

5.2.2.2 Aspect d'implémentation

Pour implémenter l'approche proposée nous avons considéré l'application de la gestion d'une bibliothèque telle que la gestion des emprunts d'ouvrages, réservation et retour des ouvrages, etc. Les changements dans ce système consistent en l'ajout ou la suppression de quelque cas d'utilisation, ce qui aura un impact sur les diagrammes.

Chaque cas d'utilisation et cas de changement est mis dans un composant Fractal primitif comme montré dans la figure 5.15, ces composants implémentent des classes Java simples qui contiennent des méthodes (Figure 5.16).

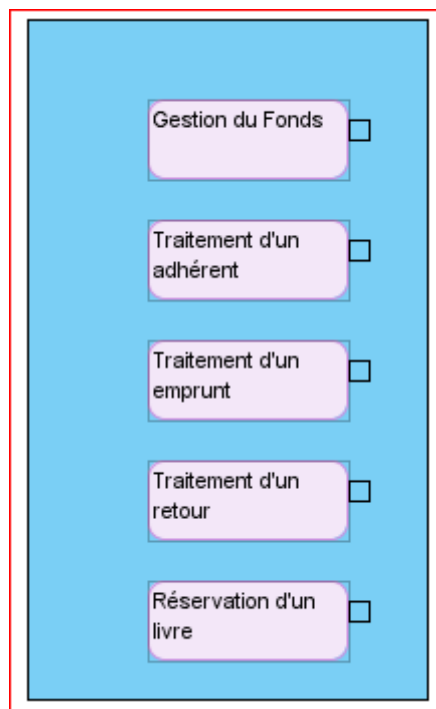


Figure 5.15 Les composants des cas d'utilisation du système de gestion d'une bibliothèque

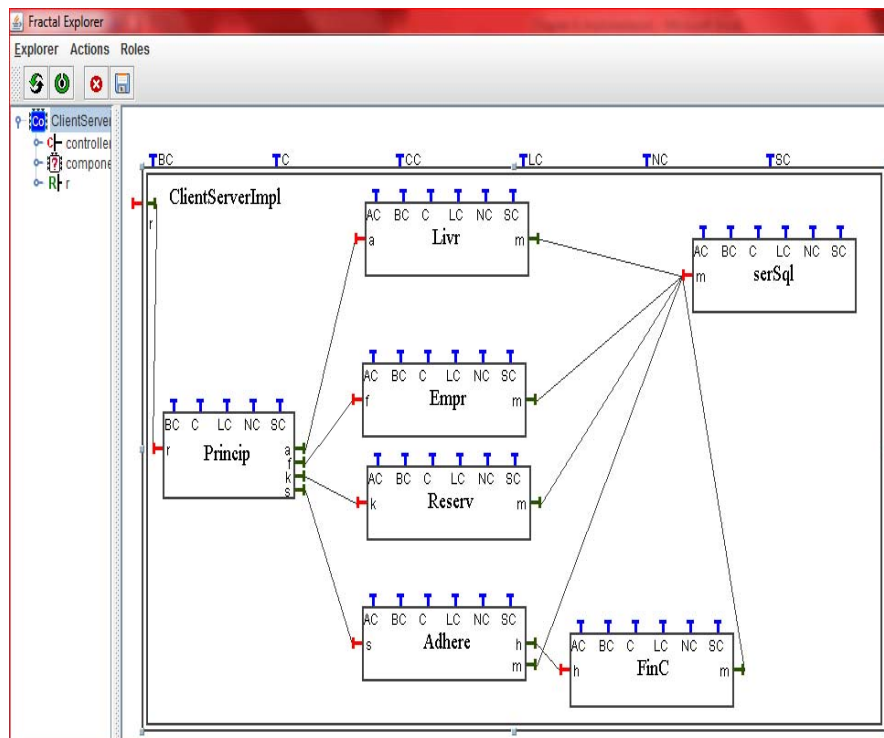


Figure 5.16 Architecture Fractal du système de gestion d'une bibliothèque

Dans notre démarche, nous avons créé :

- les composants primitifs
- une class Java qui va contenir la logique du composant (Adherent.java, Emprunt.java, Principale.java, Sql.java, FinCyc.java)
- des interfaces Java qui vont être associées avec les interfaces Fractal (AdherentINT.java, SqlINT.java, EmpruntINT.java,.....) chaque interface va exposer la méthode que nous voulons définir comme interface de composant (figure 5.17)

```

1
2
3
4
5 import javax.swing.JPanel;
6
7 public interface AdherentINT {
8
9     public void AjoutAdh(int r);
10
11 }
12

```

Figure 5.17 Interface Java du composant Adhérent Fractal

La figure 5.18 montre une partie de l'application générée.

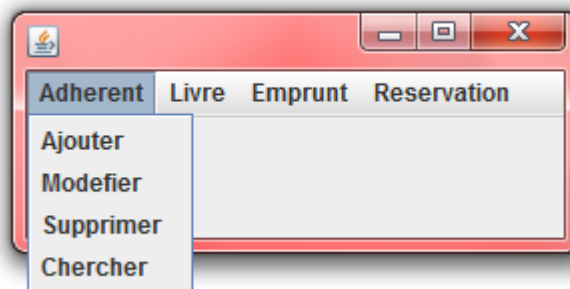


Figure 5.18 La fenêtre Principale

Conclusion

Dans ce chapitre, nous avons donné un aperçu sur la démarche proposée Onto-RUP à travers un exemple d'application simple. Nous avons donné les étapes nécessaires pour l'analyse des besoins d'évolution anticipée en utilisant les artefacts élaborés durant les activités RE d'ONTO-RUP (cas d'utilisation et cas de changement). Dans l'approche RUP l'outillage du processus de développement est primordial pour assister les développeurs dans leurs activités tout au long du cycle de développement. Onto-RUP prévoit l'utilisation d'autres AGL (comme ORCTOOL) pour la maîtrise de l'analyse, la conception et l'implémentation des évolutions des besoins anticipées et non-anticipées.

CONCLUSION GENERALE

Les activités économiques utilisent de plus en plus de logiciels pour atteindre les objectifs souhaités. Ceci a conduit à des systèmes centrés logiciel de plus en plus complexes et soumis aux exigences croissantes des utilisateurs en termes de qualité et de fonctionnalités. Ces systèmes doivent en plus pouvoir évoluer au cours de leur existence afin de répondre de manière cohérente aux changements dans leur environnement, que ces derniers soient liés à l'évolution du marché, aux exigences des utilisateurs, ou à l'évolution des technologies. Ainsi, les systèmes logiciels doivent avoir une capacité d'adaptation importante.

De nos jours, faire évoluer un système logiciel est un défi réel. Pour cela de nombreux travaux en génie logiciel tentent d'apporter des solutions aux problèmes de l'évolution.

Bien que plusieurs approches soient actuellement proposées comme des solutions aux problèmes de l'évolution, elles demeurent néanmoins des palliatifs. De plus, les chercheurs admettent, actuellement, que les solutions aux problèmes de la maintenance nécessitent une approche interdisciplinaire qui traite toutes les facettes de l'évolution, étudie les contributions des approches et évalue leurs apports.

La prise en compte de l'ontogenèse constitue un nouveau défi et une vision radicale de l'évolution qui a un effet aussi bien sur notre façon de percevoir les systèmes logiciels que notre manière de les concevoir. Bien que l'approche ontogénétique apporte une vision radicale de l'évolution des systèmes qui imite le processus biologique de l'ontogénèse et est, de ce fait, censée éliminer de nombreux problèmes dont souffre les approches actuelles de l'évolution, nous avons très vite, constater que cette vision radicale pose des défis importants aux niveaux les plus abstraits du cycle de vie du logiciel. L'ontogenèse des systèmes logiciels fait référence à la capacité qu'a un logiciel à évoluer dynamiquement et d'une manière autonome qui lui permet de s'accommoder des exigences des utilisateurs et de l'évolution de ces dernières qu'elles soient anticipées ou non-anticipées. L'évolution d'un système ontogénétique a la particularité d'être un processus continu qui le transforme depuis sa création initiale.

Cette caractéristique n'est pas en accord avec les méthodes actuelles de développement qui considèrent que l'évolution est un processus sporadique. En conséquence, les plates-formes, les outils et les méthodologies que nous employons pour développer les systèmes logiciels ne sont pas appropriés et un effort considérable est nécessaire pour les adapter afin de supporter l'ontogénèse de logiciel. Pour tirer profit du paradigme ontogénétique, nous avons entamé dans ce travail de thèse une démarche de développement des systèmes ontogénétiques.

Pour cela, nous avons opté pour une extension d'une méthodologie existante de développement de systèmes logiciels. L'extension de méthodologie existante est une voie de recherche pragmatique. En effet, il n'est pas justifié dans un contexte universitaire d'investir dans la création complète d'une méthodologie dédiée. Deux catégories de méthodologies de développement de logiciels existent : les méthodes traditionnelles, qui sont axées-plan et sont lourdes, et celles dites agiles qui ont un caractère léger, flexible et adaptatif aux changements. Nous avons opté pour l'extension du Processus Unifié Rational (RUP), qui est l'une des méthodologies agile, qui se fixe comme objectifs d'assurer le développement de logiciels de haute qualité, qui respectent les besoins des utilisateurs, l'échéancier et le budget. L'extension concerne l'adjonction de nouvelles disciplines pour modéliser les évolutions anticipées et non anticipées et d'autres éléments de modélisation.

Concernant la discipline d'ingénierie des besoins des évolutions anticipées, l'approche ONTO-RUP propose un processus d'extraction de changements futurs ayant comme artefacts d'entrée : le modèle orienté-but et le modèle de cas d'utilisation construit durant la discipline RE du RUP. Le processus d'extraction utilise la méthode « Roue des futurs ». Les artefacts de sortie de cette discipline, étant le modèle des cas de changement.

Le travail entamé dans cette thèse part d'une nouvelle vision pour l'ingénierie des systèmes logiciels et il est nécessaire de rappeler qu'il présente un challenge pour plusieurs raisons :

- La complexité du processus RUP
- Le niveau d'abstraction élevé de l'ingénierie des besoins
- Le manque de travaux existants dans la littérature concernant les évolutions non anticipées
- Absence de travaux sur le développement continu de systèmes logiciels

Résumé des contributions

Nous estimons que ce travail nous a permis d'apporter certains points de vue, d'identifier des concepts, des mécanismes et d'en proposer d'autres. Nous avons maîtrisé le modèle Mage et son apport pour la modélisation de l'ontogenèse.

Nous avons fait une étude comparative entre les méthodologies de développement des systèmes logiciel traditionnelles et agiles, en présentant les points faibles et points forts de chacune d'entre elles.

Nous avons consacré une section importante dans le deuxième chapitre pour le formalisme UML, qui reste une référence incontournable dans le domaine du génie logiciel, permettant la représentation des modèles objets en intégrant différents diagrammes permettant de

structurer le système et son comportement. Néanmoins, le processus d'élaboration de ces modèles n'est pas décrit par UML. Le RUP étant basé sur UML et ce fait constitue une raison supplémentaire qui nous a fait opter pour l'extension de RUP plutôt que de proposer une nouvelle méthode. Le processus RUP a été présenté dans ce manuscrit avec plus de détails car il représente le noyau de la démarche proposée à différents niveaux : sa structure statique et dynamique, ses concepts, ses principes, etc.

L'extension du RUP représente la contribution la plus importante de ce travail de thèse. Il est nécessaire de noter que nous nous sommes intéressés particulièrement à l'ingénierie des besoins d'évolution car c'est la phase la plus abstraite et critique dans le cycle de vie du système ontogénétique.

Nous avons proposé un processus pour l'ingénierie des besoins d'évolution en combinant plusieurs méthodes existantes dans la littérature (modèle des cas d'utilisation du RUP, modèle Futures wheel, le modèle orienté but i^* des GORE) pour produire le modèle des cas de changement.

Perspectives de recherche

Les perspectives de ce travail sont nombreuses. Dans un premier lieu, il conviendra de combler l'approche d'ingénierie des besoins anticipés en proposant une approche de passage adéquate du modèle orienté but adapté vers le modèle de cas de changement.

Dans un second lieu, il est nécessaire d'envisager l'extension de RUP à différents niveaux et non plus seulement comme ajout de discipline d'ingénierie des besoins, préparation à la maintenance et la mise à jour dynamique.

Comme troisième perspective, il est question de considérer les changements non anticipés qui nécessitent la prise en compte de l'état actuel d'un système et les changements déjà existants dans son génotype. De même la vérification formelle des propriétés d'un système suite aux changements et le maintien de certaines qualités de service constituent une perspective qui nécessite beaucoup d'efforts. Aussi il est important de noter que l'aspect implémentation des changements et l'estimation des coûts des efforts sont des voies de recherches prometteuses.

REFERENCES

- [Ada 98] C. Adami, Introduction to Artificial Life, Springer-Verlag, New York, Inc., 1998.
- [Adi 13] F. Adikara, B. Sitohang, and B. Hendradjaya, "The Emergence of User Requirements Risk in Information System Development for Industry Needs," in *6th International Seminar on Industrial Engineering and Management*, 2013.
- [Aiz 06] N. Aizenbud-Reshef, B. T. Nolan, J. Rubin, Y. Shaham-Gafni, Model traceability, *IBM Systems Journal*, Citeseer, 2006.
- [Alm 06] Hyggo Almeida, Angelo Perkusich, Evandro Costa, Glauber Ferreira, Emerson Loureiro, Loreno Oliveira, and Rodrigo Paes. A Component Based Infrastructure to Develop Software Supporting Dynamic Unanticipated Evolution. In *Anais do XX Simposio Brasileiro de Engenharia de Software*, pages 145–160, Florianópolis, SC, Brasil, 2006.
- [Arm 05] Armarego, J., (2005), Studio Learning of Requirements: towards aligning teaching to practitioner needs, *In: 1st International Workshop on Requirements Engineering Education and Training (REET) (2005)*.
- [Arn 93] Arnold, R.S.: Software Reengineering. IEEE Computer Society Press (1993) [5, 291]
- [Aur 05] A. Aurum and C. Wohlin, editors. Engineering and Managing Software Requirements. Springer Verlag, Berlin Germany, 2005.
- [Bac 05] Bachmann, F. and Clements, P. C. (2005) 'Variability in Software Product Lines', Technical Report CMU/SEI-2005-TR-012, September 2005, Software Engineering Institute, Carnegie Mellon University.
- [Ben 12] Lamia Ben Hiba, Mohammed Abdou Janati Idrissi, « Tendances des méthodes de gestion des projets », eTI, Revue électronique en Technologies de l'Information. <http://www.revue-eti.net> , 31 Mars 2012.
- [Ben 13] Benabbas Zine Eddine, Un Outil Case pour la Manipulation des Cas de Changements, mémoire de fin d'étude Master, Département d'informatique, Université Badji Mokhtar-Annaba, 2013.

- [Ber 98] Berry DM, Lawrence B (1998) Requirements engineering. IEEE Software 25(2): 26-29.
- [Ber 07] Mario M. Berón, Pedro R. Henriques, Maria J. Varanda, Roberto Uzal, Program Inspection to inter-connect the Operational and Behavioral Views for Program Comprehension², First York Doctoral Symposium on Computing. York. England. 2007.
- [Ber 98] Berry DM, Lawrence B ,Requirements engineering. IEEE Software 25(2): 26-29,(1998).
- [Bie 03] G. Bierman et al, Formalising Dynamic Software Updating, Second International Workshop on Unanticipated Software Evolution, April 5 - 13, 2003, Warsaw, Poland, <http://joint.org/use/2003/>
- [Bla 72] A.W. Blackman Jr., A mathematical model for trend forecasts, Technol. Forecast. Soc. Change 3 (1972) 441–452.
- [Boe 81] B.Boem, “Software Engineering Economics”, Prentice Hall, 1981
- [Boi 03] S. Boissau and J. C. Castella, “Constructing a common representation of local institutions and land-use systems through simulation gaming and multiagent modeling in rural areas of northern Vietnam: The sambaweek methodology,” Simulation & Gaming, vol. 34, pp. 342-357, 2003.
- [Boo 91] G.Booch, Object Oriented Analysis and Design with Application, Benjamin/Cummings, 1991.
- [Bor 13] Borjiba Othmane, Une conception des cas de changement basée composant, mémoire de fin d’étude Master, institut d’informatique, université Badji Mokhtar Annaba, 2013.
- [Bub 94] Bubenko, J, A., Kirikowa, M., “Worlds” in Requirements Acquisition and Modeling, Sweden, 1994.
- [Cas 92] X. Castellani, MCO, Méthodologie générale d’analyse et de conception des systèmes d’objets – l’ingénierie des besoins, Masson, 1992.

- [Cas 02] Jaelson Castro, Manuel Kolp et John Mylopoulos. *Towards Requirements-Driven Information Systems Engineering : The Tropos Project* . Information Systems, vol. 27, pages 365–389, 2002.
- [Cha 01] N. Chapin, J. E. Hale, K. Md. Khan, J.F. Ramil, and W. G. Tan. Types of software evolution and software maintenance. *Journal of Software Maintenance and Evolution : Recherche and Practice*, 13:3-30, 2001.
- [Chi 90] Chikofsky, E.J., Cross, J.H.: Reverse engineering and design recovery: A taxonomy. *IEEE Software* **7**(1) (1990) 13–17 [5, 106, 177, 303]
- [Cis 14] Computer Information Systems, inc. <http://www.cis.com>
- [Coc 01a] Cockburn A. and Highsmith J. (2001), *Agile Software Development: The People Factor*. *Computer*, Novembre 2001, 131-133.
- [Coc 01b] A. Cockburn, *Writing Effective Use Cases*, Addison Wesley, New York, 2001.
- [Col 10] Thomas COLLONVILLÉ, *Elaboration de processus de développements logiciels spécifiques et orientés modèles*, Thèse doctorat, l'Université de Haute-Alsace , 2010.
- [Con 01] Constantine L., *Methodological Agility*, *Software Development*, 67-69, Juin 2001.
- [Cop 98] J. Coplien et al., *Commonality and variability in Software Engineering*, *IEEE Software*, pp. 37-46, November/December 1998.
- [Cys 03] Cysneiros, L.M., Kushniruk, A.: Bringing Usability to the Early Stages of Software Development. *International Requirements Engineering Conf. IEEE(2003)* 359-360
- [Dar 93] Anne Dardenne, Axel van Lamsweerde et Stephen Fickas. Goal-directed requirements acquisition. *Sci. Comput. Program.*, vol. 20, pages 3–50, April 1993.
- [Dav 93] A.M. Davis, "Software requirements, Objects, functions and states", Prentice Hall, 1993.
- [Del 96] F. Dellaert, R.D. Beer, A developmental model for the evolution of complete autonomous agents, In P. Maes, M. Mataric, J. Meyer, J. Pollack and S. Wilson (Eds.), *From animals to animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior* pp. 393-406, 1996.

- [Dow 01] J. Dowling & V. Cahill, The K-Component architecture meta-model for self-adaptive software, Reflection 2001, The third international conference on metalevel architecture and separation of crosscutting concerns, Kyoto, Japan, September 2001.
- [Eck 96] Ecklund, E. F., Delcambre, L. M. L. and Freiling, M. J. (1996) 'Change Cases: Uses Cases that Identify Future Requirements', Proceeding OOPSLA'96, ACM Press, New York.
- [Enc 14a] Le système nerveux, Encyclopédie Microsoft Encarta en ligne, Microsoft Corporation, 2005, <http://fr.encarta.msn.com>
- [Enc 14b] Le système immunitaire, Encyclopédie Microsoft Encarta en ligne, Microsoft Corporation, 2005, <http://fr.encarta.msn.com>
- [Enc 14c] Le système endocrinien, Encyclopédie Microsoft Encarta en ligne, Microsoft Corporation, 2005, <http://fr.encarta.msn.com>
- [Erd 03] Kagan Erdil, Emily Finn, Kevin Keating, Jay Meattle, Sunyoung Park, Deborah Yoon, "Software Maintenance As Part of the Software Life Cycle", Comp180: Software Engineering Project, 2003.
- [Est 12] Estier, T. and Métrailler, A. (2012) 'EVOLIS: a Framework for Evaluating Evolution of Information Systems', Actes du Colloque AIM 2012, Bordeaux, France, pp. 12.
- [Eti 04] Etien, A., Rolland, C. and Salinesi C., (2004) 'Overview of a Gap-driven Evolution Process', AWRE'04, 9th Australian Workshop on Requirement Engineering, Adelaide, Australia.
- [Fis 98] Gerhard Fischer, "Complex systems: Why do they need to evolve and how can evolution be supported", Lecture Notes in Computer Science Volume 1454, 1998, pp 92-112
- [Fow 05] Fowler, M. (2005). The new methodology [agile methodology]. Software World, 36(1), 3-6.
- [Ger 05] Germain, E., & Robillard, P. N. (2005). Engineering-based processes and agile methodologies for software development: a comparative case study. Journal of Systems and Software, 75(1-2), 17-27.
- [Gil 03] S. F. Gilbert, Developmental biology, Seventh Edition, Sinauer associates Inc. Publishers, March 2003.

- [Gir 07] Jean-Pierre Giraudin. «Complexité des systèmes d'information et de leur ingénierie». e-TI - la revue électronique des technologies d'information, Numéro 3, 9 mai 2007, <http://www.revue-eti.netdocument.php?id=1180>.
- [Gle 72] Glenn, Jerome C., 1972. 'Futurizing Teaching vs Futures Course, Social Science Record, Syracuse University, Volume IX, No. 3 Spring 1972.
- [Gle 09] Glenn, J. C. (2009) 'Futures Wheel', Futures Research Methodology Version 3.0, The Millennium Project, Washington, D.C.
- [Gor 04] T. Gordon and J. Glenn, "Integration, Comparisons, and Frontiers of Futures Research Methods," New Technology Foresight, Forecasting and Assessment Methods, Seville, 2004.
- [Gur 01] J. V. Gorp, J. Bosch, M. Svahnberg, On the Notion of Variability in Software Product Lines, Proceedings of WICSA 2001, August 2001.
- [Hag 88] Hagelstein J., Declarative Approach to Information Systems Requirements, in Knowledge- Based Systems, Vol No4, 1988.
- [Hai 08] Hainaut, J. L., Anthony, C., Henrard, J. and Hick, J. M. (2008) 'Migration of Legacy Information Systems', in Mens, T. and Demeyer, S. (eds.), 'Software Evolution'. DOI 10.1007/978-3-540-76440-3, Springer.
- [Heu 94] J. C. Heudin, La vie Artificielle, Edition Hermes, 1994.
- [Hic 01] M. Hicks. Dynamic Software Updating. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, August 2001.
- [Hum 88] HUMPHREY, W. S. (1988). The software engineering process: definition and scope. In Proceedings of the 4th international software process workshop on Representing and enacting the software process, pages 82–83, New York, NY, USA. ACM Press.
- [IBM 12] IBM corporation, Rational RequisitePro, <http://www-306.ibm.com/software/awdtools/reqpro/>, visited 20/05/2014.
- [IBM 15] <http://www.ibm.com/developerworks/rational/library/jun07/cuellar/index.html>, visited 11/01/2015
- [Jac 01] Michael Jackson. Problem frames : analyzing and structuring software development problems. Addison-Wesley Longman Publishing Co.,Inc., Boston, MA, USA, 2001.

- [Joh 05] Bruce Johson, W W.Woolfolk, R. Miller, "Flexible Software Design", Auerbach Publications, 2005.
- [Kav 06] Kavakli, E. and Loucopoulos, P, Experiences with Goal-Oriented Modelling of Organisational Change, IEEE Transactions on Systems, Man and Cybernetics-Part C, Vol N36, Issue 2, pp.221-235, 2006.
- [ket 05] Kettunen, P., & Laanti, M., How to steer an embedded software project: tactics for selecting the software process model. *Information and Software Technology*, 47(9), 587-608, 2005.
- [Khe 9a] Kherissi, F. and Meslati, D.,Developing Ontogenetic Software Systems,International Symposium on Programing Systems ISPS, Algiers, 2009.
- [Khe 9b] Kherissi, F. and Meslati, D.,ONTO-RUP: A RUP Based Approach for Developing Ontogenetic Software Systems, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, volume 39, pp 140-151, Springer, 2009, ISSN: 1867-8211.
- [Khe 14] Kherissi, F. and Meslati, D., and Tamzalit, D., An approach based on extending the RUP for dealing with anticipated changes in ontogenetic Software Systems, To appear in IJCAT (Internationa Journal for Compter Applications in technology), Inderscience Publisher, Vol. 51, No. 4.
- [Kis 03] T. Kistler, M. Franz, Continuous program optimization : A case stydy, ACM Transactions on Programming Languages and Systems, Vol. 25, No. 4, pp 500-548, July 2003.
- [Kni 02] Gunter Kniesel, Joost Noppen, Tom Mens, and Jim Buckley, Unanticipated Software Evolution, *The First International Workshop on Unanticipated Software Evolution*. 10-14 June 2002, Malaga, Spain.
- [Kot 98] G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques, John Wiley & Sons, 1998.
- [Kro 03] P. Kroll, and P. Kruchten, The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP, Addison Wesley, 2003.
- [Kru 00] Philippe Kruchten. The Rational Unified Process An Introduction, Second Edition. Addison Wesley, 2000.

- [Kru 01] P. Kruchten, "Software Maintenance Cycles with the RUP", The Rational Edge, August 2001.
- [Lam 01] A. V. Lamsweerde, Goal-oriented requirements engineering: A guided tour, in Proceedings of the 5th IEEE International Symposium on Requirements Engineering, Toronto, Canada, pp. 249-262, 2001.
- [Lam 02] Axel van Lamsweerde et Emmanuel Letier, From Object Orientation to Goal Orientation : A Paradigm Shift for Requirements Engineering, in Martin Wirsing, Alexander Knapp et Simonetta Balsamo, editeurs, RISSEF, volume 2941 of Lecture Notes in Computer Science, pages 325–340. Springer, 2002.
- [Lam 09] Axel van Lamsweerde , Requirements Engineering : From System Goals to UML Models to Software Specifications, Wiley,2009.
- [Lar 03] Larman, C., Agile and Iterative Development: A Manager's Guide, Addison-Wesley, 2003.
- [Lew 99] B. Lewin, Genes VII, Oxford University Press, Oxford, 1999.
- [Li 09] Tong Li , An Approach to Modeling Software Evolution Processes, ISBN: 978-3-540-79463-9, Springer Berlin Heidelberg New York, 2009.
- [Lin 91] F. Lints, Génétique, Troisième édition, Lavoisier Technique et Documentation, 1991.
- [Liu 01] Lin Liu et Eric Yu, From Requirements to Architectural Design : Using Goals and Scenarios. In First International Workshop From Software Requirements to Architectures (STRAW 01), Toronto, Canada, May 2001.
- [Lod 04] Lodding, K. N., Hitchhiker's Guide to Biomorphhic Software, QUEUE, Vol. 2, N°4,pp.66-75, ACM, June 2004.
- [Lov 96] D. Loveridge, Technology Foresight and Models of the Future, Policy Research in Engineering, Science and Technology, September 1996.
- [Mao 07] Y. Mao, J. Vassileva, and W. Grassmann, A System Dynamics Approach to tudy Virtual Communities, in Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS), January 2007.

- [McD 84] John McDermid et Knut Ripken, Life cycle support in the ADA environment, University Press, 1984.
- [Mea 05] Nancy R. Mead, Ted Stehney. "Security Quality Requirements Engineering (SQUARE) Methodology." SESS 2005, 2005.
- [Men 08] T.Mens, Serge Demeyer, Software Evolution, Springer Verlag, 2008.
- [Mes 04] D. Meslati & al, The MAGE Ontogenetic Model: Towards autonomously-developed software, 17th Int. Conf. on Software & Systems Engineering and their Applications, Paris, November 2004.
- [Mes 06] D. Meslati, Mage: Une approche ontogénétique de l'évolution dans les systèmes logiciels critiques et embarqués, Thèse de Doctorat d'Etat, Université de Annaba, Février 2006.
- [Mes 08] Véronique Messenger Rota, Gestion de projet, Vers les méthodes agiles, Eyrolles, édition 2008.
- [Moi 90] J.L.LeMoigne, La modélisation des systèmes complexes, Dunod, Paris, 1990.
- [Mul 00] P. A. Muller, and N. Gaertner "Modélisation objet avec UML", Eyrolles, 2000.
- [Mur 04] Murch, R. , Autonomic Computing On Demand Series, Prentice Hall, March 2004.
- [Naw 02] Nawrocki, J. R., Walter, B., & Wojciechowski, A., Comparison of CMM Level 2 and extreme programming, 7th European Conference on Software Quality (ECSQ 2002), (pp. 288-297). Helsinki, Finland: Springer-Verlag, 2002.
- [Nel 05] R. J. Nelson, An introduction to behavioral endocrinology, Third edition, Sinauer associates Inc. Publishers, 2005.
- [Ner 05] Nerur, S., Mahapatra, R., & Mangalaraj, G., Challenges of migrating to agile methodologies. Communications of the ACM, 48(5), 72-78, 2005.
- [Nur 02] Selmin Nurcan, Judith Barrios, Colette Rolland, Une méthode pour la définition de l'impact organisationnel du changement, INFORSID 2002
- [Nus 00] Bashar Nuseibeh et Steve Easterbrook, *Requirements engineering : a roadmap*, in Proceedings of the Conference on The Future of Software Engineering, ICSE '00, pages 35–46. ACM, 2000.

- [OMG 09] Object Management Group (OMG), 2009. Spécification (UML) Unified Modeling Language., <http://www.omg.org/spec/UML/2.2/>.
- [Ous 99] C. Oussalah, D. Tamzalit, L'émergence comme processus d'évolution d'objet In Ed. C. Oussalah, Génie Objet : Analyse et Conception de l'Evolution, Edition Hermes, 1999
- [O'B 03] Micheal P. O'Brien. Software Comprehension - A Review and Research Direction. Technical Report, 2003
- [Pas 06] Passing, J., Requirements Engineering in the Rational Unified Process, Seminar Requirements Engineering, Summer term, Hasso Plattner Institute for Software Systems Engineering, 2006. http://int3.de/res/RUP/RUP_Paper_JohannesPassing.pdf.
- [PCD 14] PDCA, la roue de Deming, <http://www.logistiqueconseil.org/Articles/Methodes-optimisation/Pdca-roue-deming.htm>, visité Novembre 2014.
- [Per 07] J-M. Perronne. Une contribution Objet pour la conception de systèmes logiciels de commande plus sûrs. Habilitation à diriger des recherches. Université de Haute-Alsace, 2007.
- [Phi 00] Phillips, D., The Software Project Manager's Handbook - Principles that work at Work. 1st ed. Los Alamitos, California, IEEE Computer Society, 2000.
- [Pim 11] Pimentel, J., Castro, J., Perrelli, H., Emanuel, S., and Franch, X., Towards Anticipating Requirements Changes through studies of the future, In Proceedings of The 5th IEEE International Conference on Research Challenges in Information Science, May 19-21, Gosier, Guadeloupe, (2011).
- [Ras 08] Awais Rashid et Ruzanna Chitchyan. *Aspect-oriented requirements engineering : a roadmap*. In Proceedings of the 13th international workshop on Early Aspects, EA '08, pages 35–41, New York, NY, USA, 2008. ACM.
- [Rid 96] M. Ridley, Evolution, Second edition, Blackwell scientific publications Ltd, Oxford, 1996.
- [Rol 98] Colette Rolland, Carine Souveyet et Camille Ben Achour, *Guiding Goal Modeling Using Scenarios*. IEEE Trans. Softw. Eng., vol. 24, pages 1055–1071, December 1998.

- [Rol 03] Colette Rolland, "L'INGENIERIE DES BESOINS : L'APPROCHE L'ECRITOIRE", Journal Techniques de l'Ingénieur (2003) 1.
- [Rol 05] Rolland C., L'ingénierie des méthodes : une visite guidée , *e-TI*, n° 1, (<http://www.revueeti.net/document.php?id=726>), 25 octobre 2005.
- [Roy 70] Royce, W.W., Managing the development of large software systems, IEEE WESCON, 1970.
- [Rum 96] J. Rumbaugh, A State of Mind : Modeling behavior, Modeling & Design, JOOP, Volume 9, Number 4, pp. 6-12, July/August 1996.
- [Run 04] Per Runeson Peter Greberg Extreme Programming and Rational Unified Process - Contrasts or Synonyms? SERPS'04 - Software Engineering Research and Practice in Sweden, 2004
- [Sac 10] Sachidanandam Sakthive , Manage Requirements Volatility to manage Risks in IS development projects, ISACA Journal, Volume N°5, 2010.
- [Sad 03] Salah Sadou, Rapport d'HDR de l'université de Bretagne Sud 2003. <http://www-irisa.univ-ubs.fr/Salah.Sadou/papers/hdr.pdf>
- [Sal 04] Salinesi, C., Etien, A., and Wäyrynen, J., Towards a Systematic Propagation of Evolution Requirements in IS Adaptation Projects, Australian Conference on Information Systems (ACIS), Hobart, Australia, 2004.
- [San 02] Santander, V. F. A. and Castro, J. F. B., Integrating Use Cases and Organizational Modeling, SBES02 - XVI Brazilian Symposium on Software Engineering, Gramado. October, 2002.
- [Sha 67] J.L. Shaerer and H.H. Richardson. Introduction to System Dynamics. Addison Wesley, NewYork, 1967.
- [Shu 03] U. P. Shultz et al., Automatic Program Specialization for Java, ACM Transactions on Programming Languages and Systems, Vol. 25, No. 4, pp 452-499, July 2003.
- [Som 01] I., Sommerville, Software engineering, 6th ed. Harlow, Pearson Education Ltd, 2001.
- [Som 06] I. Sommerville. *Software Engineering 8th Revised edition*. Addison-Wesley Educational Publishers Inc, United Kingdom, 2006.

- [Sri 94] M. Srinivas, L. M. Patnaik, Genetic Algorithms: A survey, Computer, pp. 17-26, June 1994.
- [Sut 03] Alistair Sutcliffe. Scenario-Based Requirements Engineering. In Proceedings of the 11th IEEE International Conference on Requirements Engineering, Washington, DC, USA, 2003. IEEE Computer Society.
- [Tem 02] G. Tempesti et al., A POEtic Architecture for Bio-Inspired Hardware, 8th Int. Conf. on the Simulation and Synthesis of Living Systems, Sydney, Australia, Dec. 2002, MIT Press, Cambridge, 2002, pp 111-115.
- [Tes 00] G. J. Tesauro and J. O. Kephart, "Foresight-based pricing algorithms in agent economies," Decision Support Systems, vol. 28, issues 1-2, pp.49-60, March 2000.
- [Tre 07] R. Tremblay, Implantation d'une méthode Agile de développement Logiciel en entreprise : Une culture accueillant le changement, mémoire pour l'obtention du grade de maître (M. Sc.), université de Laval, 2007.
- [Uni 14a] Encyclopædia Universalis en ligne, Chapitre Evolution, Mars 2004 www.universalis-edu.com/private/
- [Wau 13a] Yves Wautelet and Manuel Kolp. Mapping i* within uml for business modeling. In Joerg Doerr and Andreas L. Opdahl, editors, REFSQ, volume 7830 of Lecture Notes in Computer Science, pages 237-252. Springer, 2013.
- [Wau 13b] Yves Wautelet, and Manuel Kolp, "On the Integration of i* into RUP", Proceedings of the 6th International i* Workshop (iStar 2013), CEUR Vol-978
- [Yu 95] Eric Yu. Modelling Strategic Relationships for Process Reengineering. Ph.D. Thesis, Department of Computer Science, University of Toronto, Canada, 1995.
- [Yu 97] Eric Yu. *Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering*. In Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, RE '97, pages 226–235, Washington, DC, USA, 1997. IEEE Computer Society.
- [Yu 12] Yu, B. L., Wooi, K. L., Wai, Y. T. and Soo, F. T. (2012) 'Software Development Life Cycle AGILE vs Traditional Approaches', International Conference on Information and Network Technology, IPCSIT, vol. 37, IACSIT Press, Singapore.

- [Yue 87] K. Yue. What Does It Mean to Say that a Specification is Complete ? In Proceedings of the IWSSD-4, Fourth International Workshop on Software Specification and Design, 1987.
- [Zav 93] ZAVE, P. AND JACKSON, M. 1993. Conjunction as composition. ACM Trans. Softw. Eng. Methodol. 2, 4 (Oct.), 379–411.
- [Zet 01] Zettel, J., Maurer, F., Munch, J., & Wong, L., LIPE: a lightweight process for e-business startup companies based on extreme programming, Third International Conference on Product Focused Software Process Improvement (PROFES 2001), (pp. 255-270). Kaiserslautern, Germany: Springer-Verlag., 2001
- [Zow 96] D. Zowghi, A.K. hose and P. Peppas, A Framework for Reasoning about Requirements Evolution, Proceedings of PRICA196, Australia, 1996.

Résumé

L'ontogenèse des systèmes logiciels se réfère à la capacité des logiciels à évoluer dynamiquement et de façon autonome pour répondre aux besoins des utilisateurs et leurs changements anticipés et non anticipés. L'évolution d'un système ontogénétique a la particularité d'être un processus continu qui le façonne depuis le début de sa création. Cette particularité ne correspond pas aux méthodes actuelles de développement qui considèrent que l'évolution est un processus sporadique. Par conséquent, les plateformes, outils et méthodes que nous utilisons pour développer des systèmes logiciels ne sont pas adaptés à la modélisation de l'ontogenèse du logiciel. L'approche proposée dans cette thèse, nommée Onto-RUP, présente une démarche pour le développement des systèmes ontogénétiques basée sur une extension du Rational Unified Process (RUP). Cette extension porte sur l'adjonction de deux nouvelles disciplines et des artefacts spécifiques.

Mots clés. Evolution, RUP, développement logiciel, système ontogénétique

ملخص

إن تطور أنظمة البرمجيات يشير إلى قدرة البرامج للارتقاء بشكل ديناميكي و ذاتي لتلبية احتياجات المستخدمين والاحتياجات المتوقعة وغير المتوقعة التغيير. إن ارتقاء النظام التطوري لديه خصوصية هي انه عملية مستمرة تشكله منذ بداية إنشائه. هذه الخصوصية لا تتناسب مع أساليب التطوير الحالية التي تعتبر الارتقاء عملية متفرقة في الزمن. وبالتالي الأسس والأدوات والأساليب التي نستخدمها لتطوير أنظمة البرمجيات لا تتناسب لقيادة تطور البرمجية. و تعرض المقاربة "أنتوروب" المقترحة في هذه الأطروحة، نهجاً لتطوير الأنظمة التطورية على أساس امتداداً لروب. يشمل هذا الامتداد إضافة اختصاصين جديدين وبعض عناصر النمذجة.

كلمات مرشدة. إرتقاء، روب، تطوير البرمجيات، نظام تطوري