On challenges and evolutions for exception handling
Abstract of LADC-EHCOS'11 Presentation
Latin-American Conference on Dependable Computing - Exception Handling Workshop

Christophe Dony*
*LIRMM, CNRS and Montpellier II University Montpellier, France

Software developers face a number of challenges related to exception handling by first having to take into account new development paradigms such as agents [SDUV04], components [RdLFF05], services, ambient systems [MDB+06], pervasive systems [MECL10], aspects, product lines programming [TDR+11] etc. It has to be noted that new paradigms raise new issues but also propose new challenging solutions for exception handling, as it has been the case for opbect [Don90b].

They also have the challenging task to deal with exception handling strategies in a multi-dimensional context that include :

- the life cycle dimension [dLR01] : exceptions strategies have to be specified, designed, programmed, tested and updated
- the structural dimension of applications architectures [MECL10] : exceptions strategies should be detailed not only at the programs blocks level but also at the class, module, package, system level. It should be possible to design strategies at different conceptual levels.
- the dimension of coordination for fault-tolerance : exception handling strategies should be compatible and behave correctly in presence of other companion techniques for software reliability such as for example the different forms of redundancy.

These challenges are compounded by a global conjunctural issue, if on the one hand there is a global agreement for what concerns programming languages primitives for exception handling[GRRX01], on the other hand there is a crucial lack of standards[RDKT04] for what concerns : (1) the terminology : the following terms, exception, error, failure, condition, alarm, etc, used in our domain papers denote either the same thing or subtly different things, (2) the classification of exceptions kinds : early terms such as "domain", "range" or "monitoring" exceptions introduced by Goodenough's seminal paper are not standardized and almost each new system provides its own classification, (3) the common patterns to handle exceptions or to write fault-tolerant or defensive programs. Research results on exception handling are only very partially integrated into mainframe software programming environments and languages (for example, UML and Java). Today's developers using these systems do not benefit from known results, they frequently reinvent existing solutions or make well-known mistakes.

We will globally discuss the above challenges and issues while presenting two systems.

The first system [DUV06] is dedicated to all kind of autonomous software agents communicating with each other asynchronously, such agent subsume autonomous distributed components and services and raise various issues common to new development paradigms : preserving agent autonomy, taking into account collaborative concurrent activities, providing mechanisms for their coordination, and handling concurrent exceptions, signaling and researching handlers asynchronously[Kri10], and executing handlers in the correct definition context.

The second system [DTUV08] addresses the coordination dimension; it proposes a specification and an implementation of a combination of two independent but complementary mechanisms : exception handling and replication. The main advantages of this combination are: to provide agent programmers with an exception handling system that behaves correctly in presence of seamless active replication, to improve replication strategies, on the base of information conveyed while propagating exceptions from replicas. It also offer new solutions to implement the forgotten resumption strategy by providing active copies of the computation state.

References

[BDBR08] I.A. Bertoncello, M.O. Dias, P.H.S. Brito, and C.M.F. Rubira. Explicit exception handling variability in component-based product line architectures. In *Proceedings of the 4th international workshop on Exception handling*, pages 47–54. ACM, 2008.

[Bla82] Andrew P. Black. *Exception Handling: The Case Against.* Phd dissertation, University of Oxford, January 1982.

[CCF+09] F. Castor, N. Cacho, E. Figueiredo, A. Garcia, C.M.F. Rubira, J.S. de Amorim, and H.O. da Silva. On the modularization and reuse of exception handling with aspects. *Software: Practice and Experience*, 39(17):1377–1417, 2009.

[CFBR06] F. Castor Filho, P.H.S. Brito, and C.M.F. Rubira. Specification of exception flow in software architectures. *Journal of Systems and Software*, 79(10):1397–1418, 2006.

[DKRT06] C. Dony, J. L. Knudsen, Alexander Romanovsky, and Anand Tripathi, editors. *Advanced Topics in Exception Handling Techniques*. LNCS, vol. 4119. Springer, 2006.

[dLR01] Rogério de Lemos and Alexander B. Romanovsky. Exception handling in the software lifecycle. *Comput. Syst. Sci. Eng.*, 16(2):119–133, 2001.

[Don90a] Christophe Dony. Exception Handling and Object-Oriented Programming: Towards a Synthesis. *ACM SIGPLAN Notices - Proceedings of the joint conference ECOOP-OOPSLA'90*, 25(10):322–330, October 1990.

[Don90b] Christophe Dony. Improving Exception Handling with Object-Oriented Design. In *Proceedings of IEEE COMPSAC'90, Fourteenth Computer Software and Applications Conference*, pages 36–42, Chicago, USA, November 1990.

[Don01] C. Dony. A fully object-oriented exception handling system: rationale and smalltalk implementation. In Romanovsky et al. [RDKT01], chapter 2, pages 18–38.

[DTUV08] Christophe Dony, Chouki Tibermacine, Christelle Urtado, and Sylvain Vauttier. Specification of an exception handling system for a replicated agent environment. In *Proceedings of WEH '08, the 4th international workshop on Exception handling - Atlanta, Georgia*, pages 24–31. ACM, 2008.

[DUV06] Christophe Dony, Christelle Urtado, and Sylvain Vauttier. Exception handling and asynchronous active objects: Issues and proposal. In Dony et al. [DKRT06], chapter 5, pages 81–101.

[Goo75] John B. Goodenough. Exception handling: Issues and a proposed notation. *In CACM*, 18(12):683–696, 1975.

[GRRX01] Alessandro F. Garcia, Cecília M. F. Rubira, Alexander B. Romanovsky, and Jie Xu. A comparative study of exception handling mechanisms for building dependable object-oriented software. *Journal of Systems and Software*, 59(2):197–222, 2001.

[HH06] Oddleif Halvorsen and Oystein Haugen. Proposed notation for exception handling in uml 2 sequence diagrams. In *Proceedings of the Australian Software Engineering Conference*, pages 29–40, Washington, DC, USA, 2006. IEEE Computer Society.

[IB01] V. Issarny and J. Banatre. Architecture-based exception handling. In *hicss*, page 9058. Published by the IEEE Computer Society, 2001.

[Kri10] Roy Krischer. *Advanced Concepts in Asynchronous Exception Handling*. PhD thesis, University of Waterloo, Canaa, 2010.

[MDB+06] Stijn Mostinckx, Jessie Dedecker, Elisa Gonzalez Boix, Tom Van Cutsem, and Wolfgang De Meuter. Ambient-oriented exception handling. In Dony et al. [DKRT06], pages 141–160.

[MECL10] Julien Mercadal, Quentin Enard, Charles Consel, and Nicolas Loriant. A domain-specific approach to architecturing error handling in pervasive computing. In William R. Cook, Siobhán Clarke, and Martin C. Rinard, editors, *OOPSLA*, pages 47–61. ACM, 2010.

[Moo86] David A. Moon. Object-oriented programming with flavors. In *Proceedings OOPSLA '86, ACM SIGPLAN Notices*, pages 1–8, November 1986. Published as Proceedings OOPSLA '86, ACM SIGPLAN Notices, volume 21, number 11.

[Pit01] K. Pitman. Condition handling in the lisp language family. In Romanovsky et al. [RDKT01], pages 39–59.

[RDKT01] Alexander Romanovsky, Christophe Dony, Jorgen Lindskov Knudsen, and Anand Tripathi, editors. *Advances in Exception Handling Techniques*, volume 2022 of *Lecture Notes in Computer Science*. Springer, 2001.

[RDKT04] A. Romanovsky, C. Dony, J. L. Knudsen, and A. Tripathi. Workshop reader : Exception hanling, towards emerging application areas and new programming paradigms. In F. Buschmann, A. P. Buchmann, editor, *Object-Oriented Technology, ECOOP 2003 Workshop Reader*, number 3013, pages 1–10. LNCS, 2004.

[RdLFF05] Cecília M. F. Rubira, Rogério de Lemos, Gisele Rodrigues Mesquita Ferreira, and Fernando Castor Filho. Exception handling in the development of dependable component-based systems. *Softw., Pract. Exper.*, 35(3):195–236, 2005.

[RM00] M.P. Robillard and G.C. Murphy. Designing robust Java programs with exceptions. In *ACM SIGSOFT Software Engineering Notes*, volume 25, pages 2–10. ACM, 2000.

[RS03] D. Reimer and H. Srinivasan. Analyzing exception usage in large java applications. *Proceedings of the ECOOP'03 workshop on : Exception Handling in Object Oriented Systems: Towards Emerging Application Areas and New Programming Paradigms*, page 10, 2003.

[SDUV04] Frédéric Souchon, Christophe Dony, Christelle Urtado, and Sylvain Vauttier. Improving exception handling in multi-agent systems. In *Software engineering for multi-agent systems II, Research issues and practical applications*, number 2940 in LNCS, pages 167–188. Springer, 2004.

[SMKD05] Aaton Shui, Sadaf Mustafiz, Jrg Kienzle, and Christophe Dony. Exceptional Use Cases. In *Proceedings of Models/UML, ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems*, pages 568–583, Montego Bay, Jamaque, October 2005.

[TDR+11] Leonardo P. Tizzei, Marcelo Oliveira Dias, Cecília M. F. Rubira, Alessandro Garcia, and Jaejoon Lee. Components meet aspects: Assessing design stability of a software product line. *Information & Software Technology*, 53(2):121–136, 2011.

[TIRL03] F. Tartanoglu, V. Issarny, A. Romanovsky, and N. Levy. Dependability in the Web services architecture. *Architecting dependable systems*, pages 90–109, 2003.

[WB06] R.J. Wirfs-Brock. Toward exception-handling best practices and patterns. *Software, IEEE*, 23(5):11–13, 2006.