



Erkki Laitila, CV

PhD (Information technology), Computer engineer
Systems thinking expert

UNDERSTANDING THE SOCIETY AS SYSTEMS



Vision

"GoodReason is a systems thinking methodology for meeting practical challenges of society by means of its conceptual model, a generic system architecture".¹

The mission is to provide synergy by systems thinking.

- Software eng. (1977), MsC (2005), PhD (Symbolic analysis, 2008)
- Systems thinking professional
- email: erkki.laitila@goodreason.fi & erkkilaitila@gmail.com
- p. 044 256 10 54
- born 2.2.1954
- Tempelinkatu 2 B 29, 20810 TURKU Finland
- personal interests: science philosophy, systems thinking, running

Academic skills and practical experience

- Systems thinking methodologies and teaching
- Information theory and core computer science (PhD dissertation)
- Problem solving skills: theory, heuristics and practice, soft system methodology
- Modeling techniques and conceptual analysis, ontologies and simulation
- Diagnostics: general theory and software skills. Fault isolation, reliability modeling.
- Software experience: logic programming, analysis, projects (25+ years): planning and implementing optimization systems for industry working as an IT - entrepreneur
- 8 books written about software and systems

Jobs

GoodReason Tmi	Systems thinker, method developer	1.1.2011 - ..
Systemityöyhdistys	Leader of the Agile IT group	24.1.2011 -
SwMaster Oy	Entrepreneur, CTO	18.5.1987 - 30.8.2011
Jyväskylän University	Researcher	5.10.2009 - 22.10.2009
Jyväskylän University	Post doc	10.9.2008 - 31.12.2008
Jyväskylän University	PhD student	1.1.2006 - 30.4.2008
Safematic Oy	Systems designer	19.12.1984 - 15.5.1987
DE Electronics Oy	Design manager	3.1.1983 - 16.12.1984
A.Seppänen Engineering	IT - designer	1.6.1981 - 31.12.1982
Nokia Oy	Diagnostics expert (DX 200)	4.8.1980 - 31.5.1981
Nokia Oy	Test systems programmer	7.6.1977 - 1.8.1980

Written books

1. Visual Prolog Perusteet. Teknolit. (1996)
2. Visual Prolog Teollisuuden sovellukset. Teknolit. (1996)
3. Takaisinmallintamisen hyödyntäminen ohjelmistotyössä (gradu 2005)
4. Symbolic Analysis as a Basis for a Program Comprehension Methodology (thesis 2008)
5. Symbolic Analysis as a Program Comprehension Methodology (VDM Publishing 2009)
6. Päättyteknikka ohjelmisto-ongelmien selvittämiseen (GoodReason, 2010).
7. Johdatus systeemiajatteluun (GoodReason, 2012).
8. Systeemiajattelun tekniikka tehostamassa arki ajattelua (GoodReason, 2015).

13.11.2015

In Turku, Finland

¹ www.goodreason.fi

Erkki Laitila

PUBLICATIONS from Logic Programming to Symbolic Analysis²

1. *Book about Prolog-programming in Finnish*: Visual Prolog Perusteet, Teknolit/Docendo, 1995, ISBN-952-9823-27-4 (322 p).
2. *Book about practical Prolog-programming in Finnish*: Visual Prolog – Teollisuuden sovellukset. Teknolit/Docendo, 1996, ISBN-952-9823-51-7 (228 p).
3. *In Finnish*: Logiikkaohjelmoinnin ja Prologin kehitysnäkymiä, Logiikka, Matematiikka ja Tietokone-conference, Vantaa 1996, Symposio-sarja, ISBN-951-22-3094-1.
4. Fast Production Control. Invited speak. Practical Applications of Prolog - conference, PAP/PACT. London, 1998 (<http://www.amast.org/archive/amast/mail/1998/03/msg00007.html>).
5. M.Sc-thesis: Takaisinmallintamisen hyödyntäminen ohjelmistokehityksessä, Jkl.University, 2004 (170 p).
6. Program Comprehension Theories and Prolog-Based Methodologies, Visual Prolog Language and Applications – conference (VipAlc'06), Faro, Portugal.
7. *In Finnish*: Lähdekoodin symbolinen analysointi tekoälyn näkökulmasta. SCAI-STeP- conference, Espoo, 2006, pp. 103-117 (http://www.stes.fi/scai2006/scai_step_program.pdf).
8. Program Comprehension Theories and Prolog based Methodologies, SCAI-STeP-conference, Espoo, 2006, pp. 133-142 (<http://museosuomi.cs.helsinki.fi/publications/2006/hyvonon-et-al-developments-in-artificial-intelligence-and-the-semantic-web-step-2006.pdf>).
9. Laitila, E., Legrand, S., Symbolic Atomistic Model for program comprehension. In Mexican International Conference on Artificial Intelligence 2007: (Aguascalientes, Mexico) IEEE CS Press 2008. ISBN 978-0-7695-3124-3, pp.363-372. Best Poster Award (2nd pos).
10. Symbolic Analysis and Atomistic Model as a Basis for a Program Comprehension Methodology. PhD-thesis. Jkl University, 2008. <http://www.jyu.fi/ajankohtaista/arkisto/2008/04/tiedote-2008-04-15-10-04-09-740451/>
11. Foundation for Program Understanding, SCAI- conference, Tukholma, 2008 (IOS Press).
12. Atomistic Design Pattern for Programming in Prolog. St.Petersburg 17-19.6.2008. Vip Alc'08-conference (http://download.pdc.dk/vip/vip_alc/2008/VIP-ALC08-Proceedings.pdf).
13. Symbolic Hybrid Programming Tool for Software Understanding. 3rd Int. Workshop on Hybrid Artificial Intelligence Systems. HAIS'2008, Burgos, Spain, Springer LNAI 5271, pp. 499-506.
14. Philosophy of Static, Dynamic and Symbolic Analysis, STEP'2008, Espoo, TKK, pp. 96-106.
15. SymAnalyzer: A Symbolic Analysis Tool for Program Comprehension, CSMR'2009. Kaiserslautern, Germany (to appear).
16. Symbolic Analysis from Theory to Practice (SteP-conference Espoo, August 2010).

Patents

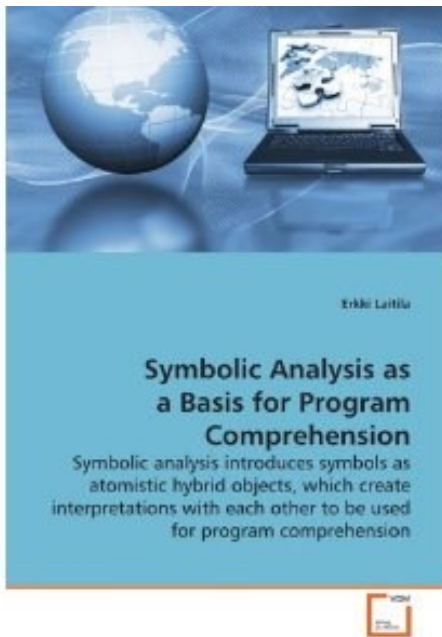
- Laitila, E. (2001). Method for developing a translator and a corresponding system. Patent: W02093371, PRH, Finland.
<http://patentscope.wipo.int/search/en/WO2002093371>

Social media

- GoodReason: www.goodreason.fi/UUTUUEDET
- Ongelmanratkaisu: www.ongelmanratkaisutekniikat.wordpress.com
- Symbolic analysis: www.symbolicanalysis.wordpress.com
- LinkedIn: <http://fi.linkedin.com/pub/erkki-laitila/8/29/175>
- Facebook (Finnish): <https://www.facebook.com/erkki.laitila>

² See: <http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/l/Laitila:Erkki.html>

Symbolic Analysis as a Basis for Program Comprehension (2009)



Research on program comprehension is important, because the amount of source code in mission-critical applications is increasing world-wide. As a solution for that challenge, a methodology with four stages is introduced. The first stage, GrammarWare, transforms the program into a symbolic form to function as a construction for the model, as the second stage, ModelWare. For the model the idea of an "atom" is presented to combine the benefits of objects for abstracting, and expressiveness of Prolog for tailoring and formalizing the model. The third stage, SimulationWare, implements a symbolic abstract machine, comparable with traditional Turing machine. With its computation model it enables symbolic analysis, which is comparable with dynamic analysis, but has a rather flexible pragmatics. The last stage, KnowledgeWare, is aimed for collecting knowledge for code inspection, error detection and verification of current operations. The methodology is programmed with Visual Prolog and implemented

in a tool, which enables the handling of Java code. The practical goal for the methodology is intelligent support for maintenance tasks.

VDM Publishing. <http://www.amazon.com/Symbolic-Analysis-Basis-Program-Comprehension/dp/363916833X/>

Päätelytekniikka ohjelmisto-ongelmien ratkaisemiseen (2010)³

Ohjelmistokehitys on tunnetusti vaativaa työtä, tiedonhaku, päättelyä, testausta sekä varmistelua ennen ja jälkeen muutosten. Tämä kirja esittelee nimenomaan ongelmien selvittelyyn tarkoitettua, kehittäjän ajattelua tukevan monitasoisen **ORT**-päättelytekniikan, joka ottaa huomioon lähes kaikki koodin piirteet.

Päätelytekniikka pohjautuu Laitilan väitöksessään luomaan symbolisen analyysin periaatteeseen, mutta sisältää myös yleisiä kognitiivisia elementtejä ongelmanratkaisun tueksi.

ORT systematisoi analysointivaiheen, mikä nopeuttaa käytännössä virheiden tunnistamista, koska se helpottaa koodin ymmärtämistä ennen muutoksia. Sille on selviä tarpeita paitsi perinteisten järjestelmien kehityksessä myös ketterässä tuotannossa, missä pitää erityisesti olla tilanteen hermolla.

Päätelytekniikka ohjelmisto-ongelmien ratkaisemiseen

Kuinka avata tuntematon tai virheellinen ohjelmakoodi



³ Lisätietoa:

Blog in English: <http://symbolicanalysis.wordpress.com/>,

Blogi ongelmanratkaisutekniikat: <http://ongelmanratkaisutekniikat.wordpress.com>

The latest book (in Finnish 2015):

A Systems Thinking Technique to Improve Outcome of Common Sense



**Kirjauutuus:
Systemit kuntoon**
Syvennä ajattelun taitojasi
Kesä 2015

**Systeemiajattelun
tekniikka
tehostamassa
arkiajattelua**

GoodReason - tekniikka
Erkki Laitila
Opettele monialaiseksi kokonaisuudeksi
Paras käsitys systeemistä!

Arkiajattelu on meille kovin tuttua kodista, kaupasta, koulusta ja työpaikoilta. Valitettavasti se ei enää riitä. Yhteiskuntamme monista kriiseistä huomaamme, että **liian yksipuolinen ajattelu** on johtamassa epärationaaliseen kilpailuun vallasta ja markkinoista ja näkyvyydestä!

Silti yhteiskunnallamme on käytössä resursseja enemmän kuin koskaan ennen. Haasteeksi tulee oivaltaa: **Kuinka saada tarpeet ja resurssit kohtaamaan toisensa koko systeemin hyvinvoinnin näkökulmasta?**

Ryhdy systeemiajattelijaksi pohtimaan kokonais kuvaa ja yhteiskunnan etua! Siinä yhdistyvät talous, politiikka, sosiologia, teknologia, työelämä ja ympäristö.

Erkki Laitila GoodReason 6.8.2015 ©

The chapters of the book are:

1. System and evaluation
2. Co-operation: cohesion
3. Society: growth
4. The reader to make influence to the ecosystem
5. Ideal future with its world view

The book has 180 pages and 180 pictures about systems.

Summary about the industrial experience

- 1977-1980 Nokia: Programming embedded test systems for Intel's 8080 processors for the whole modem production of Nokia.
- 1980-1981 Nokia: Diagnostics engineering for Nokia DX200 telephone exchange. Coordinating designers work for how to build diagnostic features to the system. Digital signalling technology and transmission protocols (PCM, HDLC). Multi processor systems 8085, 8086, 80186.
- 1981-1982 A.Seppänen Engineering Co: Programming process automation systems. HP1000 computer and process station for wood industry. A quality control system (Motorola Basic).
- 1982-1983 DE Electronics Ltd: Programming saw-automation systems by assembler. Programmable logics and roof-level industrial systems.
- 1984-1987 Safematic Ltd: System engineering for implementing a computer based condition control system. Multi-user multi-processor industrial control systems (Pascal and C and Assembler for Motorola 680x0 processors). Digital signal processing (Texas 32000) and high level planning of large multi-processor systems and real-time operating systems (OS9).
- 1986 Safematic Ltd: Building an order processing system for Safematic in Pascal.
- 1986 Skop (Bank) Corporation: Developing expert systems for bank investment consulting.
- *1987: SwMaster Ltd was founded.*
- 1987 Safematic: Developing a Basic-to-C-translator for modernizing Basic to C.
- 1987..1988 Safematic: Developing embedded lubrication automation systems in C.
- 1989..1990 Nokia: Programming a transmission control processor for a concertion ECR900 (Alcatel, AEG, and Nokia). A standalone processor to monitor BSC installations (Prolog).
- 1991..1992 Safematic: Programming drivers for data transmission in electricity networks. Implementations were embedded distributed Motorola systems using OS9 operating systems.
- 1991..1992 Enermet Ltd (later IVO/Fortum): Designing and planning optimizing systems for electricity networks for Vattenfall corporation. Unix, Oracle and embedded SQL.
- 1994: Writing two books about Prolog (published in 1996). Learning best practice and state-of-the-art of Prolog implementation throughout the world.
- 1995 M-Real Ltd: Programming an energy optimizing system for a large wood industry corporation (Prolog). High-level economical optimizing and budgeting of industrial companies.
- 1996 Jyväskylä University: Working as a teacher in Jyväskylä University, a laudatur-seminar about challenging industrial systems and Prolog.
- 1995 PKC Group: Porting large DOS applications to Windows by building a automatic porting – tool. A large cable-production system was modernized (Prolog).
- 1995..1997 PKC Ltd: Developing and maintaining a configuring and production control system. The production was delivered for manufacturing cars and trucks of Volvo, Scania etc.
- 1998..2000 PKC Ltd: Building a distributed production control system (TCP-IP). Each production machine is a client, which is connected into a server, which optimizes the work queues (Prolog).

Internet and extranet protocols and company wide networks.

- 1998 Danisco Ltd (Finnfeeds): Designing a production planning system for food industry. An optimizing system to plan month level schedules for logistics of a factory (Prolog & Excel).
- 1997..1998 ESW: A delegate of a national production systems – program (Finnish FIMET-organization). A prototype for fast production control using mass-customization (Prolog).
- 1998 ESW: Being an invited speaker, Practical Applications of Prolog and Constraints – conference, London with the title *Fast Production Control*.
- 1992..2000 ESW: Being a delegate in 10 PAP-/PACT- conferences. The topics covered practices of agent technologies, Prolog, constraint logic programming and data mining.
- 2000: A prototype for a protocol-to-C code generator to encode GPRS messages (Prolog).
- 2000..2003 SoftaGeneraattori Ltd: A company for source code analyzing was established. Since 2004 development has been continued at SwMaster.
- 2001: Patent for a automatic translator generator and the corresponding translator.
- 2004 SwMaster: Developing a symbolic method for source code analysis.
- 2004 Jyväskylä University: Master's thesis: Takaisinmallintamisen käytännön sovelluksia (Practical approaches for software reverse engineering).
- 2004 Elektrobit: Developing and programming a version control system for Nokia S60 smart phones. The code was C++ based on Symbian. Changes between different product installations of smart phones could be traced from a version to another in order to solve software problems.
- 2005 - 2008 Jyväskylä University (May): A dissertation *Symbolic Analysis and Atomistic Model as a Basis for a Program Comprehension Methodology* was finished.
- 2008 Jyväskylä University (September-December): Project coordinator for a Linux mobile software community, Jyväskylä region.
- 2009 Writing a book *Symbolic Analysis as a Basis for Program Comprehension*.
- 2009-2010 Writing a book : Päätelytekniikka ohjelmisto-ongelmien ratkaisemiseen (Logic for solving software problems).
- 2010: Consulting projects for analyzing source code.
- 2011 Writing a book: Johdatus systeemiajatteluun – GoodReason tekniikka. (Introduction to systems thinking – GoodReason technology).
- 2012-2013 Courses about systems thinking. Public sector: Åbo Akademi, Jyväskylä University, Tampere University. Courses for private sector.
- 2014 Working as a lecture in Jyväskylä University: Systems thinking theory and practices (<http://goodreason.fi/systeemiajattelukurssi-jyvaskylan-yliopistossa/>).
- 2014 Several new systems thinking courses planned.
- 2015 The book *Systeemiajattelun tekniikka tehostamassa arki ajattelua* is published.