMTP email SyncML
Keypad Descriptions: 3 Labeled Soft Keys 5-way Scrolling Grid Key Mat Qwerty Key Mat		**Keypad Descriptions:** 3 Labeled Soft Keys 5-way Scrolling Grid Key Mat Qwerty Key Mat	
Video Support: 3GPP formats (H.263) MPEG-4 RealVideo		**Video Support:** 3GPP formats (H.263) MPEG-4 RealVideo	
Network Data Support: CSD EGPRS GPRS HSCSD		**Network Data Support:** CSD EGPRS GPRS HSCSD	
PC Connectivity: Bluetooth Infrared USB WLAN		**PC Connectivity:** Bluetooth Infrared USB	
Extra Features: Handsfree Speaker IMAP4 email MMC POP3 email SMTP email SyncML VGA Camera WLAN (802.11b) Word/Excel/Powerpoint Compatibility **Compatibility**		**Extra Features:** Handsfree Speaker IMAP4 email MMC POP3 email SMTP email SyncML Word/Excel/Powerpoint Compatibility	

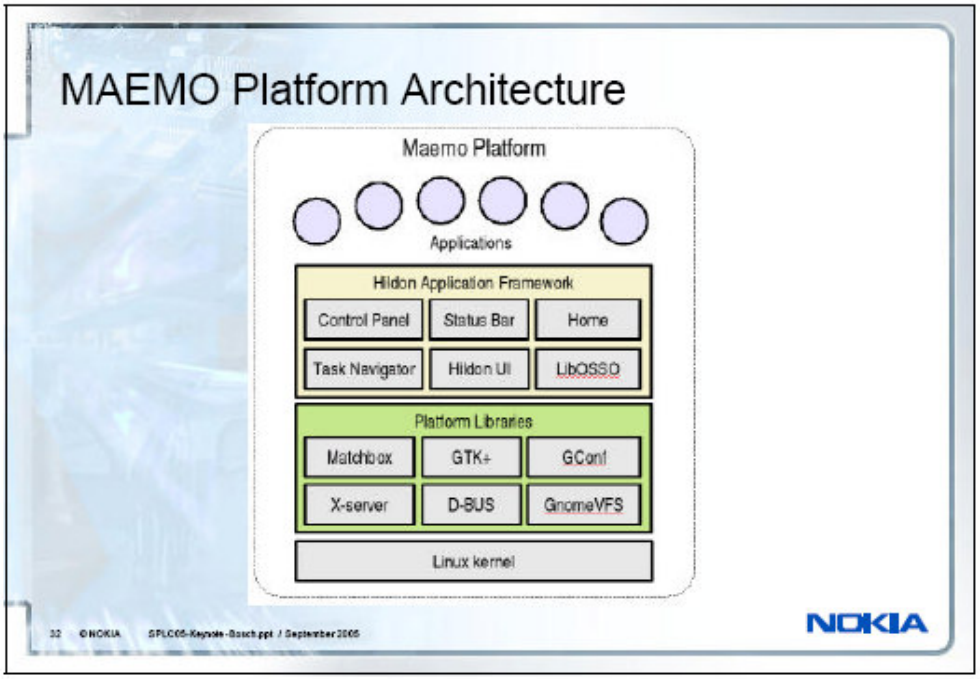
Appendix E - Nokia's Global Market Share



Bosch, J. *Software product families at nokia*. in *SPLC05*. 2005.

Appendix F -S40 Simplified Architecture and Open source architecture





Bosch, J. *Software product families at nokia*. in *SPLC05*. 2005.

Appendix G - Platform Versioning

	Series 40 platform			S60 platform			Series 80 platform
	1st Ed	2nd Ed	3rd Ed	1st Ed	2nd Ed	3rd Ed	2nd Ed
Symbian C++							
SMS Messaging	-	-	-	X	X	X	X
Web Services APIs (including XML and SOAP)	-	-	-	-	-	X	X
Socket connections	-	-	-	X	X	X	X
SSL 3.0 / TLS 1.0	-	-	-	-	X	X	X
HTTP version	-	-	-	-	1.1	1.1	1.1
IPsec and VPN client	-	-	-	-	X (FP2	X	X
SIP API	-	-	-	X (a	X (a	X	-
FTP API	-	-	-	-	X (b	-	-
Java™ 2 Platform, Micro Edition							
CLDC version	1.0	1.1	1.1	1.0	1.1 (FP2	1.1	1.1
CDC 1.0 and Personal Profile	-	-	-	-	-	-	X
MIDP version	1.0	2.0	2.0	1.0	2.0	2.0	2.0
Wireless Messaging API version	1.0 (c	1.1	1.1	1.0	1.1	2.0	1.1
Socket connections	-	X	X	-	X	X	X
SSL 3.0 / TLS 1.0	-	X	X	-	X	X	X
HTTP version	1.1	1.1	1.1	1.1	1.1	1.1	1.1
Web Services API (including XML and SOAP)	-	-	-	-	X (FP3	X	X (d
SIP API	-	-	-	-	-	X	-

Source: Forum, N., *Platforms*. 2006, *www.forum.nokia.com/main/0,6566,010,00.html*

Appendix H – Porting Applications

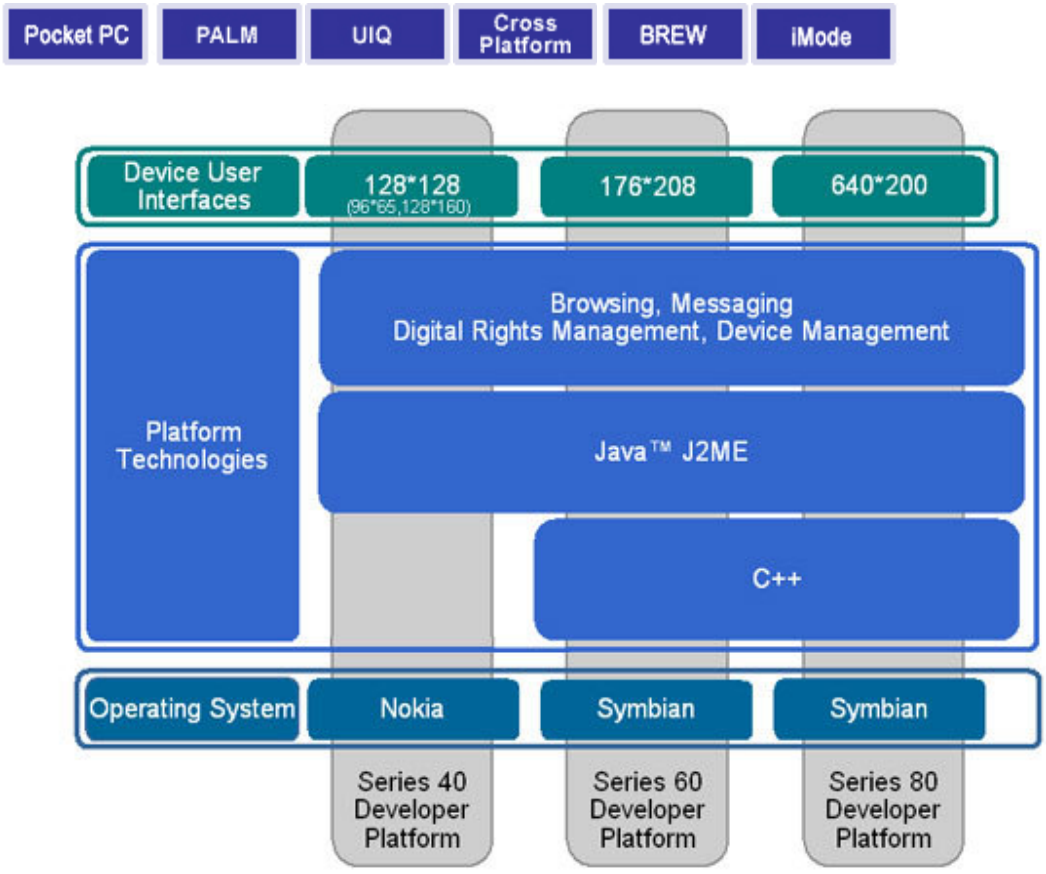
- Porting Overview
- Series 40 Platform
- Series 60 Platform
- Series 80 Platform

Overview

Platform devices have experienced widespread success in the global marketplace, resulting in a lucrative opportunity for developers. Forum Nokia wants to help developers port their mobile applications and services to Nokia's leading platforms.

Once developers have ported an application to one Platform series, it's easy and straightforward to migrate that application to another Platform series. This is because of the common application environment supported by the Series 60 Platform and the Series 80 Platform, which support native Symbian OS C++ and J2ME™.

Optimize your J2ME application to the Series 40 Platform to reach a mass-market of tens of millions of users.



Source: Forum, N., *Porting*. 2006, Nokia Forum: http://www.forum.nokia.com/porting

Appendix I- OMA Architecture Principles - Principle #4 (on adaptability)

Provide for Service adaptability based on device capabilities, network characteristics and user preference

OMA Architecture and Specifications define technologies that are used over broad range of mobile devices and networks. Devices range from basic data-enabled phones to smart phones, PDAs and fully capable laptops. Devices may have sophisticated multimodal user input and display capabilities, limited input and small screens or no user input and display at all (e.g. telemetry, sensors, automotive applications). Networks vary in bandwidth available to users and some may provide intermittent connectivity and increased latencies while others provide high reliability and low latencies. This is only a small sample of the difference in capabilities among the devices that will use OMA Technologies. In addition, user preference also can impact how content is selected or presented to the user.
OMA Architecture and Specifications must have the following characteristics.

- **Adaptability based on device and network capabilities**. OMA Architecture and Specifications must provide means for adapting or selecting content to meet the differing constraints and opportunities presented by the wide range of device and network capabilities.

- **Adaptability based on User preference**. OMA Architecture and Specifications must enable the adaptation or selection of content and services, including presentation, based on user preferences. Where it does not conflict with security or privacy, it should be possible for the adaptation to be performed by any entity in the service delivery chain if permitted by the user and content provider

- **Adaptability across wide range of devices and optimization**. User experience must be maximized across the range of possible devices and networks. It is not acceptable for OMA Architecture and Specifications to take a lowest common denominator approach that prevents use of advanced features in more capable devices and networks. At the same time, users of less capable devices should enjoy the best possible experience that those devices can provide. OMA Architecture and Specifications should consider optimisations for efficient use of, and maximizing the user experience with limited devices, while ensuring users of more capable devices benefit fully from the greater capabilities those devices offer.

- **Identifying device capabilities**. Preference should be given to technologies that permit Services and applications to adapt to different device capabilities and constraints, based on a description of the device's capabilities or network performance. Adaptive protocols that vary over time with transient conditions should also be considered.

- **Extensibility of device and network characterization and adaptation**. Applications, Services and User Equipment and Network Elements undergo continuous improvement. Methods of characterization and adaptation must be easily extensible to be able to apply to new, emerging capabilities.