Issues in Managing Variability of Medical Imaging Grid Services

Mathieu Acher[1], Philippe Collet[1] and Philippe Lahire[1]

[1]Université de Nice Sophia Antipolis, laboratoire I3S - CNRS UMR 6070
Polytech Nice - Sophia, 930 Route des Colles, BP 145, F-06903 Sophia Antipolis Cedex, France

Abstract

In medical image analysis, there exist multifold applications to grids and service-oriented architectures are more and more used to implement such imaging applications. In this context, workflow and service architects have to face an important variability problem related both to the functional description of services, and to the numerous quality of service (QoS) dimensions that are to be considered. In this paper, we analyze such variability issues and establish the requirements of a *service product line*, which objective is to facilitate variability handling in the image processing chain.

Contents

1 Introduction **95**

2 An Analysis of Variability in Medical Imaging **97**
2.1 Segmentation 97
2.2 Analysis of QoS Variability in Segmentation 97
2.3 Impact of Variability for Grid Services 98

3 Building the Service Product Line **99**
3.1 Software Product Lines 99
3.2 Principles 100
3.3 Open Issues 102

4 Conclusion **103**

1 Introduction

For several years now the clinical area has been investigating grid computing to deal with many problems related to large medical data sets manipulation, usually heavily fragmented, on very wide distributed infrastructures. In medical image analysis, there exist multifold applications to grids, from validation and optimization processes of specific algorithms to overall reduction of computing time. In the same time, image analysis tool pipelines are undergoing homogenization, strongly motivated by the need for mutualizing

software development and easily comparing results. A medical image processing library such as ITK or the statistical tool SPM are examples of these efforts. In both cases however, these approaches assume that codes are fully integrated and thus tightly coupled. Harnessing the full power of the grid implies to be able to deploy different variations of algorithms while being independent of the heterogeneity of existing codes. Moreover, these codes are often not limited to a single algorithm but are expressed as processing pipelines.

A service oriented architecture (SOA) is especially adapted to adress these requirements, with services inherently decoupled and abstracted from technical platforms, and workflows to design composed algorithms. It enables users to partially overcome the division of clinical centers and medical imaging laboratories. But building on SOA also implies to tackle two major problems in its usage on realistic use cases. First, code maintainers have to provide basic imaging services from heterogeneous codes, including detailed information enabling their composition to construct new pipelines and the control of their deployment on the grid. The second problem concerns the management of the numerous non functional properties that have to be exploited during deployment or run times, in order to ensure a quality of service (QoS) adapted to the user. These QoS properties expose different forms of variability as they may be related to a service itself (reliability, availability, cost, expected execution time...), to its provision on the grid (parallelism grain, data handling protocol, adaptability to resources...) or to some user needs (emergency of a computation, expected output quality...). The two problems put together lead to strong limitations in the application of SOA principles to the grid for medical image analysis. We identify these two problems as being related to an important variability in the service functionalities — there are similar basic services to choose from to build a complete workflow — and an even more important variability in the QoS related properties of the service — these properties can be about execution time, cost, quality of results, *etc* . —. This concept of variability is now an essential design elements in software engineering [2, 13]. Its realization can be seen as a way to describe the whole generality of an entity (a software artefact) through the specification of commonalities and differences. This approach allows software architects to i) describe the structure and the behaviour of a single entity, with the possible common parts that may exist between several similar entities; ii) propose variants of one entity and therefore specify possible variations within the structure and the functionalities that are provided; iii) define optional parts for both structural and behavioural aspects; iv) describe assembly and runtime constraints between entities (main types of constraints are mutual exclusion and dependancy constraints), and v) define behavioural variations that are implied by the specification of entity variants.

In our context, the management of variability concerns both the services and the resulting composed processes. Our long term goal is to provide a complete *software product line* [2] that describes major variations in functional and non functional properties of medical imaging services and workflows. A software product line applies the general industrial notion of engineering family of similar entities. The main focus is on ensuring or verifying appropriate properties on the instantiation of a single entity from the product line. As a start, our ongoing work is limited to a software product line handling only variability on the description of imaging services [7]. In this paper, we present the first analysis and specifications of such a software product line. We analyze the different aspects of variability in some medical imaging services, and we focus on relevant QoS properties, taking the segmentation services as a running example (Section 2). We then present the main principles of a software product line framework that would handle the identified variability points (Section 3). It notably has to enable service providers and workflow experts i) to capture the commonalities and the differences of legacy services, ii) to efficiently build the right service according to these commonalities and differences and iii) to use the line to select appropriate services according to functional and non functional criteria. We also describe how Model-Driven Engineering (MDE) techniques are used to build the service product line as a model of the manipulated services so that description, selection and compatibility checking can be made through the product line. Finally we discuss some identified open issues related to the implementation of the product line (Section 4).

2 An Analysis of Variability in Medical Imaging

Let us consider a general pipeline in medical imaging. For a same type of image processing there are most of the time several ways to build it and numerous services are available to the workflow experts for each step in the imaging process. Functional aspects supported by these services are highly variable: for instance, there are many algorithms for the segmentation stage that are designed for a specific image acquisition modality. Neverthless, the variability mostly concerns the *quality* offered by the service (QoS), which describes in which conditions functional aspects are provided. Indeed, a segmentation algorithm may perform very well for SPECT images when considering the accuracy criteria, while another one could be less accurate but much faster for the same input. These services may provide the same functionality, but optimize different QoS dimensions. It is thus not sufficient to only consider functional characteristics of services. We now analyze how and which QoS properties are computed and quantified in the medical image analysis area, considering one essential technique, namely segmentation.

2.1 Segmentation

The goal of segmentation is to select perceptual units of an image that correspond to the real anatomy of the patient, and which need to be measured or visualised by the clinical user. The process of segmentation is a crucial (and often preliminary) step for medical imaging analysis and diagnosis. Unfortunately, automatic segmentation is a problem without general solution, as segmentation results depend on many factors, such as modality acquisition, image noise level, organs / body region extracted, pathology, *etc*. Selecting an accurate and efficient segmentation technique can avoid or minimize inappropriate results. Consequently the need of a standard quality measure has been highlighted [20] in order to evaluate different qualities of segmentation algorithms.

Different evaluation methods have then been proposed and we identify them as a first degree of variability. First, the *analytical* methods directly consider principles and properties (such as requirements, utilities, complexity, *etc.*) of algorithms. Analytical methods have not received much attention, considering that they cannot obtain fine-grained properties, they only work in a particular context and they have difficulties to compare algorithms. Second, *goodness* methods compute image-specific properties of the segmented object such as intra-region uniformity, inter-region-contrast, entropy, shape, edge quality, *etc* . As pointed in [5], this approach implies the subjectivity of selected properties, but they do not require an explicit reference knowledge and they may give a fast evaluation in some cases. Finally, *discrepancy* methods rely on a gold standard. The availability of a reference segmentation, supposed to be an ideally segmented image, allows discrepancy to measure the agreement between a segmented object – the result of the segmentation process – and references. Similarity or difference measures between segmented and reference images are then computed.

2.2 Analysis of QoS Variability in Segmentation

We identify several variability points in QoS related to segmentation.

First evaluation methods have to specify the *application domain* under consideration, which according to [17] is determined by three entities: the goal of the segmentation (the task), the body region and the imaging protocol. For instance, a particular segmentation method may have high performance in determining the volume of a tumor in the brain on an MRI image, but may have a low performance in segmenting a cancerous mass from a mammography scan. It is thus necessary to introduce specific information about images and

anatomical structures users want to identify, and this can be expressed through several choices (alternatives).

This survey of evaluation methods show that the QoS is context dependent: the absence of a reference image prevents using discrepancy methods, whereas the knowledge of both clinical context and medical objectives can largely improve QoS measurements. Statistical (or empirical) analysis, which considers mostly discrepancy methods defined above, must propose common criteria (*metrics*) to compare segmentation algorithms. But these metrics do not make any sense if they are not computed in a particular context. This is clearly expressed through constraints between alternatives.

Another important aspect is that the quality of the evaluation itself has to be considered [20]. Some evaluation methods focus on specific measures: validation metrics in medicine, such as sensitivity and specificity, have drawbacks not only for medical image processing but also for evaluation of medical tests. Moreover, the metrics selected are subjective or objective and this is another variation point. Complexity involved for evaluation may be important if, for instance, a monitoring process is used to control algorithms. Evaluation requirements are also considered. All those parameters lead to describing and handling complex interdependencies.

In many research works, the main quality considered is *accuracy* (aka fidelity), which refers to the degree to which the segmentation results agree with the "true" segmentation. We claim that the influence of various parameters cannot be measured accross this single quality dimension. It is more relevant, as proposed in [15][17] specifically for segmentation algorithms, to consider many dimensions to give a meaningful answer to the performance of segmentation algorithms. This would help the expert specifying high level specification of QoS, more adapted to the context, and thus improve the selection of algorithms. For instance, if an algorithm cannot cope with certain kinds of inputs, then it should be well-documented, so that its *robustness* can be known before invoking the algorithm. In [17], the authors consider precision (reliability), accuracy (validity), and efficiency (viability) and describe a framework for evaluating image segmentation algorithms.

These research works are closely akin to common examples of validation criteria proposed in [11]. Finally, it should be noted that precision, accuracy, and efficiency factors have a complex interdependency: an attempt to increase accuracy may imply a decrease in efficiency and/or precision [15].

2.3 Impact of Variability for Grid Services

Managing QoS on the grid is a crucial issue as providing end-to-end qualities is one of the topmost user requirements [10]. QoS issues have not been addressed very well in most Grid workflow management systems while supporting QoS at both specification and execution level becomes increasingly critical [19][4]. In addition to the QoS related to the grid infrastructure, attention must be put on the QoS offered by the services themselves. QoS-aware workflow engines are able to ensure that each application meets its user requirements. To do so, five relevant QoS dimensions are generally considered: time, cost, fidelity, reliability and security [19][6]. These are high level concepts that should be refined into fine-grained QoS characteristics. Moreover, QoS grid concerns such as reliability, latency or security can be expressed through this classification. The analysis of variability in medical imaging has shown that such dimensions are also considered in the domain [11], and especially the fidelity dimension.

The QoS variability of medical images segmentation services impacts various operations in the management of workflows. For instance, for a given task of the workflow, the most adapted service can be selected considering QoS attributes of the service and the clinician requirements. Our service product line must facilitate QoS management in relation with workflow engines, so that:

- Some resulting QoS of an application can be computed *before* making the service available to customers. Application of statistical methods are likely to be conducted in this case.
- The selection (statically or dynamically) of services through their QoS can be made by the workflow engine. Scheduling of computational tasks on the grid is a complex optimization problem which may require different criteria to be considered [18]. Few research works propose a scheduling approach for multiple criteria on QoS parameters, and they mainly focus on QoS aspects of the grid infrastructure itself [3]. Our aim is thus to manage QoS on medical imaging characteristics and on properties of the grid at the same time. Just as segmentation algorithms can be provided with variable QoS attributes and deployed on the grid, the service product line should enable one to use such a scheduling approach.
- The appropriate QoS monitoring is implemented. Monitoring systems can control the fulfillment of QoS criteria and moreover offer error detection and recovery. For instance, discrepancy methods can dynamically check the process results and an erroneous segmentation process should be detected early. Consequently, the service product line should handle fine enough information on QoS so that monitoring constraints can be described and reused.
- Adaptation strategies can be implemented as well. In response to unexpected behaviour – detected by the above monitoring system – or technical conditions (delay, latency), it is necessary to adapt or reschedule a workflow, considering a set of alternative services. The service product line should be also usable in this context.

3 Building the Service Product Line

In this section we first present the concept of software product lines, which objective is to handle the identified variability points. We specify the architecture of intended *service product line* and discuss open issues regarding its implementation and integration.

3.1 Software Product Lines

Manufacturers have long employed analogous engineering techniques to create a product line of similar products using a common factory that assembles and configures parts designed to be reused across the product line. For example, automotive manufacturers can create tens of thousands of unique variations of one car model using a single pool of carefully designed parts and one factory specifically designed to configure and assemble those parts. Similarly, software product lines (SPL) refers to engineering methods, tools and techniques for creating a collection of similar software systems from a shared set of software assets using a common means of production. A SPL must support the concepts of variability in order to describe not only one specific entity but a family of entities. This means to be able to choose en entity between several possible variants and to select optional parts. This process is called the derivation process. If we compare it to the concept of generic class in object-oriented languages, the description of one family of entities corresponds to the description of a generic class and the derivation process corresponds to the instantiation of the generic class. Like the instantiation process needs to check for the type compatibility of generic parameters, the derivation process needs to check that the constraints that exist between several optional parts or variants (either mutual exclusion or dependancy) are verified by the user choices. At the code level, conditional compiling and aspect-oriented programming have been studied to manage variability at implementation and compile time [1]. More recently, variability management has gained attention in the earlier steps of the software development [21]. In [14] we have proposed to introduce variability into an

Aspect-Oriented Modeling (AOM) approach and use it to design SPL. Besides this approach is now getting more attention in the SOA domain [7].

Therefore, we want to provide to software architects capabilities to manipulate an imaging service not only as one service but as a service product line, which allows them to build easily derived services that include the functionalities and the QoS properties matching their requirements. Considering the mentioned requirements, our purpose is to tackle the determined variability issues by including services within a product line architecture. In our context, Model-driven engineering (MDE) techniques are intended to be used to capture the description of service variability and product-line capabilities. MDE constitutes the most recent evolution of models usage in software engineering [12]. MDE generalizes the usage of models and put them at the core of the software development process. Moreover, MDE is a promising approach to address platform complexity and to express domain concepts through domain-specific modeling languages described using metamodels. Models transformation ensure the consistency between application implementations and analysis information associated with functional and QoS requirements captured by models (*i.e* transformation of platform independent models into platform-specific models). Finally, we will use those models in order to generate the intended SPL behaviour on various grid infrastructures.

3.2 Principles

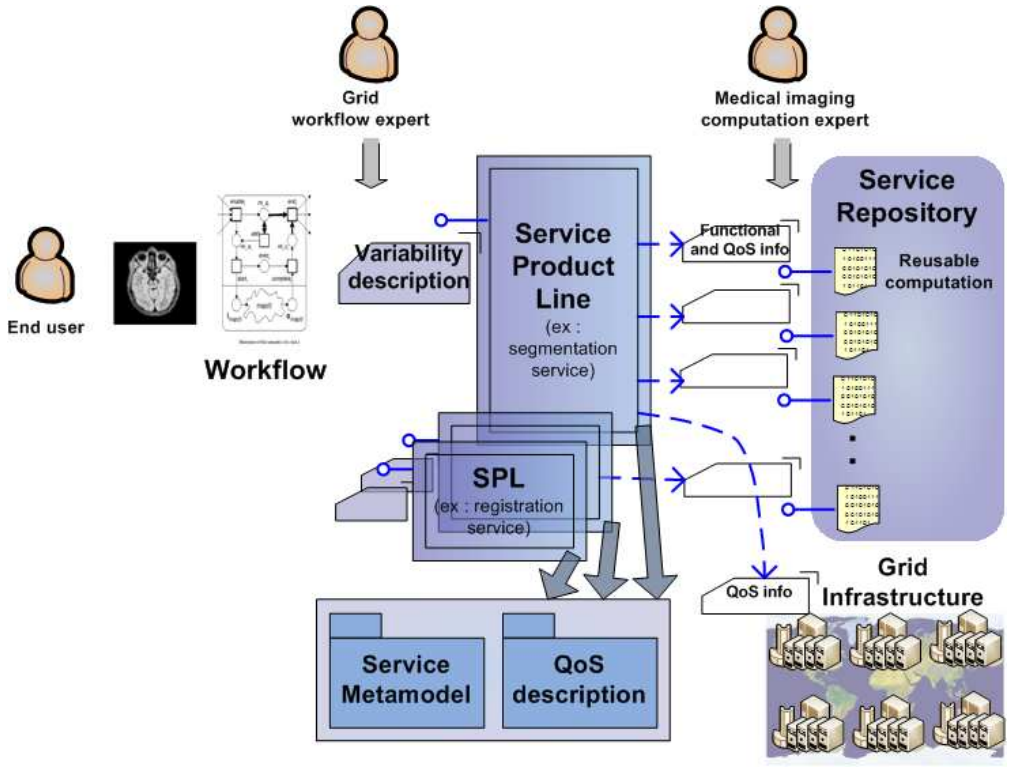


Figure 1: Overview of the Software Product Line Framework

Figure 1 sets the principles of our approach. It relies on i) a service product line framework (SPLF) describing the business domain, ii) a service repository, containing legacy services of the business domain and iii) metamodels, which capture the knowledge of the SPLF. Some metamodels specify SOA and QoS information for handling variability and other ones describe valuable information from grid infrastructures, which have an impact on SPL variability.

The software product line framework (SPLF). It describes possible workflows for the domain of medical imaging. It considers two levels of variability, *i.e* workflow variability and service variability. At both levels there are:

- a set of common properties (a structured list of assumptions that are true for all members of the domain). For example, any service for image segmentation requires a medical image as input;
- a set of possible differences. For example the format of medical image may vary depending on the service that is chosen (DICOM, Nifti, Analyze, *etc.*), so that it is necessary to record that a service supports or not a given format. This is an example of a variation point identified at the service level. The choice to insert or not one type of service (for example registration or segmentation) is another variation point, but at the workflow level.

The service repository. Services of the repository are making a collection of algorithms for image processing. According to the SPLF description, one or several services may participate to one of its possible workflows. In this repository one can find for example a subset of services which are dedicated to image segmentation. Intuitively these services may be handled through an actual service (a service interface) of type *Segmentation*, which is included in one SPL description. It makes possible to consider commonalities and to set the properties corresponding to the variation points of this type of service. In other words, one service of the repository can be considered as the result of a derivation process of a dedicated service product line. Consequently, a line of services can be seen as a service that is able to provide access to multiple services, members of the line. The result is the construction of a generic interface, which describes indirectly multiple interfaces [9].

Metamodels. Variability of grid services for medical image analysis is captured in a metamodel. It makes possible to reason on services and to achieve the operations already mentioned in section 2, such as selection, adaptation and monitoring. The SPLF relies on this metamodel to represent a software product line, which corresponds to one of its instances. As the metamodel could describe all possible software product lines of the business domain (medical imaging), it is able to represent all functional and non-functional commonalities and variations of the services belonging to the repository. Each service of this repository, which belongs to the software product line, must conform to a given model and by extension (transitivity) to the metamodel. Thus, the software architect can infer a software product line considering some services of the repository. Our metamodel helps to structure the necessary information associated to services. Semantics and knowledge are used to enhance Grid functionalities: ontology-based semantic modeling is used to enhance service-based programming on the Semantic Grid. Besides, in our preliminary implementation, we rely on an ontological approach already developed and which provides medical imaging knowledge [16]. We use feature models technology and part of our metamodels can be seen as a view on this ontology [8]. For instance, medical imaging computation experts will be able to express that a segmentation algorithm has been designed to treat brain MRIs images in DICOM format and in a precise acquisition context (*e.g* acquisition equipment).

To handle the strong needs on QoS variability, a subset of the metamodel is dedicated to QoS. For instance such information is intended to be used to compare the QoS of the members of a software product line. In section 2, the analysis of evaluation mechanisms for segmentation algorithms exhibits an important variability. The variability of QoS processing mechanisms also impacts the operations to select services, control or adapt the workflow. Selecting a service according to QoS constraints implies that the service can evaluate *a priori* the QoS dimensions. On the contrary, a few services are only able to support dynamic computation

and requires the knowledge of the output; and in this case, these services are well-adapted to perform appropriate monitoring at runtime. That is why our metamodels expresses also variability in QoS mechanisms. Besides, the impact of the grid on the SPL is intended to be handled through another metamodel that records information of the grid infrastructures.

Product derivation process. Thanks to the SPLF, the repository and the metamodels, it is possible to derive services from a given software product line. The main focus during product derivation is on satisfying complex dependencies, *i.e* dependencies that affect the binding of a large number of variation points, such as quality attributes. A key aspect in resolving these dependencies is to have an overview on these complex dependencies and how they mutually relate. An example of a complex dependency is a restriction on memory usage of a software system. An example of a relation to other dependencies is how this restriction interacts with a requirement on the performance. These examples show the need for the first-class representation of dependencies, including complex dependencies, in variability models and the need for appropriate means to model the relations between these dependencies. Thus, the SPL provides to software architect a generic interface describing the set of functional and non functional characteristics and means to express constraints in order to choose the most adapted service for each workflow task.

3.3 Open Issues

According to the principles described in Section 3.2, the service product line framework is intended to provide several functionalities, but their realization makes some open issues arise. We plan to tackle all these issues by the provision of specific operations on the service and QoS models used by the SPLF. We summarize these issues as the capability:

- to specify non functional properties toward different perspectives, according to multiple levels of abstraction, and used by all actors of the grid. Moreover, different views of QoS may cooperate. In particular, the proposed framework should allow one to transform high-level QoS properties, described by the pratictioner and guided by the clinical context, into fine-grained non functional properties of medical imaging services. For this purpose, model transformation – a key notion in the MDE approach – will be used.
- to provide to the workflow middleware means to select the best services of the repository, from the software product line, considering specific QoS properties. One possibility is that workflow engines ask to the software product line qualities offered by its members. The workflow scheduler then gets information elements in order to reason on abilities of members of the software product line. Another option is that the reasoning process is directly delegated to the software product line, which would be able to instantiate the member of the line most adapted to the associated context and constraints.
- to infer a software product line considering some of the services of the repository, by detecting automatically their common elements and their variation points.
- to master (statically or dynamically) in the derivation process i) the uncertainty of the behaviour of medical imaging services, for instance the subjectivity or even the impossibility to compute the QoS of services, ii) the context-dependency (*i.e* medical or grid context) of elements of services, iii) the intradependencies between the elements of the service (*i.e* intradependencies between QoS offered by services).

4 Conclusion

This paper has proposed to tackle the variability issues in grid services for medical imaging by using an approach based on software product lines. Functional and non functional variability of imaging services have been analysed using the segmentation step as a running example. As a start, we focused on issues in building a product line on services only, providing a service product line framework. This line will notably enable service providers and workflow experts to capture the commonalities and the differences of legacy services and to use the line to select appropriate services according to functional and non functional criteria. The service product line framework is currently undergoing implementation. All metamodels are currently operational and first validation on specific medical imaging workflows are going to start. Our long term goal is to provide a complete *service product line* that would describe major variations in functional and non functional medical imaging service specifications, as well as in process chains described through workflows. As the medical imaging field exacerbates the variability problems, we also expect that the resulting solutions are going to be applicable to other data-intensive uses of grid infrastructures and general service oriented architectures.

References

[1] Michalis Anastasopoulos and Dirk Muthig. An evaluation of aspect-oriented programming as a product line implementation technology. In *ICSR*, volume 3107 of *Lecture Notes in Computer Science*, pages 141–156. Springer, 2004. 3.1

[2] Jan Bosch, Gert Florijn, Danny Greefhorst, Juha Kuusela, J. Henk Obbink, and Klaus Pohl. Variability issues in software product lines. In Frank van der Linden, editor, *PFE*, volume 2290 of *Lecture Notes in Computer Science*, pages 13–21. Springer, 2001. 1

[3] Ivona Brandic, Siegfried Benkner, Gerhard Engelbrecht, and Rainer Schmidt. Qos support for time-critical grid workflow applications. *e-science*, 0:108–115, 2005. 2.3

[4] Ivona Brandic, Rainer Schmidt, Gerhard Engelbrecht, and Siegfried Benkner. Towards quality of service support for grid workflows. In *Proceedings of the European Grid Conference 2005 (EGC2005)*, Amsterdam, The Netherlands, 2 2005. Springer Verlag. 2.3

[5] J. S. Cardoso and L. Corte-Real. Toward a generic evaluation of image segmentation. *Image Processing, IEEE Transactions on*, 14(11):1773–1782, 2005. 2.1

[6] Jorge Cardoso, Amit P. Sheth, John A. Miller, Jonathan Arnold, and Krys Kochut. Quality of service for workflows and web service processes. *J. Web Sem.*, 1(3):281–308, 2004. 2.3

[7] Sholom Cohen, editor. *Service Oriented Architectures and Product Lines - What is the Connection? SOAPL'07, Workshop at SPLC'07, 10 September 2007, Kyoto*, 2007. 1, 3.1

[8] Krzysztof Czarnecki, Chang Hwan Peter Kim, and Karl Trygve Kalleberg. Feature models are views on ontologies. In *SPLC '06: Proceedings of the 10th International on Software Product Line Conference*, pages 41–51, Washington, DC, USA, 2006. IEEE Computer Society. 3.2

[9] Cristina Feier, Dumitru Roman, Axel Polleres, John Domingue, Michael Stollberg, and Dieter Fensel. Towards intelligent web services: Web service modeling ontology (wsmo). In *Proceedings of the International Conference on Intelligent Computing (ICIC)*, Hefei, China, 8 2005. 3.2

[10] Ian Foster. What is the grid? - a three point checklist. *GRIDtoday*, 1(6), July 2002. 2.3

[11] P. Jannin, J. Fitzpatrick, D. Hawkes, X. Pennec, R. Shahidi, and M. Vannier. Validation of medical image processing in image-guided therapy, 2002. 2.2, 2.3

[12] Stuart Kent. Model driven engineering. In Michael J. Butler, Luigia Petre, and Kaisa Sere, editors, *IFM*, volume 2335 of *Lecture Notes in Computer Science*, pages 286–298. Springer, 2002. 3.1

[13] Charles W. Krueger. Towards a taxonomy for software product lines. In Frank van der Linden, editor, *PFE*, volume 3014 of *Lecture Notes in Computer Science*, pages 323–331. Springer, 2003. 1

[14] Philippe Lahire, Brice Morin, Gilles Vanwormhoudt, Alban Gaignard, Olivier Barais, and Jean-Marc Jézéquel. Introducing Variability into Aspect-Oriented Modeling Approaches. In *ACM/IEEE 10th International Conference on Model Driven Engineering Languages and Systems (Models 2007)*, volume LNCS 4735 of *LNCS*, pages 498–513, Nashville TN USA, October 2007. Vanderbilt University, springer-verlag. 3.1

[15] Aleksandra Popovic, Matas de la Fuente, Martin Engelhardt, and Klaus Radermacher. Statistical validation metric for accuracy assessment in medical image segmentation. *International Journal of Computer Assisted Radiology and Surgery*, 2(3-4):169–181, December 2007. 2.2

[16] Lynda Temal, Michel Dojat, Gilles Kassel, and Bernard Gibaud. Towards an ontology for sharing medical images and regions of interest in neuroimaging. *Journal of Biomedical Informatics*, 2008. To appear. 3.2

[17] Jayaram K. Udupa, Vicki R. Leblanc, Ying Zhuge, Celina Imielinska, Hilary Schmidt, Leanne M. Currie, Bruce E. Hirsch, and James Woodburn. A framework for evaluating image segmentation algorithms. *Computerized Medical Imaging and Graphics*, 30(2):75–87, March 2006. 2.2

[18] Marek Wieczorek, Andreas Hoheisel, and Radu Prodan. Taxonomy of the multi-criteria grid workflow scheduling problem. In *CoreGrid Workshop*, 2007. 2.3

[19] J. Yu and R. Buyya. A taxonomy of workflow management systems for grid computing, 2005. 2.3

[20] Yu Jin Zhang. A review of recent evaluation methods for image segmentation. In *Signal Processing and its Applications, Sixth International, Symposium on. 2001*, volume 1, pages 148–151, Kuala Lumpur, Malaysia, 2001. 2.1, 2.2

[21] Tewfik Ziadi and Jean-Marc Jézéquel. *Software Product Lines*, chapter Product Line Engineering with the UML: Deriving Products, pages 557–586. Number ISBN: 978-3-540-33252-7. Springer Verlag, 2006. 3.1