ReSIST: Resilience for Survivability in IST

A European Network of Excellence

Contract Number: 026764

Deliverable D23: Knowledge base version 2

Report Preparation Date: December 2007
Classification: Public

Contract Start Date: 1st January 2006
Contract Duration: 36 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 Ciencas 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

Information Society Technologies

SIXTH FRAMEWORK PROGRAMME

Contents

1 - Introduction

“Deliverable D8: Draft planning for the next 18 months” states:

> **Deliverable D23: Knowledge base version 2**
>
> RKB version 2. A mature development of the RKB, reflecting the consolidation and maintenance performed by IT-T1, and the extensions to resilience-explicit computing support incorporated in the RKB and ontology, and from the development of challenge problems. Prepared by task IT-T1, with support from IT-T2 and IT-T3.

It further states:

> **Task IT-T1: Resilience Knowledge Base (RKB)**
>
> The plan for the Resilience Knowledge Base was to provide an integrative infrastructure to the ReSIST Project and the wider community. The aim was to deliver convenient access to extensive information from across Europe, in particular about people, projects, institutions, events, publications, etc., of relevance to system resilience researchers in both the dependability and the security domains, and to their sponsoring organisations.
>
> Deliverable D10 (month 12) provided a prototype of the RKB, which clearly showed the potential of the expected final system and in practice went further than was originally planned. The range and size of data sources that were included was an order of magnitude greater than had been expected at this stage, and facilities for adding specialist information, such as courseware information, had been made available to the project. The task also provided a project wiki with a semantic infrastructure.
>
> Specifically, there is now an RKB that has data from sources covering the whole of Computer Science and related topics, both within Europe and beyond. The “Explorer” interface was well received by project members who conducted a review, although a number of possible improvements were identified. Details of the prototype RKB can be found in deliverable D10.
>
> Task IT-T1 is now concerned with gaining value from D10, as well as supporting the resilience-explicit and resilience ontology activities.
>
> The aims of the next phase of the RKB work are twofold:
>
> 1. Maintenance and Consolidation
>
> Content – The resources needed to maintain the RKB information content rise with its magnitude; the success of the first phase has delivered much more content than was envisaged initially, and there are significant additional sources that are targeted for acquisition (IEEE material is already identified).
>
> Software – software systems need to be kept up to date; this relates to operating systems, web server software and triplestore system software.
>
> Hardware – it can be expected that the demand load on the RKB will rise, which may necessitate work to enhance the hardware base (with some more hardware purchased); the much less attractive alternative would be to restrict access and usage.
>
> Specific – Special requirements in the first phase (the wiki, the coursework page, etc.) imposed additional tasks and it is recognised that more work of this type will be needed as the Network continues to make progress; for example, a need has already been identified for a page to automatically gather the Resilience Mechanism Questionnaire data.
>
> 2. ResEx and ResOn support
>
> Support for the next phases of resilience-explicit work was planned initially and will continue, but as the new resilience ontology strand develops, the partners will need further support on both *acquisition* and *publishing*. Acquisition involves adding metadata about components and processes and their attributes to the RKB, as well as related documents. Appropriate ways to publish this data, to users and systems that are engaged in Res-Ex processes, will need to be devised

> At the prototype stage (i.e. D10), the extent to which resources in the RKB are classified according to resilience-related function is less specific than is desirable, and the RKB can only classify resources where other systems have identified concepts of interest to them that are related to resilience. To improve on this, the deployment of a system that classifies documents according to their relation to the detailed resilience terms embedded in the RKB would be required. Deploying such classification technologies is a significant task, but would yield considerable additional value. The new resilience ontology task has been created to identify and refine resilience concepts, and is also exploring the challenge of deploying appropriate classifiers, albeit from outwith the ReSIST Project. Supporting the emerging classifications in order to gain greatest benefit from this work, and any classifying systems that use them, will be an exciting challenge.

As it was with D10, the primary components of deliverable D23 are the RKB and associated service pages, and this document discusses their content.

2 – RKB Content

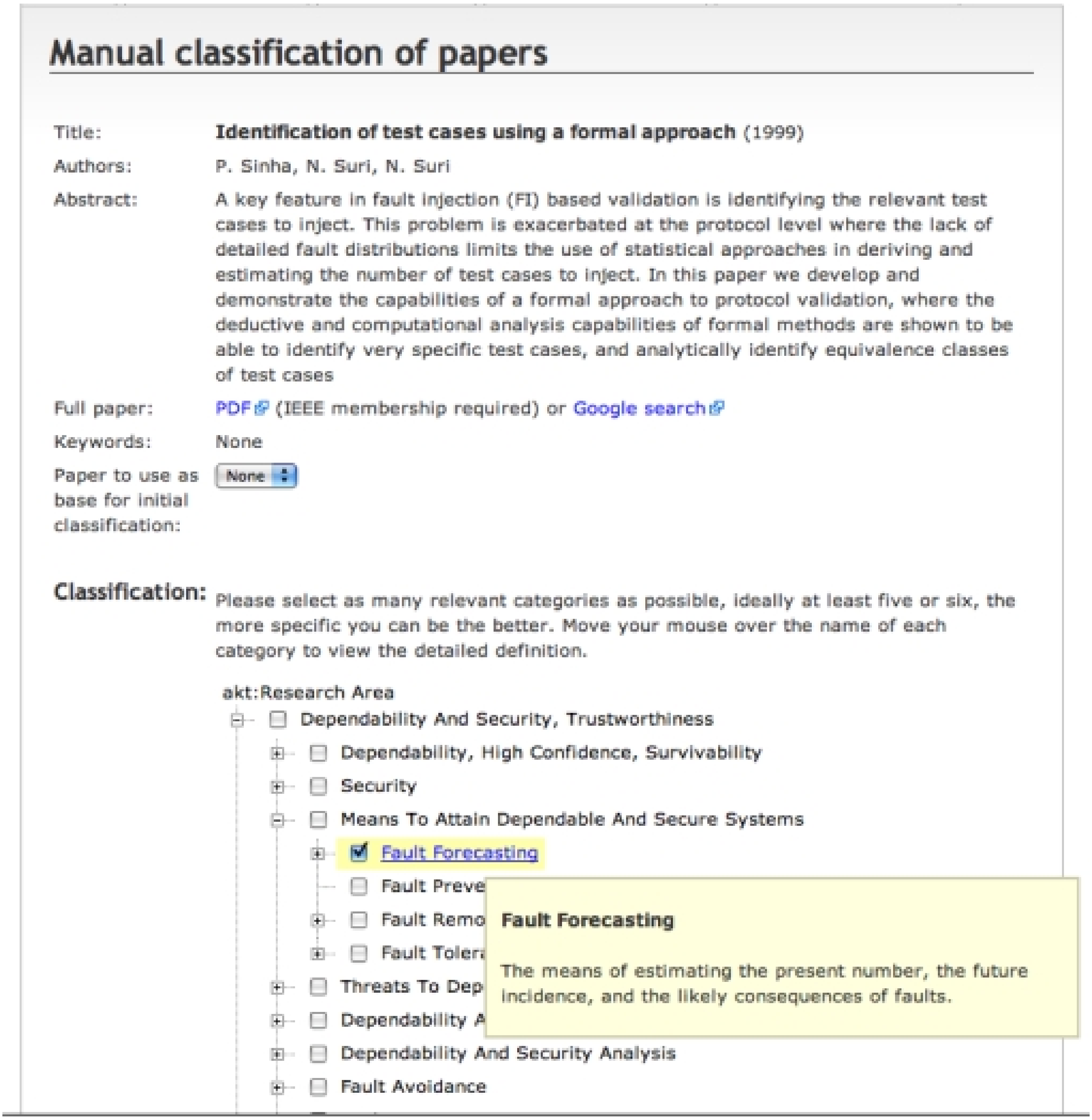
The primary function of the RKB is to provide appropriate content for the project and researchers in resilient systems. D23 provides additional content both acquired from outside sources and generated within the project. It also represents a considerable improvement in the quality of the data over the prototype D10.

2.1 IEEE

The project identified the IEEE FTCS/DSN & Oakland series of conferences as an important specialised resource. The RKB therefore now has the metadata for the series.

This has been used to facilitate the generation of further content by inviting members of the project to establish the relationship between these papers and subject classifications, in particular the resilience and ACM ontologies. The specialised page that was provided for this is shown in Figure 1 and can be found from http://resist.ecs.soton.ac.uk/classifier/manual/; an example of the data gathered is shown in Figure 2.

This activity has led to the training set based on the data collected, which is now being used by IT-T3 to build the taxonomy and classification services.



Manual classification of papers

Title:	**Identification of test cases using a formal approach** (1999)
Authors:	P. Sinha, N. Suri, N. Suri
Abstract:	A key feature in fault injection (FI) based validation is identifying the relevant test cases to inject. This problem is exacerbated at the protocol level where the lack of detailed fault distributions limits the use of statistical approaches in deriving and estimating the number of test cases to inject. In this paper we develop and demonstrate the capabilities of a formal approach to protocol validation, where the deductive and computational analysis capabilities of formal methods are shown to be able to identify very specific test cases, and analytically identify equivalence classes of test cases
Full paper:	PDF (IEEE membership required) or Google search
Keywords:	None
Paper to use as base for initial classification:	None

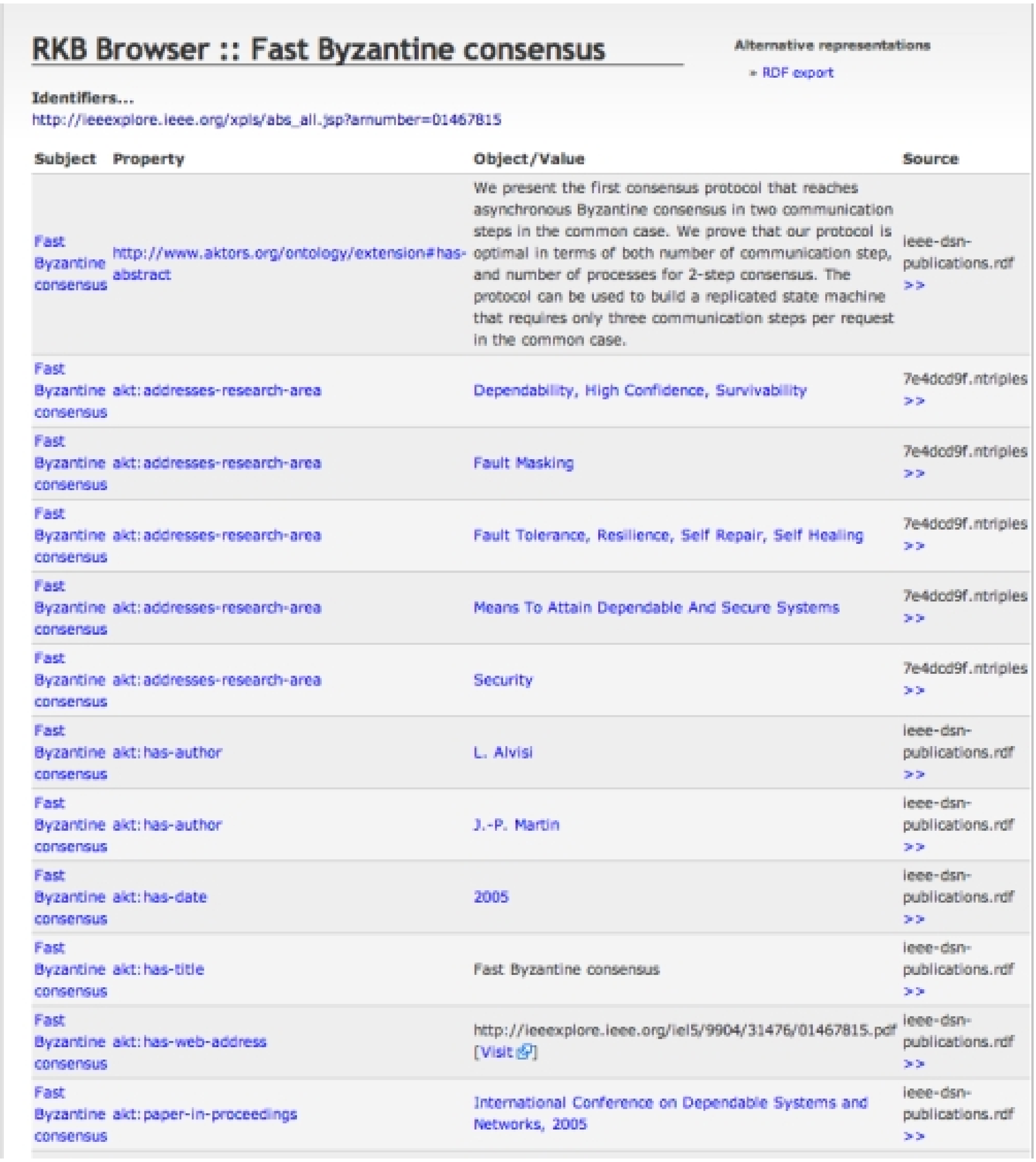
Classification: Please select as many relevant categories as possible, ideally at least five or six, the more specific you can be the better. Move your mouse over the name of each category to view the detailed definition.

akt:Research Area
- □ Dependability And Security, Trustworthiness
  - □ Dependability, High Confidence, Survivability
  - □ Security
  - □ Means To Attain Dependable And Secure Systems
    - ☑ Fault Forecasting
    - □ Fault Preve
    - □ Fault Remo
    - □ Fault Toler
  - □ Threats To Dep
  - □ Dependability A
  - □ Dependability And Security Analysis
  - □ Fault Avoidance

Fault Forecasting

The means of estimating the present number, the future incidence, and the likely consequences of faults.

Figure 1 - Paper Classification Form



RKB Browser :: Fast Byzantine consensus

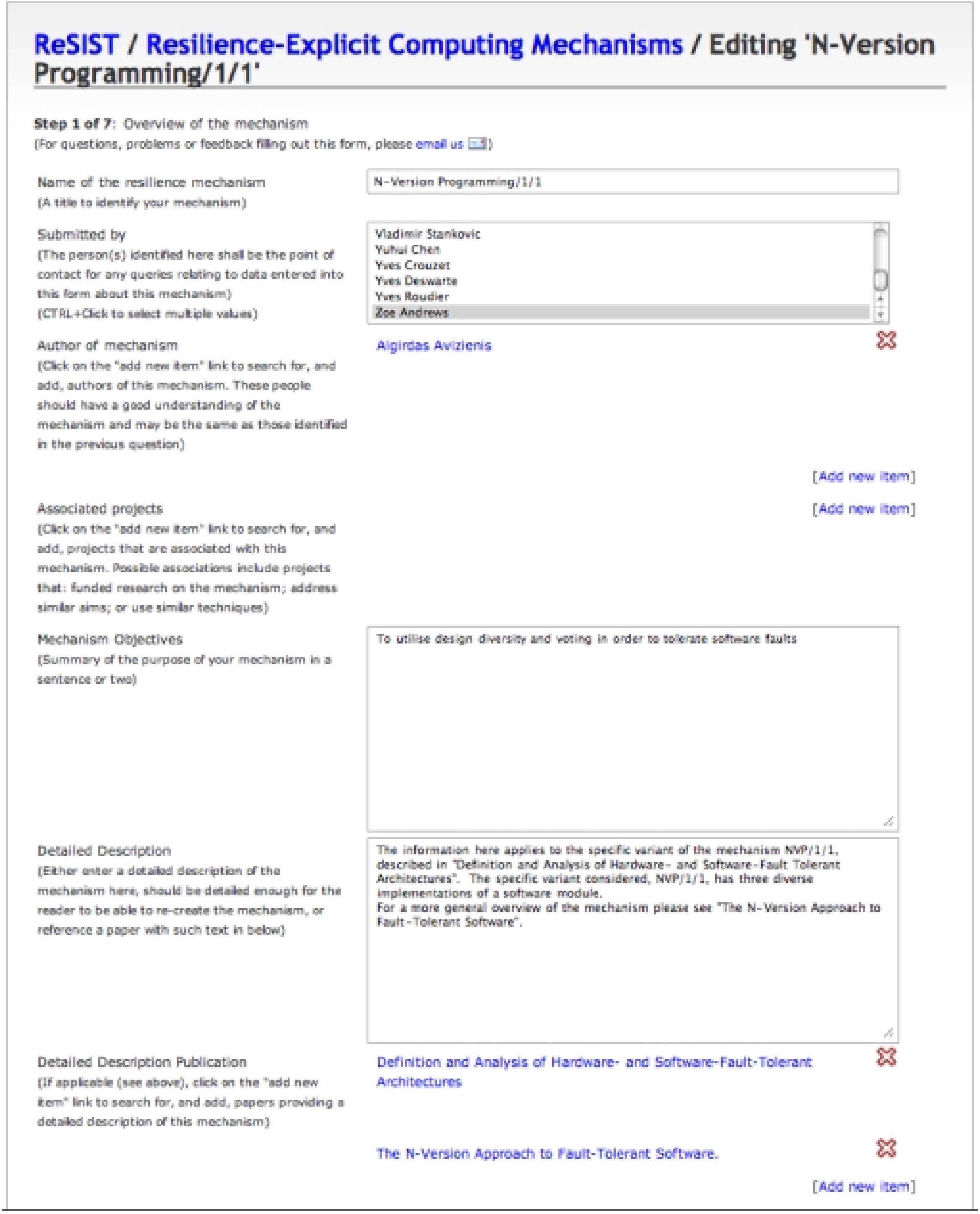
Identifiers...
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=01467815

Subject	Property	Object/Value	Source
Fast Byzantine consensus	http://www.aktors.org/ontology/extension#has-abstract	We present the first consensus protocol that reaches asynchronous Byzantine consensus in two communication steps in the common case. We prove that our protocol is optimal in terms of both number of communication step, and number of processes for 2-step consensus. The protocol can be used to build a replicated state machine that requires only three communication steps per request in the common case.	ieee-dsn-publications.rdf >>
Fast Byzantine consensus	akt:addresses-research-area	Dependability, High Confidence, Survivability	7e4dcd9f.ntriples >>
Fast Byzantine consensus	akt:addresses-research-area	Fault Masking	7e4dcd9f.ntriples >>
Fast Byzantine consensus	akt:addresses-research-area	Fault Tolerance, Resilience, Self Repair, Self Healing	7e4dcd9f.ntriples >>
Fast Byzantine consensus	akt:addresses-research-area	Means To Attain Dependable And Secure Systems	7e4dcd9f.ntriples >>
Fast Byzantine consensus	akt:addresses-research-area	Security	7e4dcd9f.ntriples >>
Fast Byzantine consensus	akt:has-author	L. Alvisi	ieee-dsn-publications.rdf >>
Fast Byzantine consensus	akt:has-author	J.-P. Martin	ieee-dsn-publications.rdf >>
Fast Byzantine consensus	akt:has-date	2005	ieee-dsn-publications.rdf >>
Fast Byzantine consensus	akt:has-title	Fast Byzantine consensus	ieee-dsn-publications.rdf >>
Fast Byzantine consensus	akt:has-web-address	http://ieeexplore.ieee.org/iel5/9904/31476/01467815.pdf [Visit]	ieee-dsn-publications.rdf >>
Fast Byzantine consensus	akt:paper-in-proceedings	International Conference on Dependable Systems and Networks, 2005	ieee-dsn-publications.rdf >>

Figure 2 – IEEE Paper Metadata

2.2 ResEx – Resilience-Explicit Computing

Further content was provided by way of the ResEx activity, and facilities to support this are provided. Apart from the simple provision of data, new metadata types can be added to the RKB, and complex structures can be acquired and represented. Figure 3 shows the form being used, and Figure 4 shows a part of the metadata collected. These can be accessed by following the links from http://resist.ecs.soton.ac.uk/resex/.

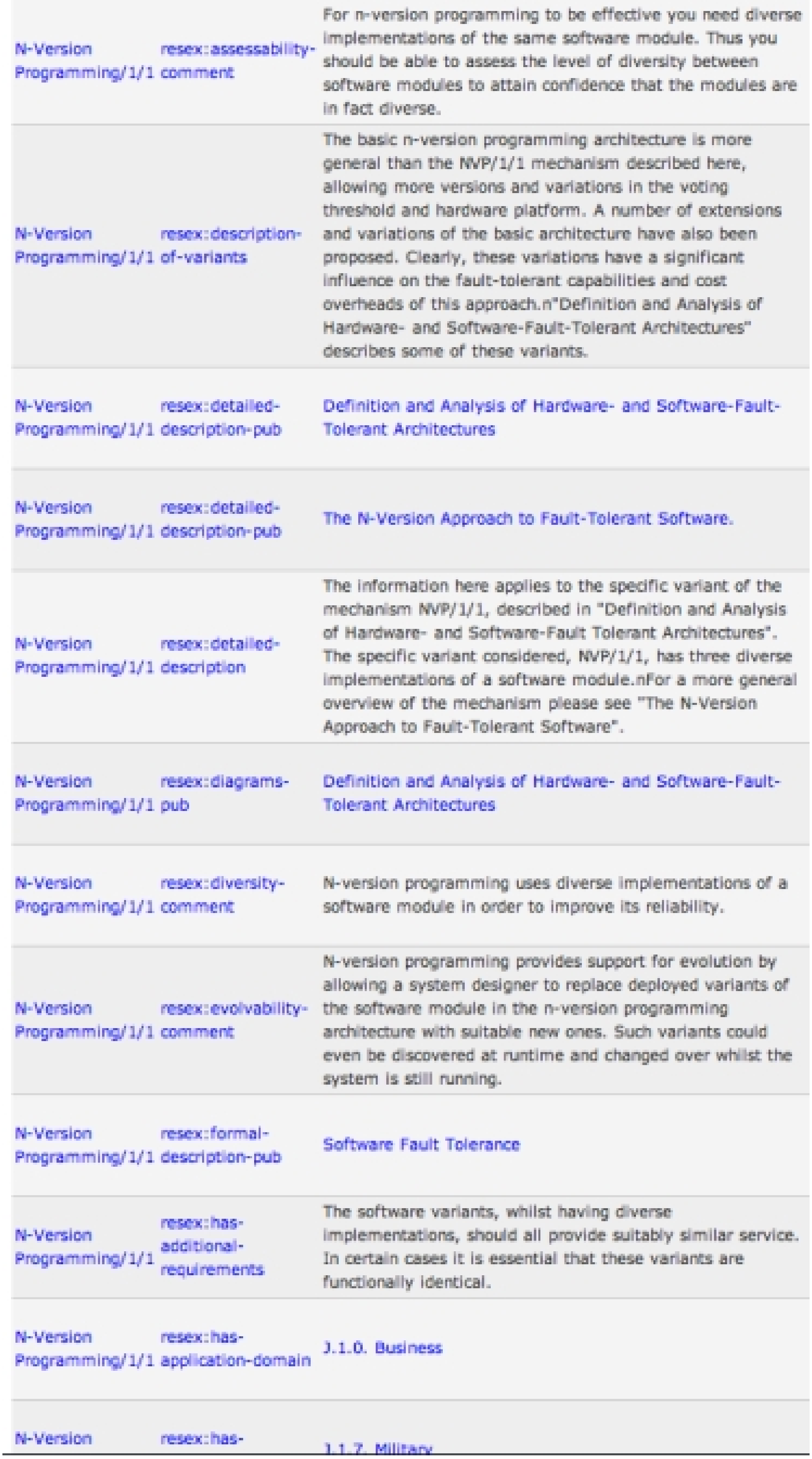


ReSIST / Resilience-Explicit Computing Mechanisms / Editing 'N-Version Programming/1/1'

Step 1 of 7: Overview of the mechanism
(For questions, problems or feedback filling out this form, please email us)

Name of the resilience mechanism (A title to identify your mechanism)	N-Version Programming/1/1
Submitted by (The person(s) identified here shall be the point of contact for any queries relating to data entered into this form about this mechanism) (CTRL+Click to select multiple values)	Vladimir Stankovic Yuhui Chen Yves Crouzet Yves Deswarte Yves Roudier Zoe Andrews
Author of mechanism (Click on the "add new item" link to search for, and add, authors of this mechanism. These people should have a good understanding of the mechanism and may be the same as those identified in the previous question)	Algirdas Avizienis ☒ [Add new item]
Associated projects (Click on the "add new item" link to search for, and add, projects that are associated with this mechanism. Possible associations include projects that: funded research on the mechanism; address similar aims; or use similar techniques)	[Add new item]
Mechanism Objectives (Summary of the purpose of your mechanism in a sentence or two)	To utilise design diversity and voting in order to tolerate software faults
Detailed Description (Either enter a detailed description of the mechanism here, should be detailed enough for the reader to be able to re-create the mechanism, or reference a paper with such text in below)	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".
Detailed Description Publication (If applicable (see above), click on the "add new item" link to search for, and add, papers providing a detailed description of this mechanism)	Definition and Analysis of Hardware- and Software-Fault-Tolerant Architectures ☒ The N-Version Approach to Fault-Tolerant Software. ☒ [Add new item]

Figure 3 – Resilence-Explicit Mechanism Input Form (Page 1)



N-Version Programming/1/1	resex:assessability-comment	For n-version programming to be effective you need diverse implementations of the same software module. Thus you should be able to assess the level of diversity between software modules to attain confidence that the modules are in fact diverse.
N-Version Programming/1/1	resex:description-of-variants	The basic n-version programming architecture is more general than the NVP/1/1 mechanism described here, allowing more versions and variations in the voting threshold and hardware platform. A number of extensions and variations of the basic architecture have also been proposed. Clearly, these variations have a significant influence on the fault-tolerant capabilities and cost overheads of this approach.n"Definition and Analysis of Hardware- and Software-Fault-Tolerant Architectures" describes some of these variants.
N-Version Programming/1/1	resex:detailed-description-pub	Definition and Analysis of Hardware- and Software-Fault-Tolerant Architectures
N-Version Programming/1/1	resex:detailed-description-pub	The N-Version Approach to Fault-Tolerant Software.
N-Version Programming/1/1	resex: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.nFor a more general overview of the mechanism please see "The N-Version Approach to Fault-Tolerant Software".
N-Version Programming/1/1	resex:diagrams-pub	Definition and Analysis of Hardware- and Software-Fault-Tolerant Architectures
N-Version Programming/1/1	resex:diversity-comment	N-version programming uses diverse implementations of a software module in order to improve its reliability.
N-Version Programming/1/1	resex:evolvability-comment	N-version programming provides support for evolution by allowing a system designer to replace deployed variants of the software module in the n-version programming architecture with suitable new ones. Such variants could even be discovered at runtime and changed over whilst the system is still running.
N-Version Programming/1/1	resex:formal-description-pub	Software Fault Tolerance
N-Version Programming/1/1	resex:has-additional-requirements	The software variants, whilst having diverse implementations, should all provide suitably similar service. In certain cases it is essential that these variants are functionally identical.
N-Version Programming/1/1	resex:has-application-domain	1.1.0. Business
N-Version	resex:has-	1.1.7. Military

Figure 4 – Resilience-Explicit Mechanism Metadata Fragment

2.3 The RKB

There has been significant reprocessing of the data that was gathered in the prototype phase, reflecting lessons learnt, as well as changes in the technical context. This reprocessing uncovered some further problems that proved to be more challenging than might have been expected.

The reprocessing falls into three categories, each of which supports the needs of the others, as can be seen in the descriptions later:

1) Building of a separate knowledge base (KB) for each source of the data;
2) "Minting" of new URIs for each concept constructed;
3) Identification and management of multiple URIs that refer to the same concept (the co-reference problem).

Separate KBs

The RKB is now divided into a separate KB for each source. This has a number of advantages, including:

individual KBs no longer needs to be co-located with the other KBs;

the KB can be moved to the data provider's site, or elsewhere;

KB management can be delegated and is easier;

the RDF can be accessed by resolving URIs (see below);

RDF assertion can be more efficient;

querying can be localised, reducing the possibility of future scalability problems.

More detailed access to the RKB data can be found from the main page at http://www.rkbexplorer.com/explorer/. From here access is available to the 25 separate but linked knowledge bases that can be used by the RKB and Explorer.

For each KB, a base page can be found which gives a summary of the KB and lists the facilities available for access. Taking the example of the ieee.rkbexplorer.com KB, they are listed as follows:

1) All the URIs used are resolvable. This, along with the SPARQL query interface (see (3) below), enables the RKB data to be exposed for anyone to use, as well as the Explorer and other ReSIST facilities;

2) It is possible to browse the particular KB, using a raw triple browser;

3) A SPARQL query interface is provided, exposing the RKB in a form that can be used by external applications;

4) The linkage of this KB with other ReSIST KBs and beyond is exposed in the Consistent Reference Service (CRS);

5) The full RDF is also published, along with Sitemap information to allow search engines to index it efficiently.

New URIs

It became apparent during the building of the prototype that few of the data providers had appropriate URIs that could be used for concepts in the knowledge base. Where they did, it was not possible to guarantee that the meaning of the URI would not change, or even if it was correct (see DBLP below). It therefore became necessary to establish our own URI management, so that we could control the meaning of the data we were effectively publishing. We therefore chose to "mint" a new URI for each concept, placed at an appropriate sub-domain referring to a data source KB, as above.

As an example, one of the IEEE papers has a URI of http://ieee.rkbexplorer.com/id/publication-00466948. This is a real URL that can be used to access the metadata we hold about the paper. This metadata includes the fact that IEE have a copy of the paper at http://ieeexplore.ieee.org/iel5/3246/9797/00466948.pdf. In fact, the system supports content negotiation on our URI: it will return *html* or *rdf* descriptions, according to the content type requested by the requesting agent (using a 303 redirect). This conforms to the latest standards in the Linking Open Data initiative.

The use of such resolvable URIs exposes the RKB knowledge in an effective way, avoiding the need to engage in SPARQL queries or process large RDF files, as before. For example the Tabulator RDF browser (http://www.w3.org/2005/ajar/tab) can explore the RKB by starting with a URI that returns RDF directly, such as http://ieee.rkbexplorer.com/data/publication-00466948 .

Identification of multiple URIs

In a KB such as the RKB, when multiple datasets are combined, the URIs from different sources must be processed to determine if they in fact refer to the same real world entities. For D10 we had deployed tools that performed this analysis, and made the results available to the RKB Explorer.

The most important and disturbing lesson learnt from the prototype was that, although these tools were effective, the data they were working on was far worse than expected.

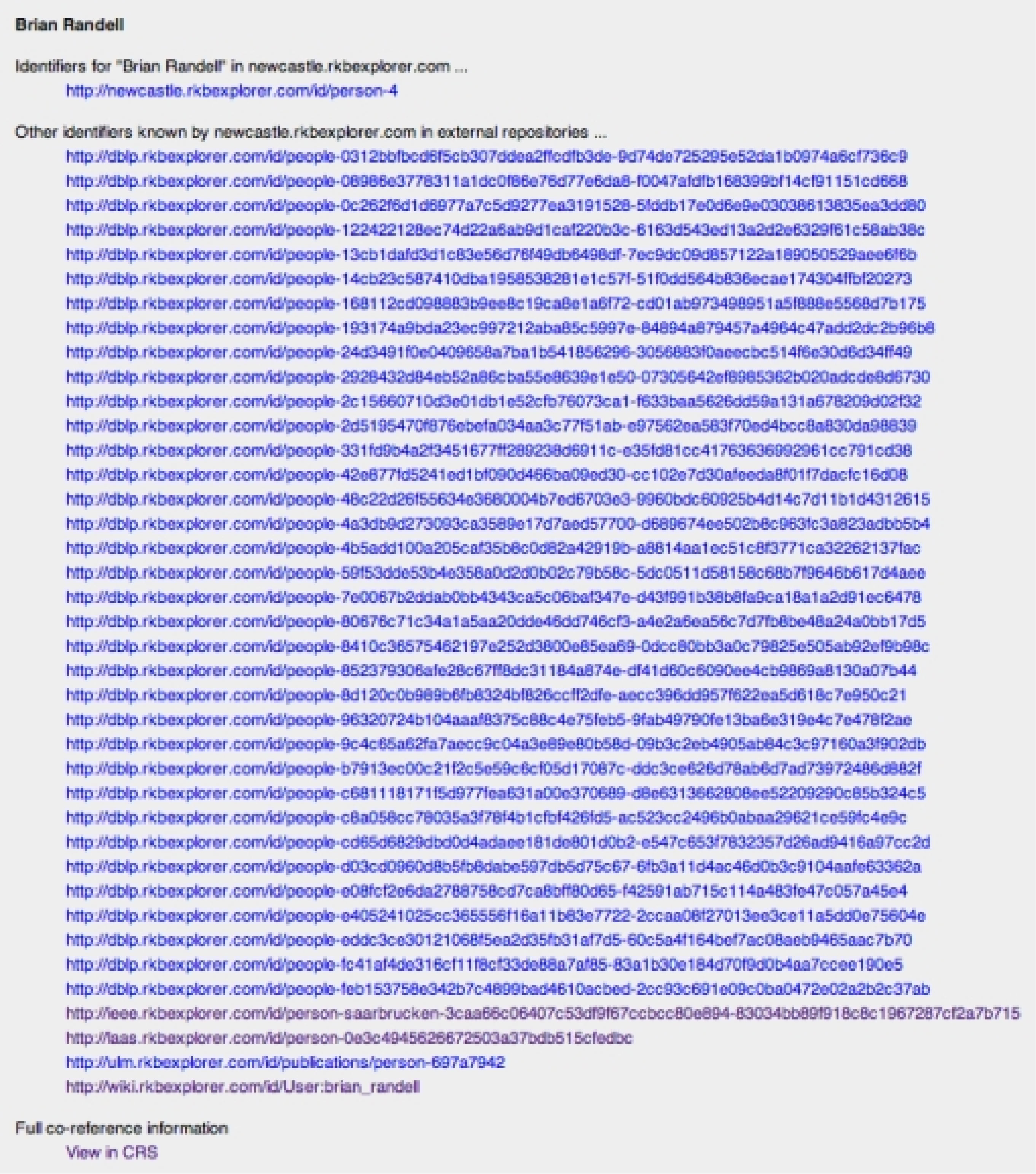
In a publication resource such as DBLP (http://www.informatik.uni-trier.de/~ley/db/), it is widely appreciated that the team has worked hard to ensure that each person is only the subject of one record, and that each person has a record of their own. When a human browses the data, they find this is largely true, and are able to discount the cases where mistakes have been made. Unfortunately, when such data is aggregated with other sources, the network effect which is being created can cause errors to be more significant and pervasive.

For example, when we looking at a particular CORDIS project, the RKBExplorer indicated that there were some NSF projects that were related, without apparent reason. Further investigation showed the reason: "Tom Anderson" was named as investigator on both. Clearly these were different "Tom Anderson"s, but unfortunately DBLP had joined the publications from the two individuals as if they were one (a single URI for both), and so the RKB had accepted the judgement. Further investigation revealed that the problem was so widespread that the only way to use this valuable resource was to ignore such judgements that DBLP was making altogether.

The result was that our system was re-engineered to generate a new URI for each author name on each paper, each paper, and many other concepts, and the RKB took on the task of doing the entire co-reference resolution. Such a "cold start" co-reference activity is in fact different from the incremental system available, and so a new system has been built and is now deployed.

Each KB was augmented with an associated Consistent Reference Service (CRS) to make this co-reference knowledge available to the RKBExplorer and other users.

An example of the newcastle.rkbexplorer.com CRS entry for Brian Randell is given in Figure 5. It can be seen that the URI that has been generated from the Newcastle source has been established as being the same as many URIs representing authors with the string "Brian Randell" (or similar) in the DBLP source. It can also be seen that the IEEE papers have the same author, and both LAAS and Ulm University have provided information including references to him. Finally, the CRS connects all these up with the appropriate person in the project wiki.



Brian Randell

Identifiers for "Brian Randell" in newcastle.rkbexplorer.com ...

http://newcastle.rkbexplorer.com/id/person-4

Other identifiers known by newcastle.rkbexplorer.com in external repositories ...

http://dblp.rkbexplorer.com/id/people-0312bbfbcd6f5cb307ddea2ffcdfb3de-9d74de725295e52da1b0974a6cf736c9
http://dblp.rkbexplorer.com/id/people-08986e3778311a1dc0f86e76d77e6da8-f0047afdfb168399bf14cf91151cd668
http://dblp.rkbexplorer.com/id/people-0c262f6d1d6977a7c5d9277ea3191528-5fddb17e0d6e9e03038613835ea3dd80
http://dblp.rkbexplorer.com/id/people-122422128ec74d22a6ab9d1caf220b3c-6163d543ed13a2d2e6329f61c58ab38c
http://dblp.rkbexplorer.com/id/people-13cb1dafd3d1c83e56d76f49db6498df-7ec9dc09d857122a189050529aee6f6b
http://dblp.rkbexplorer.com/id/people-14cb23c587410dba1958538281e1c57f-51f0dd564b836ecae174304ffbf20273
http://dblp.rkbexplorer.com/id/people-168112cd098883b9ee8c19ca8e1a6f72-cd01ab973498951a5f888e5568d7b175
http://dblp.rkbexplorer.com/id/people-193174a9bda23ec997212aba85c5997e-84894a879457a4964c47add2dc2b96b8
http://dblp.rkbexplorer.com/id/people-24d3491f0e0409658a7ba1b541856296-3056883f0aeecbc514f6e30d6d34ff49
http://dblp.rkbexplorer.com/id/people-2928432d84eb52a86cba55e8639e1e50-07305642ef8985362b020adcde8d6730
http://dblp.rkbexplorer.com/id/people-2c15660710d3e01db1e52cfb76073ca1-f633baa5626dd59a131a678209d02f32
http://dblp.rkbexplorer.com/id/people-2d5195470f876ebefa034aa3c77f51ab-e97562ea583f70ed4bcc8a830da98839
http://dblp.rkbexplorer.com/id/people-331fd9b4a2f3451677ff289238d6911c-e35fd81cc4176363699296 1cc791cd38
http://dblp.rkbexplorer.com/id/people-42e877fd5241ed1bf090d466ba09ed30-cc102e7d30afeeda8f01f7dacfc16d08
http://dblp.rkbexplorer.com/id/people-48c22d26f55634e3680004b7ed6703e3-9960bdc60925b4d14c7d11b1d4312615
http://dblp.rkbexplorer.com/id/people-4a3db9d273093ca3589e17d7aed57700-d689674ee502b8c963fc3a823adbb5b4
http://dblp.rkbexplorer.com/id/people-4b5add100a205caf35b8c0d82a42919b-a8814aa1ec51c8f3771ca32262137fac
http://dblp.rkbexplorer.com/id/people-59f53dde53b4e358a0d2d0b02c79b58c-5dc0511d58158c68b7f9646b617d4aee
http://dblp.rkbexplorer.com/id/people-7e0067b2ddab0bb4343ca5c06baf347e-d43f991b38b8fa9ca18a1a2d91ec6478
http://dblp.rkbexplorer.com/id/people-80676c71c34a1a5aa20dde46dd746cf3-a4e2a6ea56c7d7fb8be48a24a0bb17d5
http://dblp.rkbexplorer.com/id/people-8410c38575462197e252d3800e85ea69-0dcc80bb3a0c79825e505ab92ef9b98c
http://dblp.rkbexplorer.com/id/people-852379306afe28c67ff8dc31184a874e-df41d80c6090ee4cb9869a8130a07b44
http://dblp.rkbexplorer.com/id/people-8d120c0b989b6fb8324bf826ccff2dfe-aecc396dd957f622ea5d618c7e950c21
http://dblp.rkbexplorer.com/id/people-96320724b104aaaf8375c88c4e75feb5-9fab49790fe13ba6e319e4c7e478f2ae
http://dblp.rkbexplorer.com/id/people-9c4c65a62fa7aecc9c04a3e89e80b58d-09b3c2eb4905ab84c3c97160a3f902db
http://dblp.rkbexplorer.com/id/people-b7913ec00c21f2c5e59c6cf05d17087c-ddc3ce626d78ab6d7ad73972486d882f
http://dblp.rkbexplorer.com/id/people-c681118171f5d977fea631a00e370689-d8e6313662808ee52209290c85b324c5
http://dblp.rkbexplorer.com/id/people-c8a058cc78035a3f78f4b1cfbf426fd5-ac523cc2496b0abaa29621ce59fc4e9c
http://dblp.rkbexplorer.com/id/people-cd65d6829dbd0d4adaee181de801d0b2-e547c653f7832357d26ad9416a97cc2d
http://dblp.rkbexplorer.com/id/people-d03cd0960d8b5fb8dabe597db5d75c67-6fb3a11d4ac46d0b3c9104aafe63362a
http://dblp.rkbexplorer.com/id/people-e08fcf2e6da2788758cd7ca8bff80d65-f42591ab715c114a483fe47c057a45e4
http://dblp.rkbexplorer.com/id/people-e405241025cc365556f16a11b83e7722-2ccaa08f27013ee3ce11a5dd0e75604e
http://dblp.rkbexplorer.com/id/people-eddc3ce30121068f5ea2d35fb31af7d5-60c5a4f164bef7ac08aeb9465aac7b70
http://dblp.rkbexplorer.com/id/people-fc41af4de316cf11f8cf33de88a7af85-83a1b30e184d70f9d0b4aa7ccee190e5
http://dblp.rkbexplorer.com/id/people-feb153758e342b7c4899bad4610acbed-2cc93c691e09c0ba0472e02a2b2c37ab
http://ieee.rkbexplorer.com/id/person-saarbrucken-3caa66c06407c53df9f67ccbcc80e894-83034bb89f918c8c1967287cf2a7b715
http://laas.rkbexplorer.com/id/person-0e3c4945626672503a37bdb515cfedbc
http://ulm.rkbexplorer.com/id/publications/person-697a7942
http://wiki.rkbexplorer.com/id/User:brian_randell

Full co-reference information

View in CRS

Figure 5 – CRS entry in newcastle.rkbexplorer.com for Brian Randell

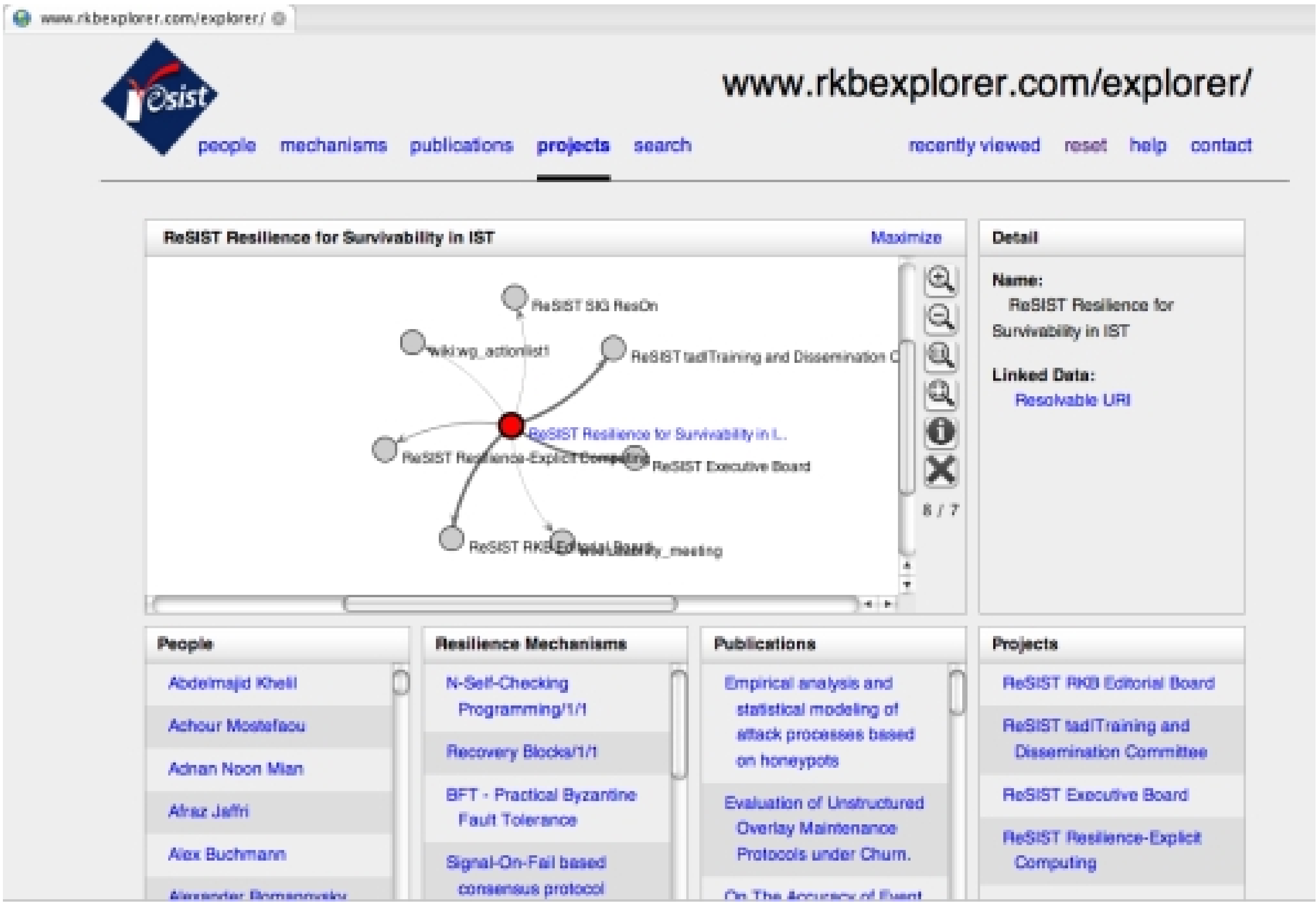
3 - RKBExplorer

A revised version of the exploration user interface to the RKB, running on the newly-commissioned hardware, can be found at http://www.rkbexplorer.com/explorer/; details of this interface, in the form of a User Guide, were given in D10, and can be found at the “Help” link from that page. It was revised to conform to the new project visual identity, and minor changes were made in response to user comments.

The main change that can be seen from a user’s point of view is that the data displayed in the interface has become more specialised towards the project subject domain, by the inclusion of ResEx Mechanisms.

The main change that is not clear to see is that it has been re-engineered to present a unified view of the many KBs that make up the RKB (rather then the single RKB previously). This means that it now also has the potential to include KBs that are from other sites and other projects.

Figure 6 shows the first screen of the latest version of RKBExplorer.


www.rkbexplorer.com/explorer/
esist
www.rkbexplorer.com/explorer/
people
mechanisms
publications
projects
search
recently viewed
reset
help
contact
Maximize
ReSIST SIG ResOn
wiki:wg_actionlist1
ReSIST tadTraining and Dissemination C
ReSIST Resilience for Survivability in I...
ReSIST Resilience-Explicit Computing
ReSIST Executive Board
ReSIST RKB Editorial Board
wiki:usability_meeting
8 / 7

ReSIST Resilience for Survivability in IST

Detail

Name:
ReSIST Resilience for Survivability in IST

Linked Data:
Resolvable URI

People
- Abdelmajid Khelil
- Achour Mostefaou
- Adnan Noor Mian
- Afraz Jaffri
- Alex Buchmann
- Alexander Romanovsky

Resilience Mechanisms
- N-Self-Checking Programming/1/1
- Recovery Blocks/1/1
- BFT - Practical Byzantine Fault Tolerance
- Signal-On-Fail based consensus protocol

Publications
- Empirical analysis and statistical modeling of attack processes based on honeypots
- Evaluation of Unstructured Overlay Maintenance Protocols under Churn.
- On The Accuracy of Event

Projects
- ReSIST RKB Editorial Board
- ReSIST tadTraining and Dissemination Committee
- ReSIST Executive Board
- ReSIST Resilience-Explicit Computing

Figure 6 – RKBExplorer

4 – Conclusions

We have presented here the status of the RKB and associated mechanisms at the end of year two. The content has been maintained and expanded, in general terms and specifically, according to the needs of the project from time to time. Considerable work has been completed to improve the infrastructure so that those past and the future requirements can be met efficiently.

Future work will continue along the same lines so that the quality and coverage of the knowledge is improved, as well as the inferences that can be drawn from that knowledge.