Table Ronde Patterns

Jdev 29-09-2011

Hervé Leblanc

- MCF Université Toulouse III depuis 2002.
  - Patrons enseignés en licence pro qualité du logiciel.
  - Patrons enseignés en Master ICE (Université Toulouse II) et Master Recherche (Université Toulouse III).

Hervé Leblanc

- Production scientifique :
  - Cédric Bouhours, Hervé Leblanc, Christian Percebois. *Bad smells in design and design patterns.* Dans / *In* : *Journal of Object Technology*, ETH Swiss Federal Institute of Technology, Vol. 8 N. 3, p. 43-63, mai / *may* 2009.
    - **URL** : http://www.jot.fm/issues/issue_2009_05/column5/
  - Cédric Bouhours, Hervé Leblanc, Christian Percebois. *Sharing bad practices in design to improve the use of patterns (regular paper).* Dans / *In* : *International Conference on Pattern Languages of Programs 2010 (PLoP 2010)*, *Reno, Nevada (USA)*, *16/10/2010-18/10/2010*, ACM DL, (en ligne), 2011.
  - Cédric Bouhours, Hervé Leblanc, Christian Percebois, Thierry Millan. *Detection of Generic Micro-architectures on Models (regular paper).* Dans / *In* : *International Conferences on Pervasive Patterns and Applications (PATTERNS 2010)*, *Lisbonne (Portugal)*, *21/11/2010-26/11/2010*, Ali Beklen, Jorge Ejarque, Wolfgang Gentzsch, Teemu Kanstren, Woo Lee (Eds.), IARIA, (support électronique), novembre / *november* 2010.

THÈSE

en vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

délivré par

l'Université Toulouse III – Paul Sabatier

discipline INFORMATIQUE

présentée et soutenue par

Cédric Bouhours

Détection, Explications et Restructuration de défauts de conception : les patrons abîmés.

soutenue le 4 février 2010 devant la commission d'examen :

JURY

Mireille BLAY-FORNARINO	HDR, Université de Nice-Sophia-Antipolis	*rapporteur*
Isabelle BORNE	Professeur, Université de Bretagne Sud	*rapporteur*
Hervé LEBLANC	Maître de conférences, Université Toulouse 3	*encadrant*
Christian PERCEBOIS	Professeur, Université Toulouse 3	*directeur de thèse*
Gilles RICHARD	Professeur, Université Toulouse 3	*président du jury*
Dominique RIEU	Professeur, Université Grenoble 1	*examinatrice*

École doctorale : Mathématiques, Informatique et Télécommunications de Toulouse
Unité de recherche : Institut de Recherche en Informatique de Toulouse – IRIT UMR 5505 CNRS
Équipe d'accueil : Modèles, Aspects, Composants pour des Architectures à Objets

Les Patterns : vers la transmission d'un savoir-faire

- Description d'une solution abstraite pour une famille de problèmes récurrents :
  - Idiomes de programmation (Coding Patterns).
  - Patrons de conception (Design Patterns).
  - Patrons d'analyse (Analysis Patterns).
  - Patrons d'architecture (Architectural Patterns).
  - Patrons d'organisation de projets (Organizational Patterns).
  - Patrons métiers ou produits.
- Description d'une leçon apprise :
  - Anti-Patterns.

Les Patterns : vers la transmission d'un savoir-faire

- Où les trouver ? Ouvrages de référence, sites web…
  - Idiomes de programmation (Coding Patterns).
  - Patrons de conception (Design Patterns).
  - Patrons d'analyse (Analysis Patterns).
  - Patrons d'architecture (Architectural Patterns).
  - Patrons d'organisation de projets (Organizational Patterns).
  - Patrons métiers ou produits.
  - Anti-Patterns.

Les Patterns : Outillage

- Modeleur UML Payant :

- Monde du Libre :

Les Patterns : Communauté scientifique

- PLOP et TPLOP
- Patterns
- **http://hillside.net/Patterns** : le site d'entrée sur la communauté scientifique travaillant sur les patrons de conception :
  - Documents, articles, conférences, …

The Third International Conferences on Pervasive Patterns and Applications
PATTERNS 2011
September 25-30, 2011 - Rome, Italy

Tracks:

Basics on patterns

Design patterns; Pattern identification and extraction; Validate patterns; Patterns' accuracy; Incomplete patterns; Patterns and noise

Patterns at work

Pattern logics and algebras; Pattern recognition; Pattern matching; Pattern languages; Patterns languages/models pitfalls; Pattern specification/modeling; Pattern validation; Pattern composition; Pattern reuse; Testing in pattern-based designed systems; Manageability and maintenance of pattern-based designed systems

Ubiquity patterns

User mobility patterns; Social networking patterns; Content dependency patterns; Content accessing patterns; Behavioral autonomy patterns; Prediction patterns /behavioral, structure, environment/; Patterns of discovery

Software patterns

Software design patterns; Software reuse patterns; Software quality patterns; Software testing patterns; Software performance, security, and safety patterns; Software management patterns; Patterns for evolving software elements

Security patterns

Security patterns; Patterns of trust; Attack patterns; Authorization patterns; Failed access patterns; Intrusion attempt patterns; Local malware patterns; Distributed malware patterns; Predictive patterns; Mobility patterns; Tracking patterns

System management patterns

Management and control patterns; Monitoring patterns; Correlation patterns; Event patterns; Visualization patterns

Discovery and decision patterns

Search patterns; Data mining for patterns; Query patterns; Knowledge patterns; Behavioral patterns; Reasoning patterns; Decision patterns; Patterns in WWW; Predictive patterns; Mobility patterns; Tracking patterns

Communications patterns

Communication patterns; Propagation patterns; Traffic/routing patterns; P2P and P4P patterns; Configuration change patterns; System abnormal behavior patterns

Domain-oriented patterns

Forensic patterns; Genomic patterns; Image patterns; Voice patterns; Speech patterns; Hand writing patterns; Text-embedded sentiment patterns; Emotion recognition patterns; Site access patterns; Service orchestration; Keyboard typing patterns; Financial/stock patterns; Shopping patterns; Dietary patterns; Global warming patterns; Job market patterns; Stock movement patterns; Investing patterns

Antipatterns and lessons learned

Architectural; Design; Development; People and project management; Social

Pattern Languages of Programs Conference 2011 October 21 - 23, 2011

PLoP'2011 will be co-located with SPLASH 2011, Portland, OR, USA

The purpose of PLoP is to promote development of pattern languages on all aspects of software, including design and programming, software architecture, user interface design, domain modeling, software processes, project management, and more.
Paper submissions may include short papers, containing one or more patterns, longer pattern languages or sequences, or work-in-progress papers willing to get an in-depth shepherding by an experienced pattern author at the conference.
In addition, you can submit proposals for free-format discussion groups or workshops aimed at bringing together people interested in a hot topic related to patterns or proven practices, for a period of about two hours.
Proposals addressing interdisciplinary topics and topics from other domains than software development are especially encouraged. Non-conventional formats are welcome.
Any other suggestions or proposals for activities are welcome and encouraged.
Please contact the chairs with your ideas.
PLoP11 is sponsored by the Hillside Group.
PLoP11 submissions are **peer reviewed** and **digitally archived** at ACM.
Papers discussed at writer's workshop at this conference qualify for submission to the journal "TPLoP - Transactions on Pattern Languages of Programming" published by Springer.
See Springer's pages on TPLoP for details of this journal.

Pattern Languages of Programs
Conference 2011 October 21 - 23, 2011

PLoP'2011 will be co-located with SPLASH 2011, Portland, OR, USA

Shepherding Process

The shepherding process is essentially a reviewing process. Shepherds are individuals, with experience in pattern writing, assigned to an author's paper with the expressed interest in helping the author improve the pattern. Most Shepherds also have experience with the shepherding procedure, either having been a *shepherd* before or a *sheep*(an author).

Shepherding is about improving the pattern itself, while the Shepherd maintains that the author is the one doing the pattern writing. The shepherding process is done before the paper is to be presented at a conference.

The Shepherd guides the sheep into a more mature understanding of his or her pattern.

Near the end of the shepherding, Shepherds must submit their recommendations to the Program Committee members, which then decide about its acceptance to the part of a writer's workshop of the conference.

After accepted, authors and shepherds can continue evolving the papers to produce the conference draft version.

For a more in-depth description of this reviewing process typical of PLoP's,
visit "The Language of Shepherding" (PDF) written by Neil Harrison.

All the papers submitted and accepted to be shepherded for PLoP are available for Program Committee members, shepherds, and authors.

PROCEEDINGS

The conference version of the papers will be publicly available, individually, by writers' workshop, and into the preliminary conference proceedings.

Being feedback and improvement the focus of the writers' workshops, papers are not considered final once they have been workshopped. Authors incorporate the feedback they receive at the writers' workshop into their papers before the papers go into the final proceedings to be produced after the conference, which will be made available through here.

Post-conference papers will be **digitally archived** by ACM.

Les Patterns contre la conception émergente des méthodes Agiles

Is Design Dead?

- *« For many that come briefly into contact with Extreme Programming, it seems that XP calls for the death of software design. Not just is much design activity ridiculed as "Big Up Front Design", but such design techniques as the UML, flexible frameworks, and even patterns are de-emphasized or downright ignored. In fact XP involves a lot of design, but does it in a different way than established software processes. XP has rejuvenated the notion of evolutionary design with practices that allow evolution to become a viable design strategy. It also provides new challenges and skills as designers need to learn how to do a simple design, how to use refactoring to keep a design clean, and how to use patterns in an evolutionary style. »*

- Martin Fowler
  - *May 2004:* Added sections on 'The Will to Design', 'Reversibility' and 'Is Design Happening'
  - *February 2001:* Article updated with sections on growing an architecture, the role of an architect, and where things that are difficult to add with refactoring.
  - *July 2000:* Original article submitted to XP 2000 and posted to martinfowler.com