

SPL development of mobile applications

Paolo Di Francesco

Gran Sasso Science Institute (GSSI), L'Aquila, Italy
paolo.difrancesco@gssi.infn.it



Roadmap

1. Introduction
2. Software Product Line methodologies
3. Modeling features and variability
4. Context, context aware computing, context-aware design
5. SPL in context-aware mobile applications
6. Context management architecture in mobile applications
7. Conclusions

Introduction

Product Line

A set of products that share common characteristics but also share a substantial number of differences

Problem

- Need to deliver software faster
- Meet the needs of individual customers
- Increasing demand in number of products

Software Product Lines goals

1. Reuse as much as possible during all stages of software life cycle
2. Manage evolution and growing complexity of products
3. Avoid organizing related products as independent units

SPL methodologies

Product line design

Products are no longer developed independently from one another, but instead only a single, but variable product is developed **product line infrastructure**

Product instances are derived, by configuration, from the product line infrastructure

Two different phases

1. Domain engineering: development and evolution of the product line infrastructure
2. Product engineering: definition of individual product instances to be derived from the infrastructure

SPL methodologies

Software mass customization

1. Different variants of the same product should be created through automated composition and configuration
2. **SPL configurator** should be developed to automate the process of composition
3. Upfront cost: assets must be created for reuse

Minimally invasive transitions

1. Little initial investment
2. Reuse existing artifact as much as possible and make just enough refactoring to allow for composition and configuration
3. Adopt incremental strategies

Bounded combinatorics

Manage the complexity of the number of variants to preserve the product line scalability

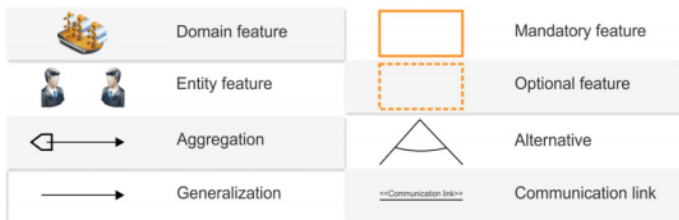
Modeling features in SPL

Features modeling

Activity of identifying externally visible characteristics of products in a product line and organizing them into a feature model

Features can be services, operations, nonfunctional characteristics and technologies of a particular product line

- Common features are modeled as mandatory
- Different features are modeled as optional
- Feature binding units identify related set of features



Modeling variabilities in SPL

Feature model should be as flexible as possible in order to design variability of products

Variabilities design

Activity of constructing and evolving a reuse infrastructure, which realizes all variability requirements specified in a feature model

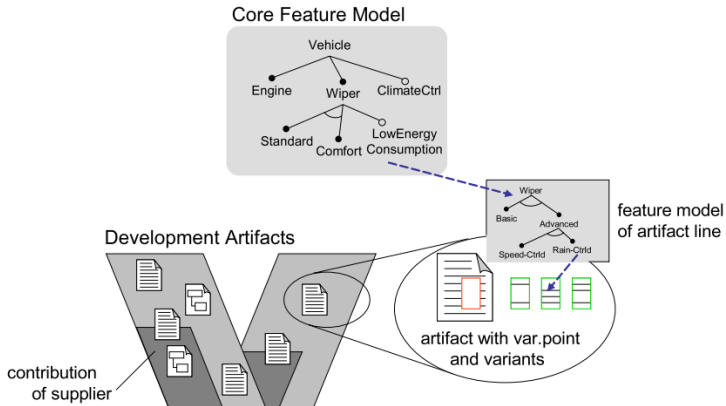
Composition of small SPL

Each development artifact may be organized as its own small product line^a

1. Artifacts
2. **Artifact lines**, small-sized product line: own feature model
3. Several artifacts may be combined and managed together as a single small-sized product line

^aM.-O. Reiser, R.T. Kolagari, M. Weber, Unified feature modeling as a basis for managing complex system families

Composition of small SPL



Artifacts in the artifact lines become independent from the global core feature model of the overall product line

Context

Context can be used to enhance capabilities of devices and software, and to adapt the behavior of an application to different situations

Context

Any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application ^a

^aA.K. Dey, Understanding and using context

Context-awareness

A system that uses context to provide relevant information and/or services to the user is said to be context-aware

Context-aware applications

Context-aware development

Developing context-aware application is a difficult task

- Architectural support is limited
- Developers must access raw data
- Every application must implement context logic

Context-aware architecture design

- Context Toolkit
- Moblinc architecture
- ContextDroid architecture

SPL for context-aware mobile application

Context Toolkit ¹ architecture is capable of extending context-aware capabilities to non context-aware applications

Context Toolkit (abstractions)

- Context widget: sense the proper context for a user's device, and make this information available to the applications
- Context interpreter: interprets the current context for the user considering one or more contexts and the historical information
- Context aggregators: gather context for specific entities, behaving as proxy to context for applications

¹A.K. Dey, Understanding and using context

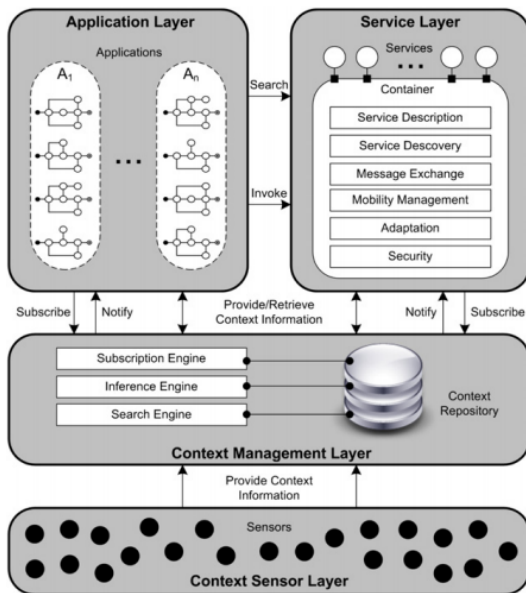
SPL for context-aware mobile application

To achieve an high level of reuse Mobiline SPL ² defines three different phases

1. Identify commonalities and variabilities of the domain and build a feature model
2. Use the feature model to perform requirements elicitation on a family of applications for a specific domain
3. Uses common assets available to the product line to create products (Application engineering)

²F. G. Marinho et al. MobiLine: A Nested Software Product Line for the domain of mobile and context aware applications

SPL for context-aware mobile application



Context management in mobile application

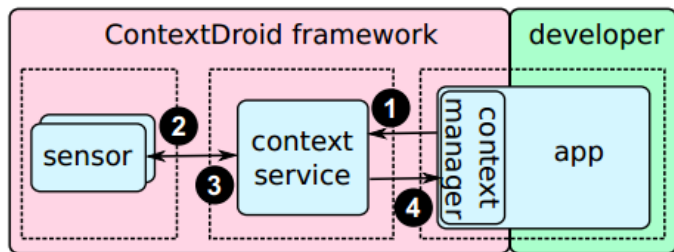
Contextdroid ³ is a framework which allow applications to register context expressions for context entities

ContextDroid framework

1. Context entity: collection of information and the current state of that entity is determined by context sensors
2. Context sensors gather information from a particular source and deliver it as a context entity to ContextDroid
3. The state of a context entity during a certain period in time is a context entity reading, and context expression can be used by programmers to easily define specific conditions.
4. All the information will be then managed in a centralized environment in order to perform better and minimize resource usage.

³R.K.T.K.H.B. Bart van Wissen, Nicholas Palmer, Contextdroid: an expression-based context framework for android

SPL for context-aware mobile application



- Support for context management
- Optimizes resource usage avoiding the problem of power consumption for same data request
- Avoids duplicating analysis of sensors data

Conclusions

Architecture definitions can be used as basis for products instantiation for context aware applications, and, in conjunction with feature models, usage of SPL can help to increase effectiveness and control over variability complexity

SPL are effective to reason about the mobile application domain, and its artifacts can be composed to analyse bigger scenarios

Advantages

1. High level of reuse
2. Composition of artifacts
3. Reducing effort and costs
4. Reduce complexity

Disadvantages

1. Domain engineering requires substantial effort
2. Initial investment







Conclusions

Mobile application domains are extensive, and they are limited by the resource consumption of devices

Context-awareness design

1. Currently, application access raw data, implement logic and analyze data
2. Context-aware architectures are similar one another
3. ContextDroid is a framework that will optimize resource usage and management of context-awareness

General SPL can be adapted to the mobile applications domain, and to context-aware features, choosing different approaches according to the existence of artifacts

-  A.K. Dey, Understanding and using context, *Personal Ubiquitous Comput.* 5 (2001) 4 – 7.
-  C.W. Krueger, New methods in software product line practice, *Commun. ACM* 49 (2006) 37 — 40.
-  J. Lee, D. Muthig, Feature-oriented variability management in product line engineering, *Commun. ACM* 49 (2006) 55 -- 59.
-  R.K.T.K.H.B. Bart van Wissen, Nicholas Palmer, Contextdroid: an expression-based context framework for android, in: *Proceedings of the International Workshop on Sensing for App Phones*.
-  M.-O. Reiser, R.T. Kolagari, M. Weber, Unified feature modeling as a basis for managing complex system families, in: *VaMoS*, pp. 79 -- 86.
-  F. G. Marinho et al. MobiLine: A Nested Software Product Line for the domain of mobile and contextaware applications, *Science of Computer Programming*, Available online 18 May 2012, ISSN 0167-6423, 10.1016/j.scico.2012.04.009.