

Architecturation d'Applications Orientées-Objet : Application aux Architectures Logicielles Embarquées

Encadrant principal : Abdelhak-Djamel SERIAI

courriel : seriai@ensm-douai.fr

tél. : 03 27 71 23 81

Co-encadrant : Sylvain Chardigny

Mots clés

Architecture logicielle, Architecturation, Re-ingénierie, Système embarqué, UML, Java.

Contexte et problème général

La modélisation et la représentation de la notion d'architecture logicielle est devenue aujourd'hui une phase importante du processus de développement des systèmes informatiques. L'architecture d'un système décrit sa structure à un haut niveau d'abstraction sous la forme d'un graphe bi-parti dont les composants et les connecteurs forment les sommets. Ce niveau d'abstraction offre ainsi de nombreux avantages tout au long du cycle de vie du logiciel.

Cependant l'architecture du système n'est pas toujours disponible. Nous distinguons deux cas de figures. Le premier est celui des systèmes ne disposant pas d'une représentation de leurs architectures. C'est le cas, par exemple, des systèmes patrimoniaux qui ne possèdent pas d'architectures définies. Le deuxième cas de figure est dû aux écarts qui peuvent exister entre l'architecture d'un système et la représentation dont on dispose : c'est le phénomène d'érosion. Ce phénomène est d'abord provoqué par les écarts entre l'architecture prévue et documentée et celle implémentée. Ensuite, pendant les différentes phases d'évolution et de maintenance, cette érosion peut s'accroître principalement à cause du manque de synchronisation entre l'architecture et sa représentation.

Dans les deux cas, l'absence d'architecture ou pire la présence d'une architecture erronée rendent la maintenance, l'évolution et la migration dangereuses.

Objectif et travail à réaliser

Partant des constats mentionnés dans la section précédente, nous nous sommes intéressés, dans le cadre du projet ROMANTIC (re-engineering of object-oriented systems by architecture extraction and migration to component based ones.), au développement d'une approche permettant d'extraire l'architecture logicielle d'un système existant (i.e système patrimonial) à partir de son implémentation objet ; c'est le processus d'architecturation.

Suivant ce processus l'architecture d'un système orienté objet existant est obtenue par l'identification de ses éléments architecturaux : ses composants, ses connecteurs et sa configuration. Dans ce cadre, nous définissons un composant architectural comme le regroupement d'un ensemble d'éléments objets (classes objets, interfaces, etc.). Ces groupements sont identifiés grâce à la définition d'une partition de l'ensemble des classes du système, où chaque partie représentera un futur composant.

Le processus d'identification est guidé par les considérations suivantes :

- La sémantique des éléments architecturaux qui détermine quels groupements d'éléments objet pouvant être considérés comme l'implémentation d'éléments architecturaux
- Les préoccupations concernant la qualité de l'architecture logicielle cible ainsi que celle de ses éléments (maintenabilité, portabilité, utilisabilité, etc.). Cet aspect permet de répondre à la question « Quelles sont les configurations architecturales du système objet existant, qui peuvent répondre aux exigences de qualité souhaitées ? ».
- Le domaine d'application et le style architectural sous-jacent.

- Le point de vue du concepteur matérialisée par les documents de conception identifiant les fonctions du système objet existant (e.g cas d'utilisation)
- Les propriétés de l'architecture matérielle cible (e.g. architecture embarquée avec des propriétés de performances et de ressources limitées).

L'objectif de ce stage est d'étudier le problème d'architecture d'applications orientés objet. Il s'agit d'étendre le processus d'architecture existant pour une prise en compte de :

- Le point de vue du concepteur de l'application objet. Ce point de vue sera considéré comme étant donné sous forme de diagrammes UML.
- Les propriétés inhérentes aux architectures logicielles embarquées.

Ainsi, le travail à réaliser dans le cadre de ce stage consiste en :

- 1) L'étude des propriétés intrinsèques aux architectures logicielles pour systèmes embarqués.
- 2) La prise en compte et ainsi le paramétrage du processus d'architecture par les propriétés de l'architecture matérielle embarquée cible.
- 3) La prise en compte et ainsi le paramétrage du processus d'architecture par les informations relatives à la conception orientée objet du système embarqué.
- 4) L'application du processus d'architecture sur deux applications orientées objets embarquées respectivement sur lecteurs DVD et téléphones portables.

Références

1. M. Oussalah & al. Ingénierie des composants logiciels Principes et fondements, juin 2005. Vuibert.
2. F. Lüders, I. Crnkovic, and A. Sjögren. Case study: Componentization of an industrial control system. In Proc. 26th COMPSAC 2002, Oxford, UK, Aug. 2002. IEEE Computer Society Press.
3. C. Szyperski. Component Software: Beyond Object-Oriented Programming. ACM, Press and Addison-Wesley, 1998.
4. Sandeep Agrawal and Pankaj Bhatt, *Real-time Embedded Software Systems: an introduction*, 2001.
5. Frank Lüders, Daniel Flemstrom, Anders Wall, and Ivica Crnkovic, *A Prototype Tool for Software Component Services in Embedded Real-Time Systems*, 2006.
6. F. Lüders, D. Flemström, A. Wall, and I. Crnkovic, "A Prototype Tool for Software Component Services in Embedded Real-Time Systems", *Proceedings of the 9th CBSE* 2006.
7. van Ommering, R., van der Linden, F., Kramer, J.: The koala component model for consumer electronics software. In: IEEE Computer, IEEE (2000) 78–85.
8. H. Hansson, M. Akerholm, I. Crnkovic, M. Torgen, "SaveCCM – A Component Model for Safety-Critical Real-Time Systems", 30th EUROMICRO Conference, 2004.
9. site : http://en.wikibooks.org/wiki/Embedded_Systems
10. site : <http://www.nexwave-solutions.fr/frontend/asps/index.asp>
11. Pinzger, M., Gall, H., Jazayeri, M., Riva, C.: Extracting Architectural Views from Large Telecommunications Software: A Case Study. Technical Report TUV- 1841-2002-50, Vienna University of Technology. (2002)
12. Riva, C.: Architecture Reconstruction in Practice. In: Proceedings of the 3rd Working IEEE/IFIP Conference on Software Architecture (WICSA). (2002)
13. Riva, C., Yang, Y.: Generation of Architectural Documentation using XML. In: Proceedings of the 9th Working Conference on Reverse Engineering (WCRE). (2002) 161-169
14. K.De Hondt, A Novel Approach to Architectural Recovery of Object-Oriented Systems, PhD Theses, Vrije Universiteit Brussel, 1998.
15. Mary Shaw and David Garlan. Software Architecture. Perspectives on an Emerging Discipline. Prentice Hall, 1996.
16. T. A. Wiggerts, Using Clustering Algorithms in Legacy System Remodularization, Working Conference of Reverse Engineering WCRE '97, pp. 33-43
17. N. C. Mendonça, "Software architecture recovery for distributed systems", PhD Thesis, Imperial College, Department of Computing, 1999.
18. R. Kazman and S. J. Carriere. Playing Detective: Reconstructing Software Architecture from Available Evidence. *Automated Software Engineering*, 6(2):107–138, Apr. 1999.
19. Nenad Medvidovic and Vladimir Jakobac. Using software evolution to focus architectural recovery,. Volume 13, Number 2 / April, 2006. Special Issue Section on Software Architecture Recovery. Automated Software Engineering journal. Springer Netherlands.