Visualising Variability Relationships in Software Product Lines

Ciarán Cawley, Steffen Thiel, Patrick Healy
Lero, University of Limerick
Limerick, Ireland
{ ciaran.cawley | steffen.thiel | patrick.healy }@lero.ie

Abstract
Software Product Line Engineering is a development paradigm that focuses on the identification and management of the commonalities and variability of a set of software products such that core assets can be developed and (re)used to derive individual product variants with a minimum of cost [3]. In industrial product lines where it is possible to have thousands of variation points, the scale of variability can become extremely difficult to manage. In this position paper we elaborate on our ideas of focussing the representation and visualisation on the variability relationships that exist between different product line elements such as decisions, features and components and not on those elements that they relate. Further, we provide a conceptual three-dimensional visualisation technique to manage these relationships in the context of specific stakeholder tasks.

1. Introduction
Software Product Line Engineering is a development paradigm that focuses on the identification and management of the commonalities and variability of a set of software products such that core assets can be developed and (re)used to derive individual product variants with a minimum of cost [3]. In industrial product lines where it is possible to have thousands of variation points, the scale of variability can become extremely difficult and costly to manage [4], [5]. How this variability is managed is key to the success of the product line. Decision and feature models are widely used to model a product line but when considering inter-feature relationships and relationships between features and other elements such as requirement decisions and system components, their representation can become highly complex [6].

Visualisation has the ability to amplify human cognition of large and complex data sets [7]. In this position paper we elaborate on our ideas of focussing visualisation on the relationships that exist between the software product line elements and not on the elements that they relate. Section 2 discusses the software product line engineering tasks and ideas that motivate this approach. In section 3 we provide a conceptual three-dimensional visualisation technique to manage these relationships in the context of specific stakeholder tasks. Sections 4 and 5 present related and future work respectively while section 6 concludes the paper.

2. Variability Relationships
Product Derivation is an application engineering process whereby a product variant is derived from core software product line assets. This can be achieved through navigation of a decision and/or feature tree where requirement decisions can be made and features can be selected or eliminated from the product being derived. In some implementations, relationships such as mutually excludes and requires are automatically resolved during this process. This type of semi-automatic product derivation aids the stakeholder and addresses some of the main challenges with large software product lines. This kind of variability management provides a helpful structured and hierarchical approach such as in Figure 1 and this proposal aims to build on this support to further enhance the efficiency of the product derivation process in large software product lines.

In this position paper we concentrate on addressing two main issues; supporting the process of semi-automatic product derivation and allowing the stakeholder to understand the consequences of actions (specifically decision making). This would be achieved while maintaining a context of the product as a whole. It has been shown that the unforeseen consequences of particular decisions can result in a large addition to the time and costs of deriving a new product variant.
This is one of the main factors affecting the product derivation process [4]. The relationships that exist between the various SPL elements are the key aspect that needs to be understood by stakeholders in order to address this problem. With this in mind, we are applying the focus of the visualisation on the relationships themselves in an attempt to address the problem directly.

Initial focus is concentrating on visualising the relationships between different elements such as Decision is implemented by Feature and Feature is implemented by Component. Specific attention is also being devoted to the relationships that can exist between elements of the same type, specifically features, such as Feature requires Feature, Feature excludes Feature, Feature recommends Feature, Feature is problematic with Feature etc.

3. 3D Interactive Visualisation

Although there are differing reports on the effectiveness of 3 dimensional visualisations to support software engineering, there seems to be general acceptance that it can be very effective for specific purposes [8], [9]. Figure 2 illustrates two points, firstly, in this work by Lange et al [1], a three dimensional view of the collective UML diagrams of a software system are visualised in a way that suggests large amounts of information can be presented without overwhelming the viewer. Secondly, this visualisation gives a contextual overview and suggests exploration would again not overwhelm and which could also allow for significant focus without losing that context.

Figure 3 shows a conceptual diagram illustrating one possible interactive visualisation that could support our ideas as outlined above. A 3 dimensional scatter plot can be created by mapping primary model elements to the 3 dimensional X, Y and Z axes. With this, we can create a 3 dimensional derivation space where a point in that space represents two or more relationships between those elements, a Relation Point. If, for example, we mapped features, components and decisions to the X, Y and Z axes, the existence of a point in the generated 3D space would at least represent a Decision implemented by Feature relationship and a Feature implemented by Component relationship. We suggest that enhanced cognition when performing derivation and impact analysis tasks is afforded. E.g. a set of relation points would be presented which identify all features impacted by a particular decision choice in a way that does not overwhelm the stakeholder.

Subsection 3.1 elaborates on the conceptualisation presented in Figure 3 while the following subsections discuss the suggested benefits and how they aid product derivation and support enhanced cognition.

3.1. Visualisation of Relationships

Figure 3 presents a conceptual sketch using parts of an example automotive restraint system control unit product line which was presented in [2]. It is not intended as a technical example but to serve as an illustration.

As mentioned, this figure presents a 3 dimensional scatter plot created by mapping primary model elements to 3 dimensional axes. In this example Features, Components and Decisions are mapped to the X, Y and Z axes respectively. For initial simplicity, the mapping is a simple sequential listing. The figure represents the impacted relationships for the Decisions to include Hardware B and Hardware C (highlighted in upper case on the Z axis).
11 relation points are presented. A grey relation point indicates a Decision implemented by Feature relationship and a Feature implemented by Component relationship. With 6 grey relation points, 12 relationships in total are visualised. If a relation point is green it identifies an additional relationship which represents Feature requires Feature. If a relation point is red it identifies the additional relationship Feature excludes Feature. If a relation point is blue it identifies the additional relationship Feature recommends Feature. If a relation point is orange it identifies the additional relationship Feature problematic with Feature. In total, 27 relationships are visualised. As an illustrative example, one of the green relation points represents the following information: the CAN Bus Interface feature is required due to the selection of the Weight Sensing Feature; the CAN Bus Interface feature is implemented by the CAN Bus Interface component and the CAN Bus Interface feature implements part of the Hardware C decision.

Exact decision, feature and component descriptions and attributes are envisioned as details-on-demand. Stakeholder interaction through visualisation rotation, mouse clicks and contextual pop-ups can present pertinent information. These are explored further below.

With this relational space in place, a variety of interactive visualisation techniques applicable to 3 dimensions could be judiciously applied such as the world-in-hand navigation metaphor, distortion, elision, layering and multiple windows. The following subsections elaborate further on the use of such techniques in relation to the benefits and cognitive aid we have suggested.

3.2. Explicit and Implicit Transformation

The derivation space outlined could be most effective when synchronised with other views such as a de-
cision or feature tree. The relation points displayed would change depending on the specific actions of the stakeholder, e.g. selecting a specific decision option would alter the content and view of the derivation space. The derivation space would show all relationships affected by that decision (explicit) including other decisions (implicit). This would immediately convey to the stakeholder the number and nature of relationships impacted without confusion, in context and with the ability to focus on and query individual relationships.

3.3. Reactive & Manipulable

The use of animation and distortion which are well known cognition support techniques [7] would play an important role. Animating transitions within the derivation space and distorting uninteresting relationships, given a specific job (such as show all relation points for a specific feature), aids preservation of the context. The derivation space would allow rotation in a world-in-hand dynamic and individual relation points would be selectable and allow querying. These allow the stakeholder to view, explore, filter and choose details on demand through mouse clicks on relation points or axis labels. Information such as the direction of a relationship (one feature requires/excludes another feature) would be displayed in contextual pop-ups. A stakeholder could display information relating to a particular relation point such as the decision and/or feature description, the implementing component(s) and other related elements. These, together with colour encoding and iconography, combine to facilitate the stakeholder in understanding different types of relationships and identifying possible problem areas such as Mutual Problematic relationships (through colour-coding) or High Risk/Cost features (through iconography) etc.

3.4. Multiple Windows & Context

As briefly mentioned in 3.2, synchronising this visualisation with a decision view, feature tree and component view like those implemented in [2] and Figure 1 would allow a connecting context and in our opinion would aid the focus+context issues challenging large scale product derivation. This could be achieved through use of the derivation space as a means to manipulate the other views so that the derivation space acts as an overview context while other views behave as a focus. E.g. by selecting a decision, feature or component label or by selecting a relation point, the corresponding element or relationships could be highlighted in the corresponding element model visualisation. This would allow the stakeholder to focus on that element while preserving the derivation space view. It is envisioned that the product derivation process can proceed through the use of any of the visualisations while the derivation space provides constant feedback.

3.5. Stakeholder Tasks

We suggest the derivation space can provide a facility to base feature selection and elimination decisions on by providing clear and pertinent information in context.

In our opinion, an interactive visualisation like this supports a number of primary tasks that need careful consideration when executing application engineering with software product lines. Product derivation is supported through the employment of an alternative contextual view that supplies immediate feedback to the stakeholder concerning the nature of the impact of a particular action such as the number and type of relationships affected. It would allow an alternative method of decision making and feature selection/elimination within this contextual view. Through the use of colour encoding, iconography, layering and distortion, specific types of elements affected by an action could be highlighted in context without cluttering other views in use. As mentioned previously, this could be used to identify high risk or high cost features which may be problematic.

4. Related Work

FeaturePlugin [10], Gears [11], pure::variants [12] and COVAMOF [13] are examples of software product line tools that employ a visual component. These tools focus on representing the main elements such as decisions and features but unlike our approach do not use 3 dimensional visualisation techniques to address focus+context challenges. They also do not give the relationships between the various elements the centred focus of our approach.

VisMOOS [14] and MUDRIK [15] are software visualisation tools that employ a variety of 3 dimensional techniques to support cognition of object oriented software systems. These tools do not directly support software product line engineering but do provide interesting techniques such as a 3D class relationship matrix. Unlike our approach these tools focus on understanding and not process support. There is still work required in evaluating these tools to ascertain the effectiveness of the techniques.

Lange et al. in [1] provide another interesting 3D UML visualisation which illustrates the ability to present large amounts of information and to provide an overview context. Again, this visualisation does not directly support software product lines and does not provide any process support.
[8] and [9] are examples of 3D information visualisations where some evaluation of their effectiveness compared with 2D equivalents has been performed. Both papers suggest that in some situations there was no perceived benefit in having a 3D visualisation while in others there was a marked increase in task performance efficiency. This work serves to provide some evidence that 3D techniques can be effective in certain circumstances.

5. Future Work

It is planned to further review existing literature where 3D visualisations have been evaluated for their effectiveness particularly in comparison to 2D equivalents and particularly where 3D scatter plots have been utilised.

It is also planned to further specify the relationships and product derivation tasks that will initially and explicitly be targeted by the visualisation.

A specification detailing the features and functionality is then planned which would facilitate subsequent implementation of an initial prototype which could be used for early user studies.

6. Conclusion

This position paper presented an approach to addressing some of the challenges of product derivation in large scale software product lines. The proposed approach focuses on providing an interactive visualisation which centres on the representation of the relationships that exist between the various software product line elements. A three dimensional interactive technique to manage these relationships was proposed which we suggested could assist stakeholders in performing specific product derivation tasks. We believe that tasks such as decision making, feature selection/elimination, impact analysis and identification of high risk/cost relationships can be supported in a large scale software product line environment.

7. References


