Deliverable D31: Final Workshop report

Report Preparation Date: March 2009
Classification: Public

Contract Start Date: 1st January 2006
Contract Duration: 39 months
Project Co-ordinator: LAAS-CNRS
Partners: Budapest University of Technology and Economics
          City University, London
          Technische Universität Darmstadt
          Deep Blue Srl
          Institut Eurécom
          France Telecom Recherche et Développement
          IBM Research GmbH
          Université de Rennes 1 – IRISA
          Université de Toulouse III – IRIT
          Vytautas Magnus University, Kaunas
          Fundação da Faculdade de Ciencias da Universidade de Lisboa
          University of Newcastle upon Tyne
          Università di Pisa
          QinetiQ Limited
          Università degli studi di Roma "La Sapienza"
          Universität Ulm
          University of Southampton
Contents

1- Introduction .................................................................................................................................. 3

2- Programme ................................................................................................................................... 5

3- Attendance List .......................................................................................................................... 11

4- Slides .......................................................................................................................................... 15
   Workshop Introduction, Jean-Claude Laprie (LAAS-CNRS, Toulouse, France) ....................... 17
   Training and Dissemination, Luca Simoncini (Pisa University, Pisa, Italy) ......................... 19
   Ontologies, Al Avizienis and Gintare Grigonyte (Vyttautas Magnus University, Kaunas, Lithuania), Thorsten Liebig (University of Ulm) ......................................................... 27
   Resilience-Explicit Computing, Tom Anderson (University of Newcastle upon Tyne, UK)... 43
   Resilience Knowledge Base, Hugh Glaser (University of Southampton, UK) ....................... 53
   Research Agenda, Jean-Claude Laprie (LAAS-CNRS, Toulouse, France) ............................ 71
   Integration, Karama Kanoun (LAAS-CNRS, Toulouse, France) ............................................ 77
   Honeypots: Malicious fault characterization exploiting honeypot data, Corrado Leita (Symantec Research Lab, Sophia-Antipolis, France) ......................................................... 79
   AROVE-v: Assessing the resilience of open verifiable e-voting systems, Eugenio Alberdi (City University, London, UK) .......................................................... 81
   ASAP: Assessment-based adaptable software architecture for dependability, Thomas Robert (LAAS-CNRS, Toulouse, France) ...................................................... 87
   FADA: Formalisms and algorithms for resilient services design in ambient systems, Matthieu Roy (LAAS-CNRS, Toulouse, France) .................................................. 97
   FAERUS: Formal analysis of evolving resilient usable systems, Mieke Massink (University of Pisa, Italy) .......................................................... 107
   FOREVER: Fault/intrusion Removal through evolution and recovery, Paulo Sousa (University of Lisbon) .......................................................... 125
   RAPTOR: Multi-agent systems with fault-tolerant agreement protocols for conflict resolution in air traffic control, Henrique Moniz (University of Lisbon) ...................... 143
   TMS: Testing in mobile settings, Hélène Waeselynck (LAAS-CNRS, Toulouse, France) .... 155
   WSNA: Formal modelling and analysis methods for wireless sensor network algorithms, Holger Pfeifer (University of Ulm, Germany) .................................................. 167
1- Introduction

The workshop was held at LAAS, on 12-13 March 2009. It was aimed at presenting the results of ReSIST.

The workshop was attended by 63 persons.

The remainder of this report gives:

1) The workshop programme.
2) The attendance list.
3) The copies of the slides presented during the workshop.
2- Programme
The challenges raised for achieving satisfactorily dependability and security of the emerging ubiquitous systems are sharpened by the statistical evidence that those systems suffer from a gap in the achieved capabilities with respect to the expectations of the stakeholders.

A central characteristic of those ubiquitous systems being the continuous evolutionary changes they are facing, scaling up their dependability and security requests a resilience view in order to cope with and to adapt to these evolutionary changes. The changes can be functional, technological, environmental, and include threat evolutions. Such changes drastically increase uncertainty about system and infrastructure behaviour.

The workshop is aimed at presenting the results and the findings of the European Network of Excellence ReSIST for resilience of computing systems and information infrastructures to enable their dependability and security to scale-up.
Workshop Schedule

Thursday 12 March

8h - 9h  Registration and welcome coffee

9h - 9h25  Workshop Introduction, Jean-Claude Laprie (LAAS-CNRS, Toulouse, France)

9h25 - 10h05  Training and Dissemination, Luca Simoncini (University of Pisa, Italy)

10h05 - 10h45  Ontologies, Al Avizienis and Gintare Grigonyte (Vytautas Magnus University, Kaunas, Lithuania), Thorsten Liebig (University of Ulm)

10h45 - 11h15  Coffee Break

11h15 - 12h30  Mini-projects 1

11h15 - 11h40  Honeypots: Malicious fault characterization exploiting honeypot data, Corrado Leita (Symantec Research Lab, Sophia-Antipolis, France)

11h40 - 12h05  AROVE-v: Assessing the resilience of open verifiable e-voting systems, Eugenio Alberdi (City University, London, UK)

12h05 - 12h30  ASAP: Assessment-based adaptable software architecture for dependability, Thomas Robert (LAAS-CNRS, Toulouse, France)

12h30 - 13h30  Lunch

13h30 - 14h10  Resilience-Explicit Computing, Tom Anderson (University of Newcastle upon Tyne, UK)

14h10 - 14h50  Resilience Knowledge Base, Hugh Glaser (University of Southampton, UK)

14h50 - 15h20  Coffee Break

15h20 - 17h20  Mini-projects 2

15h20 - 15h45  FADA: Formalisms and algorithms for resilient services design in ambient systems, Matthieu Roy (LAAS-CNRS, Toulouse, France)

15h45 - 16h10  FAERUS: Formal analysis of evolving resilient usable systems, Mieke Massink (University of Pisa, Italy)

16h10 - 16h35  FOREVER: Fault/intrusion removal through evolution and recovery, Paula Sousa (University of Lisbon)

16h35 - 17h  RAPTOR: Multi-agent systems with fault-tolerant agreement protocols for conflict resolution in air traffic control, Henrique Moniz (University of Lisbon)

Friday 13 March

8h30 - 9h  Coffee

9h - 9h50  Mini-projects 3

9h - 9h25  TMS: Testing in mobile settings, Hélène Waeselynck (LAAS-CNRS, Toulouse, France)

9h25 - 9h50  WSNA: Formal modelling and analysis methods for wireless sensor network algorithms, Holger Pfeifer (University of Ulm, Germany)

9h50 - 10h15  Research Agenda, Jean-Claude Laprie (LAAS-CNRS, Toulouse, France)

10h15 - 10h40  Integration, Karama Kanoun (LAAS-CNRS, Toulouse, France)

10h40 - 11h10  Coffee Break

11h10 - 12h30  Panel and conclusion

12h30 - 13h30  Lunch

Workshop registration

Registration to the workshop is free of charge. Advance registration for attendees not members of ReSIST is requested for logistics purposes, using the registration form at the end of the programme. Coffee breaks, lunches and Thursday dinner are part of the workshop attendance.

Workshop Location and how to reach it

http://www2.laas.fr/laas/2-4275-How-to-access-to-LAAS.php

A chartered bus will take attendees to LAAS on Thursday and Friday morning, departing at 8h00 from 28 Allée Jean-Jaurès, in front of the Flunch restaurant (map at the end).

Hotels

About ReSIST

ReSIST is a Network of Excellence that addresses the strategic objective “Towards a global dependability and security framework” of the European Union Work Programme, and responds to the stated “need for resilience, self-healing, dynamic content and volatile environments”.

It integrates leading researchers active in the multidisciplinary domains of Dependability, Security, and Human Factors, in order that Europe will have a well-focused coherent set of research activities aimed at ensuring that future “ubiquitous computing systems” – the immense systems of ever-evolving networks of computers and mobile devices which are needed to support and provide Ambient Intelligence (AmI) – have the necessary resilience and survivability, despite any physical and residual development faults, interaction mistakes, or malicious attacks and disruptions.

Network Partners

LAAS-CNRS, Toulouse, France (Coordinator)
Budapest University of Technology and Economics, Hungary
City University, London, UK
Technische Universität Darmstadt, Germany
Deep Blue Srl, Roma, Italy
IBM Research, Zurich, Switzerland
Institut Eurêcom, Sophia Antipolis, France
France Telecom Recherche et Développement, Lannion and Caen, France
Université de Rennes 1 – IRISA, France
Université de Toulouse III – IRIT, France
Vytautas Magnus University, Kaunas, Lithuania
Escola Politécnica de Lisboa, Portugal
University of Newcastle upon Tyne, UK
Università di Pisa, Italy
QinetiQ Ltd, Malvern, UK
Università degli studi di Roma “La Sapienza”, Italy
Universität Ulm, Germany
University of Southampton, UK

ReSIST Final Workshop
LAAS-CNRS, Toulouse

Registration Form

Fax to +33 (0)5 61 33 64 11 or e-mail the requested information to resistmeeting@laas.fr

Attendee:

Name (First Last):

Email:

Company/Institution:

Address:

Phone:

Special Dietary Needs:
3- Attendance List
ADNAN, Noor Miam, Roma University
AKROUT, Rym, LAAS-CNRS
ALBERDI, Eugenio, City University
ANDERSON, Tom, Newcastle Upon Tyne University
ANDREWS, Zoe, Newcastle Upon Tyne University
ARLAT, Jean, LAAS-CNRS
AVIZIENIS, Al, Vytautas Magnus University
BANÂTRE, Michel, IRISA
BARBONI, Eric, IRIT
BASNYAT, Sandra, CNES
BONELLI, Stefano, Deep Blue
BONNET, François, IRISA
BONOMI, Silvia, Roma University
CORREIA, Miguel, Lisbon University
CROUZET, Yves, LAAS-CNRS
DACIER, Marc, Symantec
DAMBRA, Carlo, Pisa University
DOMENICI, Andrea, Pisa University
FABRE, Jean-Charles, LAAS-CNRS
GASHI, Ilir, City University
GLASER, Hugh, Southampton University
GRIGONYTE, Gintare, Vytautas Magnus University
HARRISON, Michael, Newcastle Upon Tyne University
KAÂNICHE, Mohamed, LAAS-CNRS
KANOUN, Karama, LAAS-CNRS
LAC, Chidung, FT
LADRY, Jean-François, IRIT
LAPRIE, Jean-Claude, LAAS-CNRS
LEITA, Corrado, Symantec
LIEBIG, Thorsten, Ulm University
MASCI, Paolo, Pisa University
MASSINK, Mieke, Pisa University
MILLARD, Ian, Southampton University
MONIZ, Henrique, Lisbon University
MORGANTI, Michel, Fondazione Politecnico Turino
NAVARRE, David, IRIT
NGUYEN, Minh Duc, LAAS-CNRS
4- Slides
ReSIST
Resilience for Survivability in IST
A European Network of Excellence

Final workshop — Introduction

Jean-Claude Laprie

Thursday, 12 March
9h: Coffee
9h30: Introduction
10h: Training & Dissemination
10h30: Ontologies
11h: Coffee break
11h30: 3 mini-projects: Honeypots, AROVE-v, ASAP
12h: Lunch
12h30: ResEx
13h: RKB
13h30: Coffee break
14h: 4 mini-projects: FADAS, FAERUS, FOREVER, Multi-Agent
14h30: Panel and Conclusion
15h: Coffee break
15h30: Integration
16h: Coffee break
16h30: Lunch
17h: Research Agenda

Friday, 13 March
9h: Coffee
9h30: 2 mini-projects: TMS, WSNA
10h: Research Agenda
10h30: Integration
11h: Coffee break
11h30: Panel and Conclusion
12h: Lunch
12h30: Integration
13h: Coffee break
13h30: Lunch
14h: Research Agenda
14h30: Integration
15h: Coffee break
15h30: Panel and Conclusion
16h: Lunch
16h30: Research Agenda
17h: Integration

40 mins slots → 30 mins presentation + 10 mins discussion
25 mins slots → 20 mins presentation + 5 mins discussion
<table>
<thead>
<tr>
<th>Del. #</th>
<th>Deliverable name</th>
<th>WP No</th>
<th>Lead partner</th>
<th>Delivery date</th>
</tr>
</thead>
<tbody>
<tr>
<td>D28</td>
<td>Periodic activity report</td>
<td>WP0</td>
<td>LAAS</td>
<td>M38</td>
</tr>
<tr>
<td>D29</td>
<td>Periodic management report</td>
<td></td>
<td>LAAS</td>
<td>M38</td>
</tr>
<tr>
<td>D30</td>
<td>Periodic report on the distribution of the Community's contribution</td>
<td>WP0</td>
<td>LAAS</td>
<td>M38</td>
</tr>
<tr>
<td>D31</td>
<td>Final Workshop report</td>
<td></td>
<td>LAAS</td>
<td>M39</td>
</tr>
<tr>
<td>D32</td>
<td>Resilience Knowledge Base: final</td>
<td>WP1</td>
<td>Newcastle</td>
<td>M36</td>
</tr>
<tr>
<td>D33</td>
<td>Resilience-Explicit Computing: final</td>
<td>WP1</td>
<td>Newcastle</td>
<td>M36</td>
</tr>
<tr>
<td>D34</td>
<td>Resilience Ontology: final</td>
<td>WP1</td>
<td>Newcastle</td>
<td>M36</td>
</tr>
<tr>
<td>D35</td>
<td>Resilience scaling technologies: results</td>
<td>WP2</td>
<td>IRISA</td>
<td>M36</td>
</tr>
<tr>
<td>D36</td>
<td>International survey on research challenges in resilience</td>
<td>WP2</td>
<td>IRISA</td>
<td>M36</td>
</tr>
<tr>
<td>D37</td>
<td>Resilient Computing Curriculum</td>
<td>WP3</td>
<td>Pisa</td>
<td>M36</td>
</tr>
<tr>
<td>D38</td>
<td>Resilient Computing Courseware</td>
<td>WP3</td>
<td>Pisa</td>
<td>M36</td>
</tr>
<tr>
<td>D39</td>
<td>Selected Current Practices document</td>
<td>WP3</td>
<td>Pisa</td>
<td>M36</td>
</tr>
<tr>
<td>D40</td>
<td>Legacy: resilience knowledge base and courseware</td>
<td>WP1 &amp; 3</td>
<td>Newcastle and Pisa</td>
<td>M36</td>
</tr>
</tbody>
</table>

**Relationship Activities - Objectives**

- **Integration Operations**
  - Open workshops
  - Plenary meetings
  - Personnel exchange and co-advised theses
- **Resilience Integration Technologies**
  - Resilience knowledge base
  - Resilience-explicit computing
  - Resilience ontologies
- **Resilience Building and Scaling Technologies**
  - State of knowledge
  - Research agenda
  - Mini-projects
  - Student seminar
  - Summer school
- **Training**
  - Curriculum
  - Courseware
  - Publications and presentations
  - Selected practice document

**Objectives**

- Integration of teams of researchers
- Identification of research directions
- Production of research results
- Promotion of resilience culture
WP3 - Training and Dissemination

Task TD-T3 MSc Resilient computing curriculum and syllabi preparation (D37 due M36)

- MSc Curriculum in Resilient Computing completed on time; D37 already delivered
- Curriculum has been weighted in terms of student loads with the relation to the ECTS system (120 ECTS x 25 hours = 3000 hours - 1000 h lectures and labs + 2000 hours individual study)
- Curriculum presented to:
  - DSN’07, Edinburgh, UK in June 2007 and 52nd IFIP W.G. 10.4
  - European Computer Science Summit, Berlin, Germany in September 2007
  - 53rd IFIP W.G. 10.4 Natal, Brazil in February 2008
  - EDCC-7, Kaunas, Lithuania in May 2008 in a Special Session
  - DSN’08, Anchorage, Alaska in June 2008, in a Special Session and 54th IFIP W.G. 10.4
- All on-line at http://www.resist-noe.org/
Task TD-T3 MSc Resilient computing curriculum and syllabi preparation
(D37 due M36)

- the activity on the MSc Curriculum will continue after the end of ReSIST, through dissemination to European Universities, and maintaining the site and RKB.

- a Steering Committee has been identified to assure the legacy of the Curriculum and related Courseware, composed by: Tom Anderson, Algirdas Avižienis, Hugh Glaser, Jean-Claude Laprie, Brian Randell, Luca Simoncini

Task TD-T4 Courseware preparation (D38 due M36)

- Courseware for Resilient Computing completed on time; D38 already delivered
- All lines of teaching for each course has been reviewed and updated
- Original ReSIST Courseware, as set of slides, for the following Courses:
  - Fundamentals of Dependability - J-C. Laprie
  - Computer Network Security - P. Verissimo, M. Correia
  - Resilient Distributed Systems and Algorithms - P. Verissimo, M. Correia
  - Dependability and Security Evaluation of Computer-based Systems - M. Kaâniche, K. Kanoun, J-C. Laprie
  - Usability and User Centred Design for Dependable and Usable Socio-technical Systems - P. Palanque, M. Harrison, M. Winckler
  - Management of Projects - G. Lami
  - Middleware Infrastructures for Application Integration - R. Baldoni, R. Beraldì, G. Lodi, L. Querzoni, S. Scipioni
  - Software Reliability Engineering - K. Kanoun

- A very extensive search for support material has been made on the web
- Integrated into the RKB
- All on-line at http://www.resist-noe.org/
Support material from:

- LAAS-CNRS, France
- Budapest University of Technology and Economics, Hungary
- City University, London, UK
- Technische Universität Darmstadt, Germany
- Institut Eurécom, France
- France Telecom Recherche et Développement, France
- IBM Research GmbH, Switzerland
- Université de Rennes 1 – IRISA, France
- Université de Toulouse III – IRIT, France
- Fundação da Faculdade de Ciências da Universidade de Lisboa, Portugal
- University of Newcastle upon Tyne, UK
- Università di Pisa, Italy
- Università degli studi di Roma "La Sapienza", Italy
- Universität Ulm, Germany

- Aalborg University, Denmark
- Adelard, UK
- Carleton University, Canada
- Carnegie Mellon University, USA
- Chalmers University, Sweden
- Chinese University of Hong Kong, China
- CSR, London, UK
- Duke University, USA
- EPFL, Switzerland
- ETH Zurich, Switzerland
- EWICS TC7
- George Mason University, USA
- Georgia Institute of Technology, USA
- Queen Mary University, London, UK
- Katholieke Universiteit Leuven, Belgium
- Imperial College, London, UK
- Lehigh University, USA
- MIT, USA
- Saarland University, Germany
- Scuola Superiore S. Anna, Pisa, Italy
- Technical University of Madrid, Spain
- University College London, UK
- University of Aachen, Germany
- University of Bielefeld, Germany
- University of Birmingham, UK
- University of Bristol, UK
- University of California at Berkeley, USA
- University of Cambridge, UK
- University of Copenhagen, Denmark
- University of Edinburgh, UK
- University of Glasgow, UK
- University of Konstanz, Germany
- University of Melbourne, Australia
- University of Pennsylvania, USA
- University of Southern California, USA
- University of Texas at San Antonio, USA
- University of Twente, Netherlands
- University of Waterloo, Canada
- University of Yale, USA
- Weizmann Institute of Science, Israel
- Westminster College, USA

This effort has produced the first version of a comprehensive database of support material on Resilient and Dependable Computing, whose relevance and interest for the community will be maintained after the end of ReSIST.

All lines of teaching for the Courses in the MSc Curriculum, the original ReSIST set of slides, the links to the additional support material, and the links to the relevant sites are on the ReSIST web-site [http://www.resist-noe.org/](http://www.resist-noe.org/) and all material can be viewed and/or downloaded for educational purposes.
Task TD-T5: Dissemination program (D28-PAR due M36)

Papers and events:

- 134 papers from the work performed within ReSIST (ReSIST papers) of which 37 papers multi-site authored
- 93 papers related to ReSIST topics, 5 multi-site authored
- 100 events have been attended by ReSIST persons with presentation of ReSIST itself or of work achieved within ReSIST

Liaison with other European projects in the fields of dependability and security:

- The following EU Projects ADVISES, CRUTIAL, DESEREC, HIDENETS, Mobius, RODIN, SERENITY and UbiSec&Sens have been maintained informed of the activities done in ReSIST.

Curriculum presented to:

- DSN’07, Edinburgh, UK in June 2007 and 52nd IFIP W.G. 10.4
- European Computer Science Summit, Berlin, Germany in September 2007
- 53rd IFIP W.G. 10.4 Natal, Brazil in February 2008
- EDCC-7, Kaunas, Lithuania in May 2008 in a Special Session
- DSN’08, Anchorage, Alaska in June 2008, in a Special Session and 54th IFIP W.G. 10.4

ReSIST presented to:

- EDCC-7, Kaunas 7-9 May 2008
- DSN 2008, Anchorage 24-27 June 2008, USA

Joint European WS on “Human Factors in Education & Training for Safety” co-organized with EWICS TC7, NHS and Warwick Medical School, April 8, 2008, Warwick, UK
Task TD-T5: Dissemination program (D28-PAR due M36)

- Work done on Selected Current Practices
  - survey of the resilience definitions in different industrial domains:
    - ICT, critical infrastructures, industrial safety, air traffic management, resilience engineering, organisation management, financial services and seismic engineering
  - survey of the existing standards and best practices (118 entries) related to the different aspects of resilience in the different industrial domains:
    - aeronautics, Air Traffic Management, automotive, critical infrastructures, e-Services, industrial control, nuclear power plants, railway, resilient ICT systems (i.e. grouping all generic standards), space, telecommunications

- Work done on Selected Current Practices (cont.)
  - organisation of 2 workshops (Roma in 2007, Bristol in 2008) to discuss with industrialists of different domains their view of resilience in ICT
  - synthesis of the significant workshops’ outcome into 7 papers covering industrial current practices
  - publication of D39 deliverable with cross-links with D13 “From Resilience-Building to Resilience-Scaling Technologies: Directions”
Task TD-T5: Dissemination program (D28-PAR due M36)

• Papers
  – Current practices in resilient computing: public communications domain” by M. Morganti
  – “NHS Connecting for Health: Growing a sound resilience approach” by I. Harrison
  – “Resilience of Automotive Engine Management Systems (EMS)” by D. Claraz
  – “Resilience in the avionics domain: a pilot view” by A. Chialastri
  – “Resilience in Instrumentation & Control of Nuclear Power plants” by A. Lindner
  – “An Operational View of Security” by J. Riordan

• ReSIST heritage: a book on current industrial practices
  – the book will include:
    • the 7 papers published in D39
    • possibly, some extra contributions from selected authors (to be confirmed):
      – Pierre Chartier, mass transit
      – Michael Behringe, security and complexity in networks
      – David Embrey, process and power generation
  – selected publisher
    • Ashgate Studies in Resilience Engineering
Task TD-T5: Dissemination program (D28-PAR due M36)

• Some conclusions
  – While in the research arena the resilience concept is widely developed and studied, industry is starting the first steps towards the adoption of the resilience concepts. This is demonstrated by the few initiatives already in place mostly concentrated on the e-services (banking, large databases, etc.) and communication sectors.
  – Standardisation world is still concentrated on single aspects of the resilience concept (dependability, availability, security, etc.), with few remarkable exceptions; see for example
    • the standard BS2599 on business continuity management
    • the guidelines published by the Centre for the Protection of the National Infrastructure on telecommunications and virtual server implementation

• Some conclusions (cont.)
  – Without pretending to be exhaustive, it comes out from the industrial presentation that the four resilience scaling technologies (Evolvability, Assessability, Usability, Diversity) are unevenly considered in the different industrial domains.
    • In particular Evolvability is definitively the most important issue across the different domains, with the noticeable exception of the nuclear domain, reflecting the increasing dynamicity of modern industrial systems.
    • Usability seems not to be focused directly being mainly seen as a different perspective (the operators’ one) on Assessability, and few of the industrial contributors take this perspective.
  – The role of human factors in resilience appears to be a hot topic for managers of critical infrastructures.
A Question

- Dependability and Security
- Trustworthiness
- Survivability
- High Confidence
- Information Assurance
- Robustness
- Resilience
- Self – Healing

How do they differ?
A Search for Consensus

1982: Special session at FTCS-12: several concept papers

The Representation Problem

Multiple near-synonymous terms exist
Disadvantages that impair progress:
- Continuing re-invention
- Plagiarism
- Confusion among potential users
- Difficulties for referees and evaluators

The Need: a single thesaurus and ontology of dependable and secure computing
Sad Conclusion: a committee of volunteers or bureaucrats cannot do it!
A Potential Solution

Apply computer tools for human language processing

- Extract **term candidates** from a set of texts
- Build a **thesaurus**: list of important terms and related terms for each entry of the list
- Build an **ontology**: data model that represents the thesaurus
- Perform **automatic classification** of texts using automatic indexation and clustering tools

Forthcoming Publication

The Problem is Common for All of Computer Science & Engineering

- The only taxonomy of Computer S&E is the ACM CSS (Computing Classification System) devised in 1988, revised in 1998
- Dependability and security are inadequately treated in the ACM CSS
- **The Challenge:** a major revision of the ACM CSS is being initiated, therefore our thesaurus and ontology must be ready

An “Info-Skeptic” view

- Physical sciences study nature: given phenomena
- Computer S&E study information: human-made concepts
- The concepts should compete, and the fittest will survive!
- If a good concept disappears, it will reappear again, with some luck… in my research
Original Goals

(1) Fill the gap between knowledge (documents) and structured representations of their content (ontologies) in the domain of resilience by using NL tools to create and extend thesaurus and ontology.

(2) NL tool-chain to conduct document classification experiments in order to classify existing resilience literature.

Starting Points

- Document corpora
  - Compendium of FTCS/DSN conferences:
    ~830 papers presented at 9 International Conferences on Dependable Systems and Networks (2000-2008)

- Tools
  - MPRO, AUTOTERM, AUTINDEX
  - OntoTrack

- Resilience ontology
  - IEEE Avižienis, Laprie, Randell, Landwehr paper
  - OWL ontology file
Idea

- Document clusters will be represented by “clouds of thesaurus terms”
- Resilience-relevant thesaurus terms need to be linked with ontology concepts.
- Clusters will map (via their terms) into different aspects of ontology (failures, attributes of secure systems, methods to prevent faults, etc.)
- The “link structure” will tell something about the content (which aspects at which granularity)
- Experts should be able to name typical mappings.

Conceptual Architecture
The scope: Automatic extraction of lexical elements (entities) for building the thesaurus

How we got there: the process of building thesaurus

- 2830 documents
- 234,585 tokens

- Thesaurus contains 7974 terms
Circuit techniques are used to make sections of the design robust to non-delay faults. The combination of these is an asynchronous defect-tolerant circuit where a large class of faults are tolerated, and the remaining faults can be both detected easily and isolated to a small region of the design.

Different levels of linguistic processing:
- Rule based morphological analysis
- Syntactical disambiguation and tagging
- Terminology extraction techniques
Terms and … a problem

circuit techniques
non-delay faults
combination
asynchronous defect-tolerant circuit
large class
fault
remaining fault
small region
design

Solution

How we define which terms are domain specific?
- not too general
- not too “specialised”

Apply term informativity measure: MI, Log-likelihood, Jacquard's coefficient, etc.

- IDF measure: \( idf(t) = \log \left( \frac{|D|}{|\{d : t \in d\}|} \right) \)
- Obtaining IDF values and defining a certain threshold helped to prune the term list from 9,012 terms down to 7,974
The expert part: final evaluation

Term annotation system:

<table>
<thead>
<tr>
<th>Terms to annotate</th>
<th>General term</th>
<th>Vague term</th>
<th>Non-term</th>
<th>Computer science &amp; Engineering</th>
<th>Dependability</th>
<th>Security</th>
<th>D&amp;S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1. system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2. dependable system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3. fault-tolerant system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.4. real-time system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5. tolerant system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.6. safety-critical system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.7. asynchronous system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.8. critical system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.9. digital system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Document clustering

- **Paper:** Joint Evaluation of Performance and Robustness of a COTS DBMS through Fault-Injection. Diamantino Costa, Tiago Rilho and Henrique Madeira.

- **Descriptors:** data banks [100]; performance evaluation [46]; operating systems [40]; research [38]; benchmarks [29]; computer program [29]; emulations [28]; business process [28]; target system [27]; hangings [21];
The Resilience (ALRL) Ontology

- Based on Avižienis, Laprie, Randell, Landwehr paper
  IEEE Trans. on Dep. and Sec. Computing. 2004
  - OWL version from B. Randell 11/2006
    (plus mapping of ACM terms to ontology concepts)
  - Contains 180 concepts / expressivity of ALRL (RDFS)

- Discussion (Newcastle – Southampton) 09/2007 about classification scheme issues
  - “Self-checking component is not a kind of Error detection it is a concept which is related to Error detection in some way.”

- Revision/evolution of the ALRL ontology
  o make knowledge available for non-domain experts
  o make knowledge accessible for reasoning services
ALRL Fault (as of Ontology)

- Failure
  - Error
  - Fault
    - Fault-Persistence
    - Fault-Dimension
    - Configuration-Fault
    - Phenomenological-Cause-Of-Fault
    - Fault-Phase
      - Fault-Capability
      - Interaction-Fault
      - Fault-Objective
      - Reconfiguration-Fault
      - Fault-Intent
    - Physical-Fault
    - System-Boundaries
  - Permanent-Fault
  - Transient-Fault
  - Hardware-Fault
  - Software-Fault
  - Natural-Fault
  - Human-Made-Fault
  - Operational-Fault
  - Development-Fault
  - Incompetence-Fault
  - Accidental-Fault
  - Non-Malicious-Fault
  - Malicious-Fault
  - Non-Deliberate-Fault
  - Deliberate-Fault
  - Internal-Fault
  - External-Fault
  - Production-Defect
  - Omission-Fault
  - Commission-Fault
  - Trojan-Horse
  - Trapdoor
  - Zombie
  - Logic-Bomb
  - Virus
  - Worm

ALRL Fault Categories (as of Paper)

- Development Faults
  - Operational Faults
  - Internal Faults
  - External Faults

- Physical Faults

- Interaction Faults

All Development Faults are Internal Faults as well as Permanent Faults.

All External Faults are Operational Faults.
Revised Fault Sub-Hierarchy

ALRL Fault Categories (as of Paper)
Fault No. 6 (Logic Bomb)

Conceptual Architecture (Mapping)
Thesaurus – ALRL Mapping

- Bi-directional mapping between
  1) set of thesaurus terms (≈ 8000)
  2) meaningful organized concepts (≈ 180)

- Tasks:
  - Discard non-relevant terms from thesaurus
  - Introduce term synonyms
  - Create term-concept links
  - Add thesaurus terms to ontology

ReSIST Ontology Mapping Plugin

- Plugin for ontology workbench OntoTrack
  - Loads ALRL ontology, thesaurus, RKB data (fragment)
  - Manipulation of ALRL as well as thesaurus
  - Graphical bi-directional mapping via drag-and-drop operations (ALRL ⇔ thesaurus)
  - Semi-automatic mapping (via syntactical match)
  - XML-based export of mapping for further processing
  - Ontological paper annotation via mapping:
    - Import of RKB data with given descriptors
    - On-the-fly paper classification via IAI descriptor service
Conclusion

- Project work combines:
  - NLP based analysis of resilience documents
  - Structured knowledge of the domain of resilient computing

- Results:
  - Set of domain terms (thesaurus) and document clusters
  - Resilience ontology (makes resilience knowledge explicitly available for (non-)domain experts)
  - Tool chain for document annotation and selection

Outlook

- Application scenarios:
  - Automatically assigning annotated submissions to reviewers
  - Identification of related publications
  - Intelligent search in large document sets
  - Mediation between different dialects (near-synonym term problem)

- Continuation of effort
  - Forming IFIP Special Interest Group
  - Expanding scope of ontology to all of Informatics (Computer Science and Engineering)
Resilience-Explicit Computing
ResEx

Tom Anderson
Newcastle University

Work Package 1

WP1 “Integration Technologies”
Objective: to lay foundations for facilities to assist engineers in selecting and deploying resilience mechanisms and tools
• at design time
• and dynamically (during system operation and evolution)
WP1 Tasks

- IT-T1: developing a Resilience Knowledge Base (RKB) – a body of knowledge on resilience concepts, methods and tools
- IT-T2: on Resilience-Explicit Computing (ResEx) – making resilience information (metadata) explicit
- IT-T3: developing a Resilience Thesaurus and Ontology (ResOn) – to be utilised by ResEx and the RKB

WP1 Organisation
ResEx Basics

• Explicit resilience-related information (metadata)
• Support for design-time and run-time decision-making
• Requires description of resilience design patterns and tools (“mechanisms”) in terms of metadata

ResEx Objectives

• To set up a means of gathering resilience mechanism descriptions in terms of metadata
• To establish a catalogue of mechanisms in the RKB
• To encourage exploitation of resilience-related metadata in selecting mechanisms
• To explore research issues and challenges
Status at end of 2007

- 12 “first edition” Resilience Mechanisms characterised in the RKB
  - documented in deliverable (D11)
- New candidate mechanisms identified
  - acquisition policy agreed
- RKB extensions to accommodate mechanism descriptions
  - linked to ontologies
- Improved RKB interfaces for Adding/Viewing mechanisms

ResEx Goals for 2008

- Populate RKB with an extended mechanism set
  - More mechanisms
  - Increased coverage
- Identify and explore Challenge Problems
  - Workshops
  - ResEx Grid Computing; ResEx Security; ResEx Ambient

Both goals support a longer-term strategy
  - Increased utility, better understanding, so as to promote future use of ResEx, and of ResEx elements within the RKB
More Mechanisms

*Work is still ongoing, so these numbers will increase.*

- Detailed descriptions of 24 mechanisms
  - RKB template complete; reviewed and revised descriptions
- Partial descriptions of 14 mechanisms
  - Some fields in the template are incomplete
- Outline descriptions of 120 mechanisms
  - A brief overview, but with links to external descriptions

Thus the RKB now contains a total of 158 mechanisms!

Detailed Descriptions

- Process “mechanisms”
  - Robust re-encryption mixes; Ad-hoc routing in resilient ambient systems; Heuristic evaluation
- Tools
  - Model based stochastic dependability evaluation; Robustness testing; Modelworks; CLawZ; Malporte
Detailed Descriptions

• Architectural “mechanisms”
  – Consensus Mechanism; Dynamic Function Allocation; N-Self-Checking Programming/1/1; N-Version Programming/1/1; Recovery Blocks/1/1; Supervisory Systems; Cooperative Backup; Autonomic Computing Architecture; Byzantine quorum systems; CRIA - Critical Interaction Analysis Method; Dynamic Function Allocation (adaptive automation); Patterns of cooperative interaction; Self-healing for Wireless Sensor Networks; State machine replication; Trust and Cooperation Oracle; WS-Mediator

Increased Coverage

Ideally, the RKB would include a substantive ResEx description for all mechanisms that the designer of a resilient system might enquire about.

It was suggested that we seek to ensure representation for mechanisms identified in key standards documents.

We have therefore included all relevant mechanisms identified in IEC 61508 “Functional safety of e/e/programmable safety critical systems (section 7)”. 
ResEx Challenge Workshops

• First Workshop: 14 July 2008, Pisa
  – Resilience Explicit Computing in Grids
• Second Workshop: 20-21 November 2008, Malvern
  – Resilience Explicit Computing in Critical National Infrastructures
• Third Workshop: 5 December 2008, Newcastle
  – Resilience Explicit Computing with Assistive Technologies

Aims for Challenge Workshops

• Select candidate problems
  – Ideally with input from practitioners
• Benchmark current technology
• “Benchmark” resilience explicit approach
• Exploitation of metadata
  – Guidance and support for design rationale
  – Semantic interoperability
  – Runtime reasoning, policies, reconfiguration services
  – Monitoring and verification
• Seek to establish a legacy working group
Grids Workshop

- Pisa, July
- Complex network of interconnected systems delivering a range of services
- Pisa, QinetiQ, Southampton + CERN, INFN
- Exciting discussion of immediate challenges and future demands
- Follow on to report on known resilience issues in Grid domain

CNI Workshop

Malvern, November
- Systems supporting national infrastructure on which society has critical dependence
- QinetiQ, Southampton + CPNI, St Andrews
- Fascinating discussion of attack modalities and protection tactics
- Forum established (led by ReSIST champion); next workshop on “Emergency Planning”
- Looking to build on links to Southampton
Assistive Technologies Workshop

- Newcastle, December
- Technology deployed in support of people suffering from impediments – of age or infirmity (for example)
- Southampton, Birkbeck + CELS, Dundee
- Scenario enactment and discussion of perceptions of resilience/dependability
- Working group established – two champions (two flavours 😊) and initial membership
Context

- CSAKTiveSpace
  - AKT Project
  - First Semantic Web Challenge winner 2003

- ReSIST - EU Network of Excellence in Resilient Systems
  - Knowledge-enabled infrastructure
Communication

- Ontologies
  - General Scientific Endeavour
  - Domain-specific
  - Support (geospatial, etc)

- Open Local Knowledge – HTTP
  - Resolvable URIs
  - SPARQL

- Uses Remote Knowledge
  - Resolves URIs with caching
Components 1

• Semantic Web infrastructure throughout
• Triplestore for each source
  – Putting the Web in Semantic Web
  – Stores RDF – (Subject, Predicate, Object)
  – We use 3store
• Linked Data
  – 303 and content negotiation architecture with caching

Components 2

• Co-Reference Subsystem
  – CRS – more later
• Community of Practice Analysis
  – Why do you think that?
• Ontology Mapping
  – Dealing with other Ontologies
• NLP for text classification
• Caching everywhere
Components 3

- Application Middleware
  - URI Equivalence Closure
  - RDF Graph Closure
- Semantic Sitemap
  - Facilitate Search Engines

User Interaction

- Semantic MediaWiki
- Custom form interfaces
- Google Maps
- Raw Knowledge Browser

- **RKBExplorer**
- Why do you think that? information
Focusing on a Person
Why do you think that?

Alexander Romanovsky is related to R de Lemos

They are linked by 33 relations.

**Publications**

They have co-authored 3 papers:
- Coordinated Atomic Actions in Modelling Objects Cooperation
- Exception Handling in a Cooperative Object-Oriented Approach
- Integrating COTS Software Components into Dependable Software Architectures

**Affiliations**

They are both affiliated to NEWCASTLE UNIVERSITY.
This is a page that gives a simple demonstration showing papers which have been deemed related through textual analysis by IAI Saarbrucken. Up to the top 20 are listed for each paper, when they meet a simple thresholding:

1 – very strong – 0.9 – strongly – 0.7 – related – 0.6 – ignored – 0

The 1980 paper Exception Handling and Software-Fault Tolerance [browse]

is very strongly related to

- [browse] 2003 "Automatic detection and masking of non-atomic exception handling" [PDF]
- [browse] 1989 "Formal Verification of Programs with Exceptions"
- [browse] 1983 "Programming Reliable and Robust Software in ADA"

is strongly related to

- [browse] 1998 "Improving software robustness with dependability cases" [PDF]
- [browse] 1999 "Wrapping windows NT software for robustness" [PDF]
- [browse] 1981 "Exception Handling and Error Recovery Techniques in Modular Systems - An Application to the Isaure System"
- [browse] 2003 "Deadlock resolution via exceptions for dependable Java applications" [PDF]
- [browse] 2002 "Robust software - no more excuses" [PDF]

is related to

- [browse] 1995 "Fault tolerance in concurrent object-oriented software through coordinated error recovery" [PDF]
- [browse] 2004 "Implementing simple replication protocols using CORBA portable interceptors and Java serialization" [PDF]
- [browse] 1984 "Fault Tolerance Using Communicating Sequential Processes"
- [browse] 2001 "Middleware support for voting and data fusion" [PDF]
ReSIST / Resilience-Explicit Computing Mechanisms

ReSIST :: Courses / Editing 'Advanced seminars on Distributed Systems'

Step 1 of 4: Information regarding the organisation of the course

Name of the course: Advanced seminars on Distributed Systems

Taught at:
- Università degli studi di Roma, La Sapienza
- Université LUM
- Université De Toulouse 1
- University of Naples
- Université of Toulouse III

Currently being taught:

Description:
The course focuses on recent advances on distributed systems. A set of topics is selected and studied through the help of original papers and, practically, most known distributed system platforms are selected and analyzed.

Language(s) of the course:
- English
- Esperanto
- Estonian
- Finnish
- French
- Gaelic

Select Author(s):
- Roberto Baldini
- Roberto Bertini
- Roberto Bonato
- Robin Bloomfield
- Rula Mardikian
- Sadie Creese

ReSIST :: Resilience-Explicit Computing Mechanisms

Name of the resilience mechanism: N-Version Programming/1/1

Submitted by:
Zoe Andrews

Author of mechanism:
Algirdas Avizienis

Associated projects:
None

Mechanism Objectives:
To utilise design diversity and voting in order to tolerate software faults

Detailed Description:
The information here applies to the specific variant of the mechanism NVP/1/1, described in "Definition and Analysis of Hardware- and Software-Fault Tolerant Architectures". The specific variant considered, NVP/1/1, has three diverse implementations of a software module. For a more general overview of the mechanism please see "The N-Version Approach to Fault-Tolerant Software".

Definition and Analysis of Hardware- and Software-Fault-Tolerant Architectures
Editing "N-Version Programming/1/1"

Step 5 of 7: Resilience metadata - how the mechanism helps a system’s resilience
(For questions, problems or feedback filling out this form, please email us.)

Failure Modes
(Select the ways in which your mechanism can fail to function as intended. To help you to decide what the appropriate failure modes are you should treat your mechanism as a black box and think about the kinds of failures you expect to observe from it. The terms in this list are taken from the ResIST ontology on security and dependability.)

<table>
<thead>
<tr>
<th>Failure Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consistent Failure</td>
</tr>
<tr>
<td>Content And Timing Failure</td>
</tr>
<tr>
<td>Content Failure</td>
</tr>
<tr>
<td>Early Timing Failure</td>
</tr>
<tr>
<td>Erasure Failure</td>
</tr>
<tr>
<td>False Alarm</td>
</tr>
</tbody>
</table>

(CTRL+Click to select multiple values)

Threats Addressed
(Select the threats to resilience that your mechanism aims to address, i.e. the faults it aims to remove, the errors it aims to compensate for and the failures it aims to prevent. The terms in this list are taken from the ResIST ontology on security and dependability.)

<table>
<thead>
<tr>
<th>Threat Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accidental Fault</td>
</tr>
<tr>
<td>Budget Failure</td>
</tr>
<tr>
<td>Catastrophic Error</td>
</tr>
<tr>
<td>Honeypot</td>
</tr>
<tr>
<td>Commission Fault</td>
</tr>
<tr>
<td>Complete Development</td>
</tr>
</tbody>
</table>

(CTRL+Click to select multiple values)

Resilience Metadata
In this question you are asked to think about the effect your mechanism has on the resilience of a system. If you aim to compare your mechanism to a different mechanism addressing a similar aim, what data would you use to choose which was fit for a specific purpose? This question allows you to define such metrics and associate a value with them for your mechanism. New resilience metadata metrics and values can be added to this list by clicking on the “add new item” link. Existing metadata instances can be deleted or edited by clicking the cross or the pencil next to them respectively. Note that when you add some metrics a new version is saved as well as the old one, which can then be deleted.

Time-dependent probability (P(t)) of undetected failure
PFOFO (Undetected) * application software’s execution rate * t Probability
Time-dependent probability (P(t)) of detected failure

Where is it Taught?

University of Southampton
School of Electronics and Computer Science

http://resist.ecs.soton.ac.uk/gmap/resist-courses.php

Budapest University of Technology and Economics
Courses taught at Budapest University of Technology and Economics, Budapest:
Software Verification and Validation [hide]
Istvan Majzik
Management of Computing Infrastructure [show instructors...]

Go to the ResIST Partners Map
Knowledge Sources

- Partners
- Publications
- Funding Agencies
- Project Wiki
- Courseware
- Resilient-Explicit Computing

- Wide range, don’t just look where you expect to find

Some Underlying Sources

- acm.rkbexplorer.com
- budapest.rkbexplorer.com
- citeseer.rkbexplorer.com
- cordis.rkbexplorer.com
- courseware.rkbexplorer.com
- darmstadt.rkbexplorer.com
- dblp.rkbexplorer.com
- deepblue.rkbexplorer.com
- deploy.rkbexplorer.com
- epsrc.rkbexplorer.com
- eurecom.rkbexplorer.com
- ft.rkbexplorer.com
- ibm.rkbexplorer.com
- ieee.rkbexplorer.com
- irit.rkbexplorer.com
- italy.rkbexplorer.com
- kaunas.rkbexplorer.com
- kisti.rkbexplorer.com
- laas.rkbexplorer.com
- lisbon.rkbexplorer.com
- newcastle.rkbexplorer.com
- nsf.rkbexplorer.com
- pisa.rkbexplorer.com
- rae2001.rkbexplorer.com
- resex.rkbexplorer.com
- roma.rkbexplorer.com
- southampton.rkbexplorer.com
- ulm.rkbexplorer.com
- unlocode.rkbexplorer.com
- wiki.rkbexplorer.com

Range from a few 100 to more than 10,000,000 “facts”
For example

- Statistics for repository kisti.rkbexplorer.com
  - Last data assertion 2008-09-18 17:16:41
  - Number of triples 12815162
  - Number of symbols 3239105
  - Size of RDF dataset 671M

Co-Reference

- Co-Reference is a Big Problem
  - Identifying multiple URIs for one resource
  - Rejecting incorrectly conflated resources
  - Publishing
  - Using
- Coldstart
  - A serious problem
  - Nothing is linked to anything
Co-Reference Closure

Co-reference Closure involves managing and publishing co-referent information. It includes:

1. Identifying co-referent pairs using tools.
2. Asserting relationships into the CRS.
3. Querying the CRS.
4. Recommending a Canon.

The CRS, or Consistent Reference Service, is a service designed to manage and publish co-referent information. It operates by:

- Creating a URI set of co-referent objects.
- Recommending a canonical reference.

This service facilitates a consistent representation of co-referent information across multiple repositories.
CRS continued

- CRS Policies are defined by context
  - Often one per Triplestore
  - Can be many per Triplestore for different purposes
  - May not be associated with a particular Triplestore
- Maintenance
  - Provenance
  - Rollback
- Can be used to infer owl:sameAs

Dealing With Non-SPARQL KBs

- The RKBExplorer application uses SPARQL to query the KBs
  - But needs to access data from KBs that only offer resolvable URIs
- So resolve such a URI
- Cache the RDF with associated resolved RDF locally
- Query the local cache
Dealing With Different Ontologies

- The RKBExplorer application uses a particular ontology
  - Some KBs will use different ontologies
  - E.g. kisti.rkbexplorer.com
- One solution
  - Represent the ontology relationship in RDF (as far as possible)
  - Resolve the URI through the mapping service to get RDF in the required ontology

Supporting resilience

- People, Publication, Projects, Research Areas
- Resilience-related topics
- Resilience-Explicit Computing
- Educational Resources

- In the future
  - Automating discovery of issues and solutions
    - Design time
    - Run time
Finding mechanisms that are appropriate for Hardware and Aerospace

SELECT DISTINCT ?mechanismURI ?mechanismName ?metadataName ?metadataValue WHERE {
  ?mechanismURI rdf:type resex:Resilience-Mechanism .
  ?mechanismURI resex:has-application-domain acm:J.2.0 .
  ?mechanismURI rdfs:label ?mechanismName .
}

<table>
<thead>
<tr>
<th>Result</th>
<th>Binding</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>?mechanismURI</td>
<td><a href="http://resex.rkbexplorer.com/id/resilience-mechanism-267972cd">http://resex.rkbexplorer.com/id/resilience-mechanism-267972cd</a></td>
</tr>
<tr>
<td></td>
<td>?mechanismName</td>
<td>N-Self-Checking Programming/1/1</td>
</tr>
<tr>
<td>2</td>
<td>?mechanismURI</td>
<td><a href="http://resex.rkbexplorer.com/id/resilience-mechanism-e679bd05">http://resex.rkbexplorer.com/id/resilience-mechanism-e679bd05</a></td>
</tr>
<tr>
<td></td>
<td>?mechanismName</td>
<td>N-Version Programming/1/1</td>
</tr>
<tr>
<td>3</td>
<td>?mechanismURI</td>
<td><a href="http://resex.rkbexplorer.com/id/resilience-mechanism-742552f">http://resex.rkbexplorer.com/id/resilience-mechanism-742552f</a></td>
</tr>
<tr>
<td></td>
<td>?mechanismName</td>
<td>Recovery Blocks/1/1</td>
</tr>
</tbody>
</table>

Inspecting metadata, number of variants

SELECT DISTINCT ?mechanismURI ?mechanismName ?metadataName ?metadataValue WHERE {
  ?mechanismURI rdf:type resex:Resilience-Mechanism .
  ?mechanismURI resex:has-application-domain acm:J.2.0 .
  ?mechanismURI rdfs:label ?mechanismName .
  ?metadata resex:has-value ?metadataValue .
}

<table>
<thead>
<tr>
<th>Result</th>
<th>Binding</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>?mechanismURI</td>
<td><a href="http://resex.rkbexplorer.com/id/resilience-mechanism-742552f">http://resex.rkbexplorer.com/id/resilience-mechanism-742552f</a></td>
</tr>
<tr>
<td></td>
<td>?mechanismName</td>
<td>Recovery Blocks/1/1</td>
</tr>
<tr>
<td></td>
<td>?metadataName</td>
<td>Number of variants</td>
</tr>
<tr>
<td></td>
<td>?metadataValue</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>?mechanismURI</td>
<td><a href="http://resex.rkbexplorer.com/id/resilience-mechanism-e679bd05">http://resex.rkbexplorer.com/id/resilience-mechanism-e679bd05</a></td>
</tr>
<tr>
<td></td>
<td>?mechanismName</td>
<td>N-Version Programming/1/1</td>
</tr>
<tr>
<td></td>
<td>?metadataName</td>
<td>Number of variants</td>
</tr>
<tr>
<td></td>
<td>?metadataValue</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>?mechanismURI</td>
<td><a href="http://resex.rkbexplorer.com/id/resilience-mechanism-267972cd">http://resex.rkbexplorer.com/id/resilience-mechanism-267972cd</a></td>
</tr>
<tr>
<td></td>
<td>?mechanismName</td>
<td>N-Self-Checking Programming/1/1</td>
</tr>
<tr>
<td></td>
<td>?metadataName</td>
<td>Number of variants</td>
</tr>
<tr>
<td></td>
<td>?metadataValue</td>
<td>4</td>
</tr>
</tbody>
</table>
Inspecting metadata, average cost of implementing fault tolerant system vs- cost of implementing non fault tolerant system

```
SELECT DISTINCT ?mechanismURI ?mechanismName ?metadataName ?metadataValue WHERE {
  ?mechanismURI rdf:type resex:Resilience-Mechanism.
  ?mechanismURI resex:has-application-domain acm:J.2.0.
  ?mechanismURI rdfs:label ?mechanismName.
      ?mt rdfs:label ?metadataName.
      ?metadata resex:has-value ?metadataValue.
}
```

Comparison of the operational overheads in determining a fault has occurred

```
SELECT DISTINCT ?mechanismURI ?mechanismName ?metadataName ?metadataValue WHERE {
  ?mechanismURI rdf:type resex:Resilience-Mechanism.
  ?mechanismURI resex:has-application-domain acm:J.2.0.
  ?mechanismURI rdfs:label ?mechanismName.
    ?metadata resex:metadata-type id:resilience-metadata-type-3443934c.
      ?mt rdfs:label ?metadataName.
      ?metadata resex:has-value ?metadataValue.
}
```

---

<table>
<thead>
<tr>
<th>Result</th>
<th>Binding</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>?mechanismURI</td>
<td><a href="http://resex.kbexplorer.com/id/resilience-mechanism-e07b5b05">http://resex.kbexplorer.com/id/resilience-mechanism-e07b5b05</a></td>
</tr>
<tr>
<td></td>
<td>?mechanismName</td>
<td>N-Version Programming/1/1</td>
</tr>
<tr>
<td></td>
<td>?metadataName</td>
<td>Av CFT/CNFT</td>
</tr>
<tr>
<td></td>
<td>?metadataValue</td>
<td>2.25</td>
</tr>
<tr>
<td>2</td>
<td>?mechanismURI</td>
<td><a href="http://resex.kbexplorer.com/id/resilience-mechanism-367972cd">http://resex.kbexplorer.com/id/resilience-mechanism-367972cd</a></td>
</tr>
<tr>
<td></td>
<td>?mechanismName</td>
<td>N-Self-Checking Programming/1/1</td>
</tr>
<tr>
<td></td>
<td>?metadataName</td>
<td>Av CFT/CNFT</td>
</tr>
<tr>
<td></td>
<td>?metadataValue</td>
<td>3.01</td>
</tr>
<tr>
<td>3</td>
<td>?mechanismURI</td>
<td><a href="http://resex.kbexplorer.com/id/resilience-mechanism-7f425f52">http://resex.kbexplorer.com/id/resilience-mechanism-7f425f52</a></td>
</tr>
<tr>
<td></td>
<td>?mechanismName</td>
<td>Recovery Blocks/1/1</td>
</tr>
<tr>
<td></td>
<td>?metadataName</td>
<td>Av CFT/CNFT</td>
</tr>
<tr>
<td></td>
<td>?metadataValue</td>
<td>1.75</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Result</th>
<th>Binding</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>?mechanismURI</td>
<td><a href="http://resex.kbexplorer.com/id/resilience-mechanism-7f425f52">http://resex.kbexplorer.com/id/resilience-mechanism-7f425f52</a></td>
</tr>
<tr>
<td></td>
<td>?mechanismName</td>
<td>Recovery Blocks/1/1</td>
</tr>
<tr>
<td></td>
<td>?metadataName</td>
<td>Errors cp time overheads</td>
</tr>
<tr>
<td></td>
<td>?metadataValue</td>
<td>One variant and acceptance test execution</td>
</tr>
<tr>
<td>2</td>
<td>?mechanismURI</td>
<td><a href="http://resex.kbexplorer.com/id/resilience-mechanism-367972cd">http://resex.kbexplorer.com/id/resilience-mechanism-367972cd</a></td>
</tr>
<tr>
<td></td>
<td>?mechanismName</td>
<td>N-Self-Checking Programming/1/1</td>
</tr>
<tr>
<td></td>
<td>?metadataName</td>
<td>Errors cp time overheads</td>
</tr>
<tr>
<td></td>
<td>?metadataValue</td>
<td>Possible result switching</td>
</tr>
<tr>
<td>3</td>
<td>?mechanismURI</td>
<td><a href="http://resex.kbexplorer.com/id/resilience-mechanism-e07b5b05">http://resex.kbexplorer.com/id/resilience-mechanism-e07b5b05</a></td>
</tr>
<tr>
<td></td>
<td>?mechanismName</td>
<td>N-Version Programming/1/1</td>
</tr>
<tr>
<td></td>
<td>?metadataName</td>
<td>Errors cp time overheads</td>
</tr>
<tr>
<td></td>
<td>?metadataValue</td>
<td>Usually negligible</td>
</tr>
</tbody>
</table>
Concluding Remarks

- Major Data Fusion using Semantic Web Technologies
- Many things can be cast in a Semantic Web framework
- Linked Data works pretty well
- RDF works pretty well
- A little Ontology goes a long way
  - But is tractable

RKBExplorer.com/explorer/ – Try it!

http://eprints.ecs.soton.ac.uk/17025
ReSIST
Resilience for Survivability in IST
A European Network of Excellence

Research Agenda — International Survey
Jean-Claude Laprie

From the 2nd review outcome

« We are worried that the deliverable D13 contains the favoured research directions of the authors, but may omit the concerns of others outside the ReSIST project »

« The project should make a serious attempt to reach the 200-300 top researchers, research groups and leading industrial experts in the fields related to resilience (dependability, safety, security), ask them all for their list of the five most prominent issues for the next, say, 10 years in their field of interest, and make sure that this query is answered »
**Research Agenda Process**

Research gaps and challenges

**Clustering**

- Evolvability
- Assessability
- Usability
- Diversity

**Resilience**

- Resilient ubiquitous systems
- Adaptation and self-organisation
- Models for ubiquitous systems
- Resources and infrastructures for ubiquitous systems
- Assessing evolvable systems
- Methods and techniques to assess evolvable systems
- Assessability as an engineering discipline
- Improved development processes
- Contextual usability
- Going beyond standard usability
- Small-scale diversity
- Large-scale diversity

**Resilience-building WGs**

Architecture Algorithms Socio-technical Verification Evaluation

**Projection**

41 ‘gaps & challenges’, grouped into 12 clusters, of the 4 resilience-scaling technologies

55 co-authors from 17 partners
International survey

- Coordinators: Michel Banâtre, Karama Kanoun, Jean-Claude Laprie
- Contributions expected under the form of texts, structured according to the four resilience technologies when relevant
- Call for contributions sent to 236 carefully selected addressees, from academia and industry, and providing an extensive coverage of the field, broader than the expertise represented in ReSIST
- Flyer produced and distributed at DSN 2008
- Forty one contributions have been received. The contributions have been synthesized by the four working group leaders:
  - Evolvability: David Powell
  - Assessability: Aad van Morsel
  - Usability: Philippe Palanque
  - Diversity: Lorenzo Strigini

Flyer produced and distributed at DSN 2008
Forty one contributions have been received. The contributions have been synthesized by the four working group leaders:

- Evolvability: David Powell
- Assessability: Aad van Morsel
- Usability: Philippe Palanque
- Diversity: Lorenzo Strigini

The European Network of Excellence ReSIST (Resilience for Survivability in Information Society Technologies) is performing, at the request of the European Commission, an international survey of the research gaps and challenges in resilience of ubiquitous computing systems.

As a DSN attendee, you naturally qualify for contributing to this survey. Providing your views under the simple form of one or two paragraphs will be most welcome. Please, send your response at your earliest convenience, and by July 7 at the latest, at:

resist-survey@ias.fr

The outcome of the survey should be exploited by the European Commission for defining future workflows, including international cooperations. All contributions will be acknowledged.

ReSIST defines resilience as the persistence of service delivery that can justifiably be trusted when facing changes, i.e., the persistence of dependability when facing changes. Resilience is viewed as involving four major properties: a) evolvability, i.e., the ability to successfully accommodate changes, including adaptivity, i.e., the capability of evolving while executing, b) assessability, in both senses of verification and evaluation, c) usability, and d) diversity. In order to facilitate the processing of your response, indicating to which of those properties the research gaps and challenges you view relate to, would be of great help.

Information about ReSIST is available on the website: http://www.resist-noe.eu

ReSIST partners

- LAAS-CNRS, France (Coordinator)
- Budapest University of Technology and Economics, Hungary
- City University, London, UK
- Technische Universiteit Darmstadt, Germany
- Deep Blue Srl, Italy
- Institut Eurécom, France
- France Telecom Recherche et Développement, France
- Bill Research GmbH, Switzerland
- Universidad de Rennes 1 – IRISA, France
- Università di Torino – IIT, Italy
- Vägstrans Magres University, Kaunas, Lithuania
- Faculdade de Ciências da Universidade de Lisboa, Portugal
- University of Coventry, UK
- Università di Pisa, Italy
- Cyber Associati Limited, UK
- Università degli Studi di Roma “La Sapienza”, Italy
- Universitädt Ulm, Germany
- University of Southampton, UK
Syntheses

- All but one contributions referred to
- References to contributions:
  - Evolvability: 26
  - Assessability: 22
  - Usability: 11
  - Diversity: 15
- Globally, contributions to the survey provide a lower coverage than D13 (25 ↔ 41)

<table>
<thead>
<tr>
<th>Research gaps and challenges of D13</th>
<th>Evolvability</th>
<th>Assessability</th>
<th>Usability</th>
<th>Diversity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research gaps and challenges of D13 identified as being related in the contributions</td>
<td>GE1, GE2, GE4, GE5, GE6, GE7, GE8, GE9, GE10</td>
<td>GA3</td>
<td>GU1, GU5</td>
<td></td>
</tr>
<tr>
<td>Assessability synthesis</td>
<td>GA1, GA4, GA5, GA7, GA8, GA10, GA17, GA18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usability synthesis</td>
<td>GA1, GA4, GA10</td>
<td>GU1, GU6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diversity synthesis</td>
<td>GE7</td>
<td>GA15, GA16</td>
<td>GD3, GD4</td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>9</td>
<td>11</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Number of research gaps and challenges in D13</td>
<td>11</td>
<td>18</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>
Interesting complements to D13 research gaps. Examples:

- From broad viewpoints:
  - Widespread belief in importance of metrics
  - Need for toolsets
- From focused viewpoints:
  - Accessibility by disabled persons
  - Usable security
  - Possible erosion of diversity by collective human behavior
- From differing contexts or environments of contributors:
  - Space industrialists: focus on goal-directed autonomy, and, as a consequence, on observability

Unsurprising confirmation: incompatibility of safety-critical systems and of evolvability

- Licensing/certification issues
- Long term perspective?

One new research gap, regarding usability: plug-and-play systems, i.e., usable from start-up → Contextual usability cluster
Integration indicators

• Exchange of personnel
  – 2006: 5 long visits, 6 short visits
  – 2007: 8 long visits, 8 short visits
  – 2008: 6 long visits, 10 short visits

• Co-advised doctorate theses: 9

• Researchers in doctorate committees at other ReSIST partners

• Joint publications
  – 2003-2005: 18 (3 by 3 institutions) / 533 = 3 %
  – 2006-2009: 59 (8 by 3 institutions, 2 by 4 institutions, 2 by 5 institutions) / 484 = 12 %
    • 2006: 6
    • 2007: 12
    • 2008-2009: 41
AROVE-ν
“Assessing the Resilience of Open Verifiable E-voting” (scientific results)

Presented by Eugenio Alberdi
City University, London

ReSIST Mini-Project - Partners

• Newcastle:
  – Peter Ryan, Kieran Leach, Johannes Clos

• IRIT, Toulouse:
  – Philippe Palanque, Marco Winckler, Nathalie Kaing, Regina Bernaupt

• City, London:
  – Lorenzo Strigini, Eugenio Alberdi

• Surrey:
  – David Bismark (né Lundin)
Reminder: Intro to AROVE-v

- A practical, case-study-based learning about building dependability cases for large, integrated socio-technical systems
  - application: E-voting (as a good example of such complex systems) – voting scheme: Prêt à Voter
- MAIN GOAL:
  - to identify necessary components of a case supporting the claim that a certain E-voting system is fit for use

Reminder: E-voting

- Voting: complex socio-political activity requiring a combination of
  - accuracy, privacy, security, trust, successful termination,...
- E-voting systems have been presented as solutions to some of the limitations of conventional “paper based” systems
  - e.g. automatic ballot counts can improve accuracy
  - BUT: needs lots of "trusted" software
    - votes disappear into the machine
  - cryptographic algorithms can provide verifiability while preserving privacy & accuracy; etc. (Prêt à Voter)
    - E-voting algorithms and mechanisms have been tested and shown to work
A Dependability Case for E-voting

- A GAP in the literature:
  - Practical need for a complete case demonstrating that a specific system as a whole has sufficiently high probability of exhibiting the desired properties when in use in an actual election

- Components of a CASE:
  - what are the claims made?
    - for a start ... what were the requirements?
  - what are sound arguments for believing such claims
    - for a real, flesh-and-blood-and-copper-and-silicon system?
  - where would one get the evidence to support these arguments?

Reminder: Activities

Ongoing Literature Review

System Description (..., Task Analysis, ...)

HAZOP
(and similar analyses for identifying deviations)

Previous literature on Prêt à Voter

Preliminary Probabilistic Analyses

CASE (subset)

Observations from Trial

Recommendations for cases and for evidence collection
Prêt à Voter in a nutshell

– based on public-key crypto and an intuitive, paper-based user interface
– no need for expensive ad hoc machinery
– encrypted votes and decryption/counting results are visible on a web bulletin board
– each voter receives a receipt
  • allowing him/her to verify that his vote is being counted
  • but no-one to guess how he voted
– decryption, counting in multiple phases performed by mutually suspicious parties

it's magic!

– Will a specific implementation work with real voters, politicians, machines, election officials, adversaries?

Towards a Structure of a Case

Considerations for organising the CASE:

• a set of requirements (4)
• the components of an election (functions in the system)
• Prêt à Voter "at least as good as" POPS (Plain Old Paper System)
• "stages": attack-corruption-detection-recovery
Requirements: Highest Level Sub-Claims

- **ACCURACY requirement**: if and when the election system declares the election successfully completed, the final election result that it has produced will match (within reasonable margins of error) the votes that voting intentions of all legitimate voters as they enter the ballot booth.

- **PRIVACY requirement**: under no circumstance, not even with the connivance of the voter, shall any person gain from the election system evidence of for whom or for what the voter voted, apart from the vote tallies that the election system is required to publish.

- **TRUSTEDNESS requirement**: most citizens will trust the election process "enough" to take part, using it as required (i.e., they will act on the assumption that the other three requirements are met), and to accept its results.

- **SUCCESSFUL TERMINATION requirement**: the election system will declare the election successfully completed, by a deadline specified in its requirements - it has a very high probability to succeed with all the above requirements being met.
“Deviation Chain” (e.g. for “Ballot Box Integrity”)\(^\text{11}\)

Each claim about undetected corruption to be structured according to the sequence of stages:
attack (attempt/fault) – corruption – detection - recovery

---

Case Skeleton (1)

- **Voter Accuracy**
  - (When filling in the ballot) VOTER is unlikely enough to submit a different vote from that intended

- **“Ballot Box” Integrity**
  - Undetected CORRUPTION of votes while being recorded and case stored in the BALLOT DATABASE ("ballot box") is unlikely enough

- **Count Integrity**
  - Undetected deviations of COUNT INTEGRITY are unlikely enough
Case Skeleton (2)

Undetected Corruption of the 'Ballot box' (WBB) integrity is unlikely enough to happen

Undetected Corruption of the transmission of electronic ballot receipts to the WBB is unlikely enough to happen

Undetected Corruption of the electronic ballot receipts while in the WBB is unlikely enough to happen

Case Skeleton (3)

Undetected Corruption of the transmission of electronic ballot receipts to the WBB is unlikely enough to happen

Undetected alteration of electronic ballot receipts is unlikely enough to happen

Undetected invalidation of electronic ballot receipts is unlikely enough to happen

Undetected illegal adaption (counterfeit) of electronic ballot receipts is unlikely enough to happen

If hacking occurs, it is likely enough to be detected

Digital signature (is signature validated) is unlikely enough to happen

Various checks are performed (signature validated is unlikely enough to result in starting)

Theorem about cryptography and software testing results

Trends of First a Model
Outcomes

• Integrated description of relevant aspects of the whole socio-technical system
• Case skeleton focusing on ‘accuracy’
• Dissemination:
  – 3 accepted conference papers
  – more to come
• Future work on:
  – expanding case for other requirements (beyond ‘accuracy’)
    • trade-offs amongst requirements
  – case structure and probabilistic modeling (City)
  – recovery mechanisms (Newcastle/Luxembourg)
  – design rationale (IRIT)
• Plans for joint post-ReSIST proposals

questions?
The ASAP project: Assessment-Based Adaptable Software Architecture for Dependability

JC. Fabre, T. Robert, T. Pareaud
P. Popov, V. Stankovic, I. Gashi
F. Taiani, S. Lin
I. Zutautaite-Seputiené

LAAS-CNRS, Toulouse, France
City University London, UK
Lancaster University, UK
University of Kaunas, Lithuania

European Network of Excellence ReSIST
Resilience for survivability in IST

Problem statement

• Why Adaptive Fault tolerant system?

Adverse operational conditions

Environment

Adaptive Fault Tolerant System

1

2

Fault tolerance (FT) ==
Global property binding the 1 & 2

Software modifications due to Adaptation

Ideally Adaptation should preserve FT properties in both cases

12-13 March 2009
ReSIST Final Workshop - ASAP mini-project

2
Adaptation Triggers

- **Conventional adaptation triggers**
  - Update of the functionalities
  - Performance optimization through resource consumption tuning.

- **Adverse operational conditions**
  - Mismatch between operational conditions and design assumptions made for the deployed Fault Tolerant mechanisms (FTMs) relevance

- **Side effects of local adaptation on global FT**
  - What happens when a functional service $S$ has to be updated, while $S$ is combined with at least one FTM

---

Why On-line Assessment?

- **Problem**
  - Adaptation decisions often rely on several quantitative estimation of the implementation attributes
  - Estimation of an attribute $A = \text{value} + \text{uncertainty}$
  - Adaptation trigger $= \text{Prob}(A < T) > \text{Confidence}$?

- **Off-line estimation methods limits**
  - Difficulties to model all operational conditions
  - Require huge Data Set $=>$ very costly or impossible

- **Solution: On-line assessment**
  - Be able to take advantage of knowledge built off-line
  - Update the attribute estimation with observations collected on-line
The ASAP Framework

ASAP Framework Architecture

- **A Reflective architecture separating:**
  - Fault tolerance mechanisms
  - Adaptation of FT application
  - System attributes assessment

- **Architectural principles**
  - **Software Adaptation** ⇔ Fine-grain modification at runtime of software to minimize adaptation cost
  - **Adaptation Triggers** ⇔ (i) Adverse operational context, (ii) Side effects of application software modifications
Fault tolerant design & Adaptation

- Fault tolerant open software system
  - Provide design patterns for fault tolerance
  - Provide means to add/remove/modify at runtime the software system (code, state, ...)

- Component based design + reflection

Fault Tolerant Software

Fault Tolerance Mechanisms

Application software

FT Software configuration

Combination of an FTM with the application

Separation of concerns (S.o.C.) and software decomposition

Fault tolerant Software Design

- Decomposition for adaptation of the fault tolerant Software

Decomposition

- Reflective Component Based Middleware (OpenCOM)
  - Observe and modify the state of the components
  - Observe and control the interactions between components at runtime
  - Observe and modify the component architecture (creation, destruction, insertion and removal of components)
Componentization for Adaptation (1)

Leader Follower Replication agent

Componentization & Adaptation (2)

Primary Backup Replication agent
Bayesian Assessment (1)

- **Idea:** use probabilistic models to represent attributes together with their uncertainty.
  
  *How to model and assess the uncertainty of a parameter? (quantification of uncertainty)*

- **Solution:** The Bayesian approach provides the opportunity to quantify the uncertainty using probabilistic models
  
  - It allows one combining:
    - The prior belief (which is ‘subjective’ and possibly inaccurate) about the values of a parameter, e.g. a probability distribution.
    - The (‘objective’) evidence from seeing the modeled artifact in operation.
  
  - To obtain a posterior belief, a new probability distribution, about the values of the assessed parameters.
    - This posterior distribution updates *quantification of uncertainty of parameters*
    - This posterior distribution takes into account both the prior knowledge and the empirical evidence.
Bayesian Assessment (2)

Assessment & Bayesian Inference engine

- **Assessed attribute**: a parameter modelling the Failure probability of a service \( (pfd) \)

- **Assessment implementation**:
  - Deployment of a quarantine state to perform service assessment on-line
  - An observer collect Success and Failure observations along the assessed service execution. \( (Oracle) \)
  - The parameter distribution (the prior) representing the current knowledge of the parameter
  - The decision block that check if the attribute position with respect to a threshold can be decided \( (taking \ in \ account \ the \ estimation \ uncertainty) \)
  - The Bayesian Inference engine is implemented in Java and integrated to OpenCOM as a component
**Smart update example**

- **Initial state:** a functional service \( S \) currently implemented by a component \( V_1 \)
- **Trigger:** a new version of \( S \) is available and loaded as \( V_2 \) in the system
- **Expected behaviour:** replace \( V_1 \) by \( V_2 \) to enhance \( S \)
- **Implicit expected adaptation:** the framework adapts the fault tolerance mechanisms, according to the dependencies between the implementation of \( S \) and its \( FTMs \)
- **Restriction:** the new version should be enforced iff its probability of failure on demand is lower than \( P_{\text{max}} \) with a confidence greater than \( C_{\text{min}} \)

\[ S.V_1/FTM1 \rightarrow S.V_2/FTM(S.V_2)? \]

**Smart update context**

- **Available FTMs:**
  - Leader follower replication (LFR)
  - Primary backup replication (PBR)
- **Assumptions: Service implementation vs FTM**
  - \( PBR \) can always be enforced for any version of \( S \)
  - LFR is applied when the service version is deterministic
- **Version \( V_2 \) of \( S \) exhibits different attributes**
  - Determinism (known a priori)
    - \( V_2 \) is not fully deterministic \( \Rightarrow \) LFR not relevant
  - Probability of failure on demand
    (uncertain knowledge \( \Rightarrow \) assessment)

\[ S.V_1/FTM1 \rightarrow S.V_2/FTM(S.V_2)? \]
Conclusions

- **Framework and technologies for assessment-based adaptation**
  - Reflection enables separation of concerns
  - CBSE enables fine-gain software adaptation
  - On-line assessment of quantitative parameters controls adaptation

- **A simple case study for proof of concepts**
  - Partial application of the smart update process to versions management of a software controller
  - Decomposition of LFR and PBR FT replication strategies
  - Software adaptation using OpenCOM and BI engine as a software component in Java

- **Other activities**
  - FT Software adaptation based on structural and behavioural modelling
  - Integration of Bayesian parameter assessment in a Gossip protocol

**Promising work, still work to be done in a long term project.**
Questions?
Context

* Two fundamental technological shifts:
  * internet -> ambient systems
  * deployment of user-carried systems
  * wireless communication (short range) + localization devices
  * link between physical and logical (network) world
Where do we stand?

- Extensive research in “closed” systems
- Abstractions, models, algorithms for resilience
- Extensive research on Internet
- Routing, models, structures (overlays)
- Can we get the “best of both world”
- I.e. provide localized abstractions

System’s characteristic parameters

<table>
<thead>
<tr>
<th>“Classical” systems</th>
<th>Mobile systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure (node, link)</td>
<td>Normal behaviour: disconnections, unreliable wireless communication</td>
</tr>
<tr>
<td>(Small) fixed number of nodes</td>
<td>Variable and huge size system</td>
</tr>
<tr>
<td>No link between physical world and network</td>
<td>Strong coupling with physical environment</td>
</tr>
</tbody>
</table>
# System's characteristic parameters

<table>
<thead>
<tr>
<th>&quot;classical&quot; systems</th>
<th>mobile systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>failure (node, link)</td>
<td>normal behaviour: disconnections, unreliable wireless communication</td>
</tr>
<tr>
<td>(small) fixed number of nodes</td>
<td>variable and huge size system</td>
</tr>
<tr>
<td>no link between physical world and network</td>
<td>strong coupling with physical environment</td>
</tr>
</tbody>
</table>

Many parameters: how to model?

---

# FADA approach

* (Re)define building blocks (abstractions) for a given physical region of interest
  * local consensus
  * local group membership
  * local storage

A toolbox to ease applications deployment, and ensure resilient computing
Local computing

* Different approach from GeoQuorums that focus on global data dissemination
* Local = geo-localized
* The architecture must be (re)defined w.r.t. a particular location in space.
* Semantics must be consistent with systems’ characteristics:
  * When no user populates a region, it’s not possible to keep a state alive

What are the applications to this?

* Real-life physical examples
  * users deploy a white board
  * perform better GPS route calculation
    * based on users’ experience of the traffic
  * cooperative backup of critical data
  * distributed black box, etc..
* augmented games
Local Shared Storage

- Provide a Register-like semantics in a particular location $A$
- Following 1985 Lamport’s registers
  - regular/safe/atomic
  - non-concurrent $\rightarrow$ concurrent
- Here: regular, non-concurrent writes. No crash of processes (only mobility)

System definition

- Entities $(p_i)_{i \geq 0}$
  - evolve in space with bounded speed
  - equipped with positioning device ("infinite" precision)
  - communication using wireless device
  - do not crash...
- Let’s concentrate on an area $A$
Simplified Architecture

Every thing defined w.r.t. $A$

Geo-reliable broadcast

Assumed to be provided by the system

$\star$ $(\delta, A)$ geo-reliable broadcast:

$\star$ every process in $A$ can issue a broadcast($m$)

$\star$ if $m$ is broadcasted at time $t$ by a process that remains in $A$ from $t$ to $t+\delta$ then all processes in $A$ during $[t, t+\delta]$ deliver the message
Geo-reliable broadcast

But...

- If a process leaves $A$ during the sending interval... no guarantee
- Core region (geographic definition)
  - a subset $A'$ of $A$ s.t. every message sent by a process in $A'$ will be delivered by all correct processes that were in $A'$ when the message was sent

Geo-registers

- Simple case: Non concurrent writes
  - write is allowed in the core region $A'$
  - read is allowed in $A$ (after some delay)
Non concurrent write semantics

What is the "last written value"?

\[ \mathcal{V} = \{ \text{last written value, concurrently written values} \} \]

(here \( \mathcal{V} = \{ y, z, t \} \))

If, since the last completed write operation,

* 1) core region was never empty, then \( v \in \mathcal{V} \) must be returned
* 2) else it returns \( v \in \mathcal{V} \) or \( \bot \)

---

Geo-registers

Structure induced by the model:

- 1 geographic thread
- 1 communication thread

Geographically controlled thread:

\begin{verbatim}
when p enters A:
    Rp ← void;
    wait for
        (W(x) is received) : Rp ← x; exit;
        (2δ time delay elapsed)
    RB_send(REQ)
    wait for
        (REP(v) is received) : Rp ← v;
        (W(x) is received) : Rp ← x;
        (2δ time delay elapsed) : Rp ← ⊥;
when p leaves A:
    free(Rp);
\end{verbatim}

Communication controlled thread:

\begin{verbatim}
upon reception of (REQ) : if (Rp ≠ void) then RB_send(REP(Rp))
upon reception of (W(x)) : Rp ← x
\end{verbatim}

Read and Write operations:

When \( p \) is in \( A \):
\begin{verbatim}
read() : wait until (Rp ≠ void) then return(Rp);
\end{verbatim}

When \( p \) is in \( A' \):
\begin{verbatim}
write(x) : RB_send(W(x));
\end{verbatim}
Properties...

* Region/core region interest:
  * abstracts away physical parameters (network parameters, speed)
  * clean definitions
  * simple implementation of shared storage

Properties...

* Register semantics:
  * applications that need to store information only when users populate an area
  * store user-centered information
  * no user ? no information (e.g. mean speed car-to-car)
Current work / Extensions

- Concurrent writers case
- Behavior in presence of failures
- Experimental evaluation
- Implementation in one-hop communication model
- Distant reading of the state of the storage

Future work

- New abstractions
  - counting/membership,
  - consensus-like
  - stronger semantics
- Weaker assumptions
  - geo-broadcast is sufficient, but what is the weakest building block needed?

ideally: provide a complete toolbox for simple ambient systems programming
FAERUS

Formal Analysis of Evolving Resilient Usable Systems

Mieke Massink (CNR-ISTI, Pisa, Project Leader)

FAERUS final review meeting, Toulouse, March 12, 2009

—Project Participants—
Maurice ter Beek (CNR-ISTI), Jeremy Bryans (Univ. of Newcastle), Giorgio Faconti (CNR-ISTI), Michael Harrison (Univ. of Newcastle), Nathalie Kaing (IRIT), J.F. Ladry (IRIT), Diego Latella (CNR-ISTI), Philippe Palanque (IRIT), Marco Winckler (IRIT)

Outline

1. Introduction
2. Participants
3. Goals
4. Scientific Approach
5. Results:
   5.1 A Fluid Flow approach to usability analysis in CSCW
   5.2 A Fluid Flow approach to crowd modelling in smart env.
   5.3 Stochastic analysis of resilience to interrupts
   5.4 Advanced probabilistic and stochastic modelling languages
6. Conclusions and Outlook
Introduction

Future vision:
- ubiquitous networked devices
- context aware services
- interaction techniques vary due to
  - dynamic reconfiguration, implicit interaction
  - dynamic availability of a huge variety of services

Contemporary HCI models are not adequate:
- Interaction techniques cannot be assumed to be a fixed set
- Environment and context change continuously
- Users are mobile and susceptible to (frequent) interrupts
- Users do not only interact with system but also with each other
- Users are many and their behaviour influences system and other users

Participants

CNR-ISTI, Pisa:
Mieke Massink (PL), Maurice ter Beek, Diego Latella, Giorgio Faconti

IRIT, Toulouse:
Nathalie Kaing, Jean Francois Ladry, Philippe Palanque, Marco Winckler

Univ. of Newcastle:
Jeremy Bryans, Michael Harrison
Goals

Overall goal:

investigate user centered modelling of usability issues in ubiquitous systems

Gaps addressed:

- modelling of usability related non-functional aspects of interaction
- dealing with diversity of interaction techniques and resilience to interrupts
- aspects of context and mode confusion

Main objectives:

- development of formal stochastic models to analyse resilience of multi-modal interfaces to interrupts
- development and analysis of formal models to analyse combined user and system behaviour in the presence of many autonomous users (many: ranging from 10 to several thousands).

Scientific Approach

- Stochastic model checking applied to joint user and system model addressing resilience to interrupts
- Application of the Fluid Flow approach (with PEPA and ODE) to multi-user, distributed systems to study the effect of different use-patterns
- Feasibility study of Fluid Flow approach to analyse crowds in smart environments
**Project Meetings**

- Kick-off meeting Pisa, 18-20 Feb, 2008, Plenary
- Skype meetings and email collaborations:
  - March 15-April 2, email, Pisa-Newcastle: Fluid-Flow
  - April 23, Skype, 15.00-17.00 IRIT-Pisa: Resilience
  - April 30, Skype, 10:00-12.00 IRIT-Pisa: Resilience
  - May 6, Skype, 15:00-18:00 IRIT-Pisa: Resilience
  - May 15, Skype, 15:00-17:00, IRIT-Pisa: Resilience
  - April 23, Skype, 10.00-12.00 IRIT-Pisa: Resilience
  - May 6, Skype, 15:00-18.00 IRIT-Pisa: Resilience
  - May 15, Skype, 15:00-17:00, IRIT-Pisa: Resilience
  - May 15-May 29, email, Pisa-Newcastle: Fluid Flow
  - May 26, Skype, 15:00-16:00, IRIT-Pisa: Resilience
  - May 30, Skype, 13:00-14:00, IRIT-Pisa: Resilience
  - June 5, Skype, 9.30-10.30, Plenary
  - August-December, regular email and Skype collaborations
- Meeting Newcastle, 9 June, Plenary
- Meeting Pisa, September 24, Plenary

**Results (1)**

Publications (joint publications in blue):


Results (2)


Multi-user Systems

Collaborative system
**A Fluid Flow Approach to Usability Analysis [HCSE08]**

Collaborative design system with 90 users and 30 file managers:

\[
\frac{d}{dt} \text{ Work}(t) = -\min(\text{Work}(t) \times w, \text{FMbusy}(t) \times \text{top}) \\
+ \frac{r}{(r+a)} \times \min(\text{Retry}(t) \times (r + a), \text{FMfree}(t) \times \text{top}) \\
+ \frac{a}{(r+a)} \times \min(\text{Client}(t) \times (r + a), \text{FMfree}(t) \times \text{top})
\]

**PEPA and Fluid Flow analysis [Hillston, QEST 2005]**

**Evolution of the system**

Rates (per hour): $a=0.5$, $w=0.25$, $r=5*a$
Modelling abstractions

Lessons learned:
- Abstraction from identity of clients can be justified:
  - For performance analysis it is irrelevant which of the clients that made a file request gets served first
- Abstraction from identity of files means that clients are randomly requesting any file (free or occupied)
- Clients are handing in files to any available file manager
- All requests get eventually served (with probability 1)

Allows comparison of:
- File management policies: retry vs. queues
- Performance assuming different use patterns
- Performance of policy with very large number of users

Multi-User Systems

Shared space system
Modelling crowds in smart environments

Scenario:
- Guidance system for people visiting buildings composed of many spaces
- Shared display with many slots in each space
- Implicit communication between visitor and environment

Visitor:
- Enters building, gets electronic ticket with final destination
- Takes a seat, watches display
- Request is made implicitly
- Display shows slot with required information
- Visitor gets up and moves to next indicated space until final destination has been reached

Models

Many different formal models developed and analysed:
- Detailed model in Promela (SPIN model checker)
- Stochastic models in PEPA (Performance Evaluation Process Algebra)
  - Version with shared displays with several slots in each space
  - Version with several single slot displays in each space
Single slot display

Example of an experimental situated display (Lancaster University)

Example configuration

Building layout:

A B C D

E

Four groups of visitors:
- 25 from A to D
- 75 from C to E
- 100 from A to C
- 200 from D to A

In each room 100 places to sit and 2 slots on the shared display
PEPA models

Models of:
- Visitor
- Place
- Slot
- Arbitrator
- Slotmanager

specified in PEPA and composed together
(details in technical report)

Fluid flow and simulation results

Average number of visitor arrivals
Fluid flow and simulation results

Fluid Flow

Simulation

Average number of occupied places to sit

Automatic generation of specifications

Given:

- Building layout
- Groups of visitors and destinations
- Resources in each room
- Routing table

A corresponding PEPA specification can be generated and then used for analysis

First exploration: 26 rooms, 420 visitors.
**Resilient User Interfaces**

Satellite Uplink Control Center

**Resilience to interrupts**

- Two interaction techniques: drag’n’drop and speak’n’drop
- Multi-modal (e.g. mouse and voice)
- User main task interrupted: e.g. pop-up windows
- Model of user part informed by cognitive theory (ICS, Barnard 1985) and results on human factors (e.g. Fitts’ Law studies)
- Joint stochastic model comprising behaviour of user, system and interrupts
- Performance Process Algebra models (PEPA)
- Analysis by stochastic model checking (PRISM)
Interacting Cognitive Subsystems

[Barnard & May, 1993]

Operating a mouse

---

**Fitts’ Law**

Movement time (MT) depends on **Distance** and **Width** of object:

\[ MT = a + b \log_2 \left( \frac{D}{W} + 1 \right) \]

Movement has different phases [Faconti & Massink, 2007]:

- planning
- ballistic
- approaching (under visual control)
- adjustment (under visual control, optional)
Drag’n’drop model

(SysPlanMove \mathcal{A} \{\text{interrupt,clickOK}\} \text{Interrupt})

Speak’n’drop model: User

Mieke Massink — CNR-Istituto di Scienza e Tecnologie dell’Informazione “A. Faedo” – p. 27/33

Mieke Massink — CNR-Istituto di Scienza e Tecnologie dell’Informazione "A. Faedo" – p. 28/33
S’n’D model: System & Interrupts

Parameter values

DnD:

- \(im = 1000/910\); time of planning (240 ms) plus ballistic (670 ms) movement
- \(vc_1 = 1000/290\); time of approach + adjust movement
- \(vc_2 = 1000/290\); as above (1000/120 for procedural case)
- \(in\); interrupt time variable
- \(pb = 1000/120\); time of completion of movement finishing with a push button
- \(sd = 1000/680\); time planning (0) and ballistic (680 ms)
- \(dd = 1000/120\); time to release (120 ms)
- \(ok = 1000/1300\); time needed to handle pop-up interrupt (1300 ms)

SnD:

- \(im = 1000/910\); time of planning initial movement plus ballistic movement
- \(vc = 1000/290\); time of visual control
- \(in\); interrupt time variable
- \(mc = 1000/80\); time of completion of movement finishing with a mouse click
- \(ss = 1000/630\); time for user to start speaking and completing the utterance
- \(es = 1000/1000\); time for user to end speaking (plus recognition and feedback)
- \(dd = 1000/120\); time to drag icon to trash and drop it there
- \(ok = 1000/1300\); time to handle pop-up interrupt
Resilience of D’n’D vs. S’n’D

Reward measure: $R(\text{drops}) = \sum_{t} C_{t} < 300$
Cumulative number of drop-actions over 300 seconds

Conclusions and outlook

Resilience to interrupts:
- Validation of results by means of empirical data
- Inclusion of error behaviour and mode confusion
- Further interaction techniques
- Bridge between ICO/Petri-Nets and stochastic reward model-checking

Modelling crowds in smart environments:
- Modelling arrival and behaviour patterns
- More sophisticated synthesizer programs
- Validation of the models by means of empirical studies
- Theoretical issues of Fluid Flow analysis

Thanks ReSiST project for support and funding and participants for their contributions
3rd International Workshop on

Formal Methods for Interactive Systems

2 November, 2009
Eindhoven
The Netherlands

http://homepages.cs.ncl.ac.uk/michael.harrison/fmis

Satellite of Formal Methods 2009 Conference

Organizers:

Michael Harrison, Newcastle University
Mieke Massink, CNR-ISTI, Pisa
Fault/intrusion REmoval through Evolution & Recovery

Final Workshop

March 2009

Outline

- Project: Team, Metadata and Summary
- Motivation
- The FOREVER Service
  - Architecture
  - Diversity Management
  - Prototype
  - Evaluation
- Conclusions and Future Work
- Internal Workshops & Publications
Project Team (Institutions)

- **3 ReSIST Partners**
  - Universidade de Lisboa (Portugal)
  - City University (UK)
  - Università di Pisa (Italy)

- **2 ReSIST Affiliates**
  - Universität Erlangen-Nürnberg (Germany)
  - Universidade do Estado de Santa Catarina (Brazil)

Project Team (Persons)

- Alysson Bessani @ Lisboa
- Alessandro Daidone @ Pisa
- Tobias Distler @ Erlangen-Nürnberg
- Ilir Gashi @ City
- Rüdiger Kapitza @ Erlangen-Nürnberg
- Rafael Obelheiro @ Santa Catarina
- Hans Reiser @ Lisboa
- Paulo Sousa @ Lisboa
- Vladimir Stankovic @ City
**Project Metadata**

- **Keywords**
  - Byzantine Fault Tolerance
  - ACM D.4.5 Fault-tolerance
  - Intrusion Tolerance and Resilience
  - Self-healing
  - ACM H.2.2 Recovery and Restart
  - Fault Removal
  - Design and Configuration Diversity

**Motivation**

- Byzantine fault-tolerant (BFT) replica coordination protocols are a fundamental component of intrusion-tolerant systems.

- Looking at BFT in terms of security:
  - We have to tolerate faults caused by a malicious and intelligent adversary, not faults that follow some statistical distribution.

- The main motivation for FOREVER are two assumptions typically stated on BFT papers:
  1. "The system is correct if at most $f$ out of $n$ replicas are faulty"
     
     *If an attacker can intrude $f$ replicas, he will potentially intrude one more if he has sufficient time.*

  2. "We assume fault independence (i.e., faults are uncorrelated)"
     
     *An attacker will try to find and exploit a vulnerability on some component that is used on every replica.*
Project Summary

- Goal: to develop a middleware service devoted to Fault/intrusiOn REmoVal through Evolution & Recovery
  - i.e., middleware service performing system recoveries (removing faults and/or intrusions) and patching the system over time letting it evolve wrt vulnerabilities
  - This service can be used to enhance the resilience of replicated systems, namely those that can be affected by malicious attacks

- Addresses some research gaps identified in ReSIST D13 deliverable, namely:
  - GE1: Evolution of Threats
  - GD1: Diversity for Security

- Three main tasks
  - T1: Definition of the FOREVER service architecture
  - T2: Analysis of how diversity can be managed
  - T3: Evaluation of the FOREVER service

The FOREVER Service (1)

- Recovery
  - Time-triggered periodic recoveries
    - Every replica is rejuvenated periodically
  - Event-triggered reactive recoveries
    - When malicious behavior is detected or suspected

- Evolution
  - Recovered replicas are different from previous incarnations
    - operating systems are changed
    - configuration diversity rules are applied (e.g., password change, port randomization)
The FOREVER Service (2)

Hybrid model and architecture

Internet (clients, attackers, ...)

- Fault/Intrusion-Tolerant application
- BFT replication library
- FOREVER

Can be compromised!

Cannot be compromised!

Diversity Management

- **Offline** diversity generation
  - Pool of pre-built OS images (e.g., Linux, OpenBSD, Solaris)
  - Different OS image started in each recovery
  - FOREVER selects the OS image that is less similar than the OS images running in the remaining replicas

- **Online** diversity generation
  - FOREVER applies a set of configuration diversity rules to the selected OS image
Similarity between OSs

- Based on vulnerability data collected from the NIST National Vulnerability Database (NVD) [http://nvd.nist.gov]
  - 1999-2007
  - 7 different operating systems

Configuration Diversity Rules (examples)

- Address Space Layout Randomisation (ASLR)
  - randomizes the memory location of programs data and code in each recovery
  - reduces the probability of a successful buffer overflow attack (one of the most serious security threats)

- Port Randomization
  - randomizes network port numbers in each recovery
  - an attacker needs to find out on which port a service is running before he can access it
    - even if he discovered it in the past!

  **Ongoing attacks need to be restarted after a recovery!**
The FOREVER Prototype

- Hybrid architecture implemented using a virtual machine hypervisor (Xen)
  - FOREVER monitors run in a privileged domain (dom0)
  - Application replicas run on a non-privileged domain (dom1)
- File system repository (FSR) on dom0

Evaluation (1)

- We conducted a preliminary assessment of the FOREVER service
- Goal: to evaluate the probability of overall system failure when some parameters are varied:
  - time between recoveries
  - (replicas) fault rate
  - probability of common vulnerabilities
  - effectiveness of configuration diversity rules
Evaluation (2)

Main results of model-based evaluation:
- Recoveries help in lowering down failure probability
- Running diverse OS in the replicas offers a tenfold improvement in security
- Configuration diversity rules decrease failure probability

Conclusions

BFT systems rely on two “problematic” assumptions:
- At most $f$ faults can happen
- Different replicas do not share the same vulnerabilities

FOREVER service aims to improve the coverage of these assumptions in order to make BFT replication both intrusion-tolerant and intrusion-resilient

FOREVER uses online and offline diversity generation mechanisms
- Offline: pool of pre-built OS images + similarity engine
- Online: configuration diversity rules

Preliminary model-based evaluation shows effectiveness of FOREVER
Future Work

- **WAN replication**
  - Degraded service with a partial synchronous FOREVER

- **Improved Similarity Engine**
  - Extend NVD analysis to take into account
    - other software packages
    - vulnerabilities type, severity, access vector, …

- **Prototype**
  - Implement fully-fledged prototype and release as open source

- **Experimental Evaluation**
  
  *To be addressed in a long-term project! (we hope)*

Internal Workshops

- **1st Workshop @ Lisboa, Portugal**
  - 19-20 February 2008
  - 7 participants
  - 3 technical sessions, total of 6 presentations

- **2nd Workshop @ Erlangen-Nürnberg, Germany**
  - 14-15 July 2008
  - 8 participants
  - 3 technical sessions, total of 5 presentations

- **3rd Workshop @ Firenze, Italy**
  - 14-15 October 2008
  - 8 participants
  - 3 technical sessions, total of 5 presentations
Publications

- **The FOREVER Service for Fault/Intrusion Removal**
P. Sousa, A. Bessani, R. Obelheiro

- **Efficient State Transfer for Hypervisor-Based Proactive Recovery**
T. Distler, R. Kapitza, H. P. Reiser

- **FOREVER: Fault/intrusiOn REmoVal through Evolution & Recovery**

- **On the Effects of Diversity on Intrusion Tolerance**
A. Bessani, R. Obelheiro, P. Sousa, I. Gashi

- **Enhancing Failure / Intrusion Tolerance through Design and Configuration Diversity**
A. Bessani, A. Daidone, I. Gashi, R. Obelheiro, P. Sousa, V. Stankovic
Submitted.

Thank You!
http://forever.di.fc.ul.pt/
Honeypots: malicious fault characterization exploiting honeypot data

Corrado Leita
Olivier Thonnard
Jouni Viinikka

Vladimir Stankovic
Ilir Gashi
Urko Zurutuza
Marco Serafini

Challenges

Quantitative data + Analysis tools = Knowledge
Getting quantitative data

• Honeypots: “information system resource whose value lies in unauthorized or illicit use of that resource” (Spitzner)
• The main challenge: monitoring the “Internet weather” is a complex task

SGNET

• Distributed honeypot deployment
  – 30 sensors deployed in different networks all around the world
  – Partnership open to anybody interested

• What makes it “different”:
  – Protocol agnostic approach (ScriptGen): we do not assume to know a priori what we are going to face
  – Oriented to code injection attacks: exploitation of software vulnerabilities to take control of a victim
    • Common propagation vector for self-propagating malware
    • Allows to collect malware samples
SGNET data collection framework

Mining the data

How to identify interesting events? What is their impact?

- Event identification (*RAID 2008*)
  - Identify interesting events/anomalies
  - Correlation: is an event witnessed on multiple sensors? Why?
- Attack impact (*submit at IEEE NCA09*)
  - How “dangerous” are these activities?
  - How do modern AV products perform in detecting the downloaded malware?
Event identification

- Combination of
  - Clustering techniques developed in EURECOM
  - Time series analysis techniques developed by Orange Research for IDS alert logs

Identified challenges

- Problems
  - Inertia: big peaks “mask” smaller ones
  - “False positives”: identification of minor activity peaks in the middle of the activity period

- Lesson learnt:
  - The characteristics of the time series are different from typical IDS alert sequences
  - Possible ways to circumvent these problems
Malware and AV detection

• How to benchmark AV engines?
  – **Complexity problem**: the engines exploit diversity using different analysis techniques to detect malware. Not all components can be easily evaluated (example: behavioral detection)
  – **Labeling problem**: it is a difficult (impossible?) task to determine the correctness of an alert
    • Example: given malware M, if a detector classifies it as N, is it correct?
    • How can I know that a malware is M in the first place??
  – **Ambiguities**: should a corrupted malware sample be recognized?

• Analysis simplifications:
  – Consider solely the signature-based detection engine
  – Consider detection as binary: any alert is a success
  – Filter out corrupted binaries

SGNET dataset and AV detection

• Automated interaction with VirusTotal
  – On the download day, the sample is analyzed with the most up-to-date version of the AV signatures

• Submission policy
  – Each sample is submitted multiple times to VirusTotal
    • At least 30 days
    • Stop condition: last 7 reports are identical
  – Evolutionary view on the detection rate
    • How long does it take to detect a previously undetected sample?

• Analysis carried out on 1599 malware samples downloaded by SGNET over a period of 8 months
Malware “difficulty” (over 20 days)

The malware (59 total) - ordered by their lowest to highest difficulty / failure rate

Failure in all 20 days
Evolutionary changes
Success in all 20 days

“Temporal clustering”

- Cluster together vendors exhibiting similar temporal behavior in their detection rate
Regressions

- How did the detection ability evolve?
- The expected case (0 to 1) was not the only one observed
- We identified a considerable number of “regressions” (1 to 0)
  - Possible reason: signature pulled back because of false positives
- Can we eliminate these regressions through diversity?

<table>
<thead>
<tr>
<th>AV Name</th>
<th>Number of Malware the AV regressed on</th>
<th>Number of instances the AV regressed on</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV-20</td>
<td>586</td>
<td>1691</td>
</tr>
<tr>
<td>AV-8</td>
<td>374</td>
<td>538</td>
</tr>
<tr>
<td>AV-3</td>
<td>71</td>
<td>78</td>
</tr>
<tr>
<td>AV-11</td>
<td>58</td>
<td>59</td>
</tr>
<tr>
<td>AV-1</td>
<td>37</td>
<td>235</td>
</tr>
<tr>
<td>AV-32</td>
<td>36</td>
<td>38</td>
</tr>
<tr>
<td>AV-2</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>AV-16</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>AV-9</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>AV-14</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

1oo2 evaluation

- What is the performance improvement obtained by combining together two vendors?

**96 pairs** reach 0% failure rate

**163 pairs** reach a failure rate better than any engine considered alone
Perspectives

• The “honeypots mini-project”
  – Many institutions and research backgrounds
  – Integration of different backgrounds, research perspectives
  – Some interesting results…
  – … and even more open doors to future research!

Thank You!

Corrado Leita
Corrado_Leita@symantec.com
Project

- Members
  - University of Lisboa
  - University of Pisa
  - Deep Blue

- Opportunity for multi-disciplinary collaboration
  - Distributed Systems (Dependability)
  - Air Transportation

- Goal: apply distributed system models and techniques to devise dependable solutions for decentralized air traffic management
Outline

- Application Scenario: Air Traffic Management (ATM)
- Current Approach in ATM
- An alternative approach: Airborne Self-Separation
  - Satisficing Game Theory (SGT)
- Evaluation of SGT
- The RAPTOR Architecture
- Conclusion

Application Scenario: Air Traffic Management

- Air Traffic Management (ATM) is the dynamic and integrated management of air traffic flow to minimize delays and congestion while guaranteeing safety and efficiency of operation in airspace.

- ATM presents a wireless operational environment with strict safety requirements
  - Failures can result in catastrophic consequences

- Provides an opportunity to address some ReSIST research gaps within a specific real-world application scenario
  - GE3 Distributed System Models
  - GE9 Complexity and Self-Organization
  - GA8 Evaluation of Dynamic Systems
  - GA10 Trust and Cooperation
Current Approach in Air Traffic Management

- Based on rigid off-line flight planning
  - airspace statically divided into sectors and airways
  - Air Traffic Controllers (ACTOs) are the central authority within each sector
- Heavy Reliance on ATCOs
  - Controllers’ skills are a fundamental factor
  - Little to no autonomy for aircraft crews

Problem Statement

- Current approach does not scale and is close to saturation

- With the increase in air traffic worldwide, future generation of ATM will require more automation and sophisticated decision support tools to solve conflicts and improve global system performance
An alternative Approach: Airborne Self-Separation

- Pilots can choose the route of the aircraft at run-time
  - Scalable
  - Economic
  - Convenient

- Must be supported by appropriate technologies and procedures
  - Aircraft are already equipped with a communication system which allows aircraft to exchange messages useful to assist the flight crew
  - What is missing is a reliable & decentralized procedure for conflict detection and resolution
    - A conflict, within our context, is any two or more aircraft who come within an unsafe distance of each other

Decentralized Procedure for Automated Conflict Detection and Resolution

- Requirements
  - Aircraft must coordinate their maneuvering to prevent collisions
  - Maneuvers must ensure an overall traffic optimization, in terms of aircraft trajectories and global delays
  - The solution must scale to high traffic densities

- A Solution
  - Satisficing Game Theory (SGT), supported by appropriate services for a robust and dependable system
Satisficing Game Theory in ATM

- Independent, Collaborative, Adaptive Agents are used to model Air Traffic

- Agents exchange their state with other agents in the proximity radius, and apply a deterministic algorithm to guarantee conflict resolution and to optimize the overall traffic flow

The SGT Algorithm

- At each time step, each aircraft will exchange information about its position, current direction, destination, flight time and delay with the neighbouring aircraft.

- These information are used to calculate selectability and rejectability functions

- Selectability considers the benefits of a given direction in terms of the aircraft goals and of the overall traffic optimisation

- Rejectability functions considers the costs of a given direction in terms of potential safety problems

- Each aircraft chooses the direction that maximises the difference between selectability and rejectability
Pros and Cons of Satisficing Game Theory

- **Pros**
  - Cooperative approach that optimizes the overall traffic, while ensuring conflict detection and resolution.
  - The decentralized, distributed, and automated nature of the approach ensures good scalability.

- **Cons**
  - Strong assumption on communication services: synchronized communication and without any kind of error.

Some Evaluation Results

- We are evaluated the system in Omnet++ for different scenarios and test cases, gathering insights on SGT behaviour, when real-world aspects, such as transmission delays, packet loss, and different types of maneuvering options.

- Neighbourhood size perceived by an aircraft during the seconds preceding a conflict.
- The trace fluctuates, which means that some aircraft were suddenly disappearing from the point of view of the considered aircraft, since the wireless communication is inherently lossy.
Some Evaluation Results

- In order to contrast the effects of message loss, we instrumented SGT such that the position of neighbouring aircraft could be approximated by using also the most recent received messages.
- Additionally, we allowed aircraft to perform sharper direction changes (upto 10 degrees per time unit).

Number of incidents during ten simulations for different aircraft settings: aircraft settings 1 and 2 allow direction changes up to 5 degrees, while setting 3 and 4 allow direction changes up to 10 degrees; setting 2 and 4 approximate the position of neighboring aircraft by using also the most recent received messages.

Failure Scenario

- Aircraft are ranked based on their state. This ranking determines who must maneuver around whom.
- If two aircraft in a collision course have incomplete or outdated information about each other, it is possible for each of them to calculate contradictory rankings that, in turn, may lead them into maneuvering decisions that further puts them into a conflict.
- Additionally, may the communications subsystem of one of them fail, even if only temporarily, before the information about each other is harmonized, it is possible that a collision happens since both of them could be convinced that it is responsibility of the other one to maneuver around.
Gap between SGT assumptions and the environment

- Strong assumptions: aircraft have consistent and fresh information
  - Synchronous
  - Reliable

- Wireless environments are not reliable
  - Noise, fading, interference, etc.
  - Messages can be lost or corrupted

- A system model that considers unreliable communication links
  - Synchronous (GPS makes possible clock synchronization with enough accuracy)
  - Unreliable links

The RAPTOR Architecture

Agreement Protocols
- Terminating Reliable Broadcast
- Multi-Valued Consensus
- Binary Consensus

Services for Airborne Self-Separation
- View Augmentation
- Rank Consistency
- Group Membership

Conflict Resolution Algorithm
- SGT
Agreement Protocols

- The system is modeled as a set of $n$ processes (i.e., the aircraft) that exchange information in synchronous steps.

- In order to capture the transient nature of faults in wireless environments, it is determined that the transmissions of up to $f$ processes per round may be faulty where $n = 3f + 1$. (Future protocols are unrestricted in terms of fault source distribution)

- This includes both omission faults (where a message is lost) and corruption faults (where the contents of a message are changed).

Protocols

- Binary Consensus: agreement on a binary value
- Multi-Valued Consensus: agreement on a value from an arbitrary domain
- Terminating Reliable Broadcast: all processes receive the same message

The RAPTOR Architecture

Agreement Protocols

- Terminating Reliable Broadcast
- Multi-Valued Consensus
- Binary Consensus

Services for Airborne Self-Separation

- View Augmentation
- Rank Consistency
- Group Membership

Conflict Resolution Algorithm

SGT
Services for Airborne Self-Separation

- **Group Membership service**: based on the aircraft geographic distribution at each instant, organizes aircraft into groups
- **Rank Consistency service**: ensures a consistent ranking of the aircraft (from an SGT perspective) within each group
- **View Augmentation service**: provides a consistent view of the adjacent groups of aircraft

Conclusions

- We explored the possibility of enhancing the resilience of an algorithm based on Satisficing Game Theory (SGT) for distributed conflict resolution and traffic optimization in Air Traffic Management.

- While evaluating SGT in Omnet++, we obtained insights on the reliability of the approach (or lack thereof), and pointed out the shortcomings when introducing real-world constraints, such as unreliable communication.

- A fault-tolerant architecture was designed to obtain a more robust system. We propose a layered approach to develop an effective and dependable conflict resolution system for Airborne Self-Separation.
Publications


- ‘A Distributed Systems Approach to Airborne Self-Separation’. Book Chapter for ‘Computational Models, Software Engineering and Advanced Technologies in Air Transportation: Next Generation Applications’, to be published in 2010 by IGI Global (accepted for publication)

Testing in Mobile Settings (TMS)

Zoltan Micskei (BUTE), Minh Duc N’Guyen (LAAS), Nicolas Rivière (LAAS), Hélène Waeselynck (LAAS)

Mobile computing systems

- Dynamicity of system structure
  - Involved nodes, connectivity
- Communication with unknown partners in a local vicinity
- Context awareness
  - Policies to update the view and react to contextual changes
Testing: state of the art

- Traditional distributed systems
  - Platforms with dedicated test interfaces, dedicated test languages (TTCN-3)
  - Use of graphical scenario languages (MSC, UML SD) to support design & validation activities
  - Formal approaches in the protocol community
    - SDL model x test purposes → test cases
  - Passive testing approaches

- Mobile computing systems
  - Experimental platforms with simulation facilities (mainly for evaluation purposes)
  - Testing issues have been little explored so far
  - Pioneering work based on SDL models (but SDL is not well-suited to mobile settings)
  - No established modeling framework for mobile computing systems

  In TMS, investigation of scenario-based approaches

Scenario-Based Testing

- Requirement scenarios: capture key properties
- Test purposes: behavior to be covered by testing
- Test cases: interactions of test components and SUT, verdict assignment
- Test execution traces: actual, monitored traces
Overview of the mini-project

- Definition of extensions to current test scenario languages
  - Example: UML 2.0 Sequence Diagrams

- Development of automated treatments for test scenario descriptions
  - Graph matching problems
  - Semantics of UML 2.0 Sequence Diagrams

- Conclusion and perspective

Interaction scenarios in mobile settings?

- Current languages: focus on the partial order of communication events
- But the underlying spatial configuration is equally important to characterize scenarios in mobile settings
- Absence of broadcast constructs
- How to represent broadcast in a local vicinity (e.g., « hello » messages for group discovery)?
Example of usage: requirement scenarios

Does the test trace fulfill the requirement expressed by the scenario?

1. Determine which physical nodes of the trace match the nodes specified in the spatial view.
2. Analyze the order of events in the identified configurations.
Overview of the mini-project

- Definition of extensions to current test scenario languages
  - Example: UML 2.0 Sequence Diagrams

- Development of automated treatments for test scenario descriptions
  - Graph matching problems
  - Semantics of UML 2.0 Sequence Diagrams

- Conclusion and perspective

Basic facility: graph homomorphism building

Does G1 appear as a subgraph of G2?

Build a graph homomorphism from G1 to G2

- Has been extensively studied in the literature
- Including for graphs with:
  - Tuples of labels, e.g. node < "140.93.5.235", 1, 5>
  - Label variables, e.g. node < x, 1, 5>

Tool from [Guennoun et al.] mapping of nodes + valuation that unifies the labels
Reasoning on sequences of graphs

- Our need: search for a sequence of configuration patterns in a concrete trace

Patterns (from the scenario)  \[ P_1 \rightarrow P_2 \rightarrow P_3 \]
Concrete configs (from the trace)  \[ C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow C_5 \rightarrow C_6 \rightarrow C_7 \rightarrow C_8 \]
(Note: a pattern \( P_i \) may occur in several \( C_j \) before the config changes to \( P_{i+1} \))

- A match is defined as:
  - A valuation for all variables in the patterns (including symbolic node ids)
  - Start & end dates for the successive configurations in the trace

Implementation of a tool: GraphSeq

GraphSeq (1)

- Ensures consistent valuation choices throughout a sequence

If variable \( x \) appears in \( P_1, P_3 \), it must keep the same valuation in \( C_2, C_3, C_5, C_6, C_7 \)

Patterns  \[ n_1 \]
Trace  \[ 1 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow \cdots \]

- Accounts for nodes that appear and disappear

Transition \( P_1 \) to \( P_2 \) may be detected at \( C_3, C_4 \) or later

If matching is \( n_1 := 1 \)

2 cannot match \( n_2 \) (2 is not new)
GraphSeq (2)

- Temporal window of the match is maximal
  
  GraphSeq does not return this match (not maximal)

- First experimentation with GraphSeq
  - Validation with 900 randomly generated sequences
  - Analysis of traces from a location-based GMP case study
  - Connection to a mobility simulator ([Bai et al.], Univ. South California)

Overview of the mini-project

- Definition of extensions to current test scenario languages
  - Example: UML 2.0 Sequence Diagrams

- Development of automated treatments for test scenario descriptions
  - Graph matching problems
  - Semantics of UML 2.0 Sequence Diagrams

- Conclusion and perspective
Goal: analysis of the event view

Does the test trace fulfil the requirement expressed by the scenario?

1. Determine which physical nodes of the trace match the nodes specified in the spatial view
2. Analyze the order of events in the identified configurations

UML SD Semantics?

- Problem does not originate from our mobility-related concepts...
- ... But from the core UML SD constructs
- Informal semantics in the OMG specification
  - Scattered throughout the text
  - Unclear meaning of some operators
- Semantics variation points allowing specialization to target domain of usage
  - Not always explicit where the variation points should be...
- Nothing such as « the » semantics of UML SD!!!
Formal semantics

<table>
<thead>
<tr>
<th>Name</th>
<th>Reference</th>
<th>Formalism</th>
<th>Years</th>
<th>Comments / Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAIRS</td>
<td>[17]</td>
<td>traces of events, transitional systems</td>
<td>2003-2007</td>
<td>Implemented in Maude</td>
</tr>
<tr>
<td>Cavarra and Filipe</td>
<td>[9]</td>
<td>ASM</td>
<td>2004</td>
<td></td>
</tr>
<tr>
<td>Grosu and Smolka</td>
<td>[18]</td>
<td>Büchi automaton</td>
<td>2005</td>
<td></td>
</tr>
<tr>
<td>Hammal</td>
<td>[19]</td>
<td>partial orders</td>
<td>2006</td>
<td>synchronous systems, S2A tool</td>
</tr>
<tr>
<td>CPN</td>
<td>[16]</td>
<td>Colored Petri nets</td>
<td>2007</td>
<td>synchronous systems</td>
</tr>
</tbody>
</table>

Overview of 12 semantics

Categorization of the semantics choices

- **Interpretation of a basic chart**
  - what is a trace?
  - complete / partial traces

- **Introducing operators (CombinedFragments)**
  - weak sequencing as the default composition operator
  - synchronization on entering and exiting fragments

- **Computing partial orders**
  - General Approaches: interleaving semantics vs. true concurrency, partial orders are explicitly given (automata, event structures) or not (rules to generate traces)
  - (Guarded) choices: non local choice, well-definedness of predicates, when to evaluate guards?

- **Introducing gates**
  - ill definedness problems, in-lining vs. composition

- **Interpretation of conformance-related operators**
  - Assert/Negate
  - Ignore/Consider
  - Nesting of operators
  - Traces that are both valid and invalid
Example

Interpretation of conformance-related operators: Negate

- The trace is both valid and invalid (e.g., Knapp, STAIRS)
- The trace is invalid (e.g., MSD)
  - Neg is syntactic sugar for a global false condition at the end of the fragment
- Definition of alternative operators to express forbidden behavior: not (Knapp), refuse (Lund)
- Syntactic restrictions on the use of Neg: should be used only at the top level (Störrle)

Outcome of the review of the semantics

- Structured view of where the choices are, and what the alternatives consist of
- Can be used as a guide for choosing a semantics suitable for a target domain
- Allowed us to define TERMOS (Test Requirement language for Mobile Settings)
  - Syntactic restrictions to UML SD + interpretation choices
Conclusion and perspective

- No established modeling framework to support model-based testing of mobile computing systems

- Our investigation: interaction scenarios in mobile settings
  - Spatial configuration must be a first-class concept ...
  - ... which yields graph matching problems (GraphSeq tool)
  - Close look into the semantics of UML SD (allowing us to propose a semantics well-suited for our purpose)

- Perspective: enrich the spatial view
  - Min/max duration constraints for the configurations
  - Constraints on the valuation of configuration variables
  - ... Any other extension to enrich the representation of the context?
Objectives

Focus: methods to reason formally about large-scale ubiquitous systems

- complexity issue
- scalability (cf. state explosion problem)

Objective: investigate scaling techniques for

- temporal logic model checking and
- CSP refinement checking

using

- abstraction techniques and
- compositional reasoning

in the context of wireless sensor networks (WSNs).
Context

Context of this work are wireless sensor networks. Characteristics include:

- system composed of a large number of sensor nodes
- nodes with limited computing capabilities
- densely deployed, position typically not predetermined
- unreliable, bounded-range communication
- ad-hoc networking techniques, multi-hop communication
- frequent topology changes due to link failures
- self-organising capabilities required for protocols

Typical task:

- transportation of sensor data to a base station

We have chosen the Surge routing algorithm as a vehicle for our studies.

Surge routing protocol

Principle of operation:

- route information service: nodes periodically communicate to neighbours their own distance to the base station
- message sending service: nodes send data to the "best" neighbour (shortest distance / best link quality)

Top-level properties (under ideal conditions):

- Surge establishes and maintains a spanning tree rooted at the base station
- all messages sent will reach the base station eventually, i.e. within bounded time
Topics of this presentation:

1. modelling aspects: structure and abstractions
2. model-checking analyses: symbolic, bounded, induction
3. CSP analyses: refinement checking, assumption-commitment

Modelling of the Surge algorithm

We have developed a series of models for Surge:

- state-based models for temporal logic model checking (in SAL)
- event-based models in CSP for refinement checking (with FDR)

These models focus on different aspects intended to be complementary. Initially:

- modular model in SAL, to prove properties of individual layers: application / routing / data link
- CSP model to combine layer properties using assume-guarantee

In the course of the project we moved to a service-oriented view:

- SAL to prove properties of Surge routing service
- CSP to combine properties of routing and sending service
Abstraction aspects

Models employ several abstractions, e.g.
- node IDs and sets of neighbours to model position and connectivity
- scheduler, or more generally, a “sending oracle” to abstract from collision avoidance service of the MAC layer

some may be specific for Surge, such as
- abstraction from link quality
- decomposition into route information and sending services

but others can be used generally for WSNs, or grid topologies, e.g.
- models of layers of the protocol stack
- sending / scheduling of messages
- classification of neighbours “closer to base station” ("left-or-up")
Temporal-logic model checking

We used the SAL suite of model checkers (of SRI Intl.)
- symbolic model checker (BDDs)
- bounded m/c (SAT-based), also for infinite state space

Focused on
- Surge's route information (spanning-tree) service (STS)
- grid topologies

STS execution at nodes modelled as rounds, divided into steps
- reception of messages: new best next hop?
- message sending: announce own distance, when new
composed (synchronously) with model for message distribution
- gather all messages from nodes that send
- deliver one msg after the other to respective neighbours

Properties

Top-level property to be checked:
- Surge establishes a spanning tree rooted at the base station
  divided into two parts
- either there is a node w/o best next hop, or the best next hops form a spanning tree

\[
\text{sts : THEOREM} \\
\text{system} \models \ G( \exists \text{mote} : \text{mote} \in \text{Motes} ) : \text{nexthops}[\text{mote}] = \text{mote} \text{ OR} \text{is\_spanning\_tree}(\text{nexthops}, \text{is\_neighbour}) ;
\]

- eventually, all nodes have chosen a best next hop

\[
\text{eventually\_all\_nodes\_know\_hop : LEMMA} \\
\text{system} \models \ F( \forall \text{mote} : \text{mote} \in \text{Motes} ) : \text{nexthops}[\text{mote}] \neq \text{mote} ;
\]
Symbolic model checking results

Symbolic model checking for small grid sizes
- liveness part proved for $3 \times 3$ and $4 \times 4$ grids in seconds and minutes, resp.
- safety part proved in seconds for $2 \times 2$ grids, but memory exhaustion for larger sizes
- alternative formulation for `is_spanning_tree` only traded memory for run-time

Introduced weaker property, instead of `is_spanning_tree`: distances decrease along best next hops.

```
FORALL (i:Mote):
  (nexthops[i] = basestationID AND mydists[i] = 1) OR
  (nexthops[i] /= i AND mydists[i] > mydists[nexthops[i]])
```

No real improvement though for symbolic model checker.

Bounded model checking

Bounded model checking
- based on SAT-solving
- refutation method

Bounded model checking with *k*-induction
- generalises traditional induction:
  - prove $P$ for a sequence of $k$ states starting from an initial state
  - prove that if $P$ holds for $k$ successive states (starting from an arbitrary state) then $P$ holds also for state $k + 1$
- proof method
- $k$ can grow exponentially with state space

Proved weak safety property for $2 \times 2$ grid
- however, needed $k = 18$
- $k = 60$ insufficient for $3 \times 3$ grid
Conjoin the Surge model with an abstractor and a monitor model.

Abstractor model:
- define predicates that describe certain configurations of Surge
- announce at each state which predicates hold
- formulate predicates such that each implies the desired property

Monitor model:
- consists of abstract states corresponding to each configuration
- transitions describe the admissible steps from one configuration to another
- it moves into a special bad state in unforeseen situations

Proof idea: show that system never reaches the bad state.

Objective: avoid exponential growth of $k$, by using this property as an additional requirement in the $k$-induction proof.

Disjunctive Invariants: results

We defined an abstractor and a monitor model for Surge
- using (only) 3 configurations
- configurations roughly correspond to different stages of an execution of Surge: no node has best hop, some have, all have
- more fine-grained model currently in progress

Succeeded to prove “never-bad” property, and weak correctness
- with $k = 1$
- in particular $k$ does not increase with system size
- for up to $5 \times 5$ grids
- simpler properties can be proved for grid sizes up to $15 \times 15$

Can even handle arbitrary topologies (with reasonable properties) of up to 8 nodes.
CSP refinement checking

Initial CSP model for STS developed by manual translation of SAL model
- systematically, to obtain broadly equivalent CSP process
- differences where alternative modelling suited CSP better

Applied optimisations to the model
- to reduce compilation effort
- to reduce reachable state space, e.g.
  - by removing insignificant interleavings of message deliveries
  - by enforcing broadcast events to occur in defined order

Introduce different form of STS correctness property
- best next hop must be one hop closer to base station
- \textit{left-or-up} in a grid, when base station is top left
- yields stronger property: spanning tree must be minimal

CSP refinement checking: results

Check safety and liveness properties for STS
- weak correctness with minimality property:
  - if each node has settled to a choice of best next hop, these choices are all \textit{left-or-up}
  - eventually, all nodes settle to a choice of best next hop

Properties were checked for
- static grid topologies
- grid sizes up to $10 \times 10$

Run times in the range of several hours (for largest size considered)
**Assumption / Commitment reasoning**

![Diagram](image-url)

\[ COM \subseteq SYS \parallel ASS \]

- **principal goal**: investigate whether A/C facilitates checking WSN properties
- **hope to exploit** Surge structure for compositional reasoning
Assumption / Commitment reasoning

If

- $COM_1 \subseteq_T SYS_1 \parallel ASS_1$ and $COM_2 \subseteq_T SYS_2 \parallel ASS_2$
- $ASS_1 \subseteq_T ASS \parallel COM_2$ and $ASS_2 \subseteq_T ASS \parallel COM_1$
- (and some applicability conditions)

then

$$(COM_1 \parallel COM_2) \subseteq_T (SYS_1 \parallel SYS_2) \parallel ASS$$

Applying A/C to Surge

To apply A/C to Surge, we

- developed model of sending service (SS)
- defined the various assumption and commitment processes
  - chose overall assumption True, represented by Run (all events)
  - commitment of sending service (message will be delivered)
    needed to be formulated as a safety property
    - we used: “there will never be a loop in the message transmission”
  - instantiated the theorem, and carried out the checks

Results for the weak correctness safety property

- succeeded in proving both component A/C properties
- and most of the side conditions
- however, state space of $SYS_2 \parallel ASS_2$ (sending service) grew very quickly
Improving A/C formulation

To improve tractability
- introduced strengthened assumption \( \text{ASS}_2 \) for sending service
  - assume best next hop choices occur in diagonal order (closest to base station first)
  - strengthen commitment \( \text{COM}_1 \) accordingly
- introduced auxiliary process to avoid dead states

Effect was limited, so developed a helpful abstraction of SS
- rather than record ID of best next hop, record its “direction”
  - \textit{Self}, \textit{LeftUp}, or \textit{Other}
- defined straightforward abstraction and concretisation functions from IDs to direction classes
- confirmed validity by checking \( \text{SS}'(i) \subseteq \text{SS}(i) \) for every node \( i \)
- validity seems to hold \textit{by construction} – on-going formalisation to prove this

A/C results

Results of applying A/C reasoning to \textit{Surge}
- checked the weak correctness safety property as before
- but with hardest property \( \text{COM}_2 \sqsubseteq_T \text{SYS}_2 \parallel \text{ASS}_2 \) much easier
- feasible grid sizes upto \( 10 \times 10 \)
- can do \( 15 \times 15 \) in about 30 mins for easier parts (i.e. not STS A/C property or liveness property)
- potential for other techniques to address the STS properties more efficiently, hence effective combination of techniques
Conclusions

Investigated scaling techniques for m/c and refinement checking
- in the context of WSNs
- using abstraction and compositional reasoning

SAL and CSP models for Surge, employing various abstractions
- some are specific to Surge
- others are generally useful for WSNs / grid topologies

SAL-based model checking
- scalability of k-induction achieved with disjunctive invariants
- configurations constitute abstract stages of Surge execution

CSP assumption-commitment reasoning
- successfully applied A/C theory to prove weak correctness
- applied according to functional decomposition
- employed technique to abstract component model
  - for improved scalability
  - not restricted to WSNs
- combination of A/C with CSP compositional reasoning

Future work

A lot is left to be done
- improve / generalise abstractions for modelling
- extend abstractor / monitor approach to other algorithms
- investigate ways to apply A/C to liveness property
- develop approaches to combine SAL and CSP proofs
- extend models to allow non-grid topologies
- ... and topology changes
- consider other WSN aspects, e.g. link quality, energy consumptions
- ...