Return to the Theorem Prover's House: Application of the Learning Grid to Formal Methods

Juan Bicarregui, Damian Mac Randal, Brian Matthews, Brian Ritchie
CCLRC Rutherford Appleton Laboratory

1. INTRODUCTION

Some years ago, in a light-hearted appendix to the book "*mural*: A Formal Development Support System" [1], Alan Wills gave a speculative vision of a future "Theorem Prover's House". The piece is a fantasy imagining the typical day in the life of a professional theorem prover of the future at a time when theorem proving becomes a normal activity in software engineering, with a high demand for contract theorem proving supplied by highly-paid specialist engineers. Furthermore, the theorem prover is assisted in his work by a series of advanced tools providing access to network-accessible theory libraries and proof tools, powerful user-interfaces and other tools embedded into the fabric of the house.

We give an extract to give a flavour of the piece, which is given in full in an appendix.

> *"Collapsing the Business Admin browser back into its cartouche, he chooses instead the Theorem Proving role. The summary of the Incoming Jobs Channel shows the originators, subject matter and fees of a couple of small jobs and one big one: "STL, Group Theory, 4000kW; Jones, prog transformation, 3500kW; Praxis, spec lang consistency, 32000kW". He fingers the last, getting the details showing deadline, originator's informal description, originator id and reference, agent's id and reference, expenses incurred (initialised to 0), sundry other administrative and commercial details, and of course the formal description of the problem itself. This he expands to a long scroll over the desk, then splits off some interesting bits into separate windows. He spends an hour or so just reading it, crawling over it on his hands and knees, often juxtaposing various parts, or revealing elided detail (and sometimes getting irritated when a window gets 'stuck' to his knee as he moves around).*

Twelve years on, we return to the Theorem Prover's House on its cliff-top above the sea to see how a contemporary reading of the narrative reveals how far technology has progressed towards realising this vision and to discuss to what extent the envisaged features are available or expected to become available today. From this perspective, we go on to discuss how formal methods might be both carried out in practise and taught to students in the future.

2. FEATURES OF THE THEOREM PROVER'S HOUSE

A close reading of the appendix reveals that the following features were envisaged as being available to the Theorem Prover:

- shared workspaces for developing specifications and theorems;
- access to searchable large-scale on-line theory libraries;
- access to formal reasoning and theorem provers systems remotely;
- shared visualisation on a large scale with haptic interaction devices over wall-sized displays;
- rapid prototyping via access to remote machines;
- role based access control and work-flow;
- negotiation and specialist service agents

Whilst these features have yet to be put together in the combination envisaged in the appendix, all the above features are now within the realms of technical feasibility. Indeed, they are being brought together in the work supporting large-scale interaction in the various Grid initiatives.

2.1 Processes and Agents

> *…a considerable problem for himself and many others, when a date-dependent bug appeared in the accountant last April. Since then, he has changed to a formally-certified accountant; the old one was later convicted of unverified behaviour , and descheduled.*

The "accountant" mentioned in the scenario is clearly an example of a software agent, an autonomous, context aware system handling tax matters on behalf of the Theorem Prover. Agent technology is rapidly becoming a mainstream technology, with OMG's MASIF standard for mobile agents [2], and FIPA (Foundation for Physical Intelligent Agent) focusing on Intelligent Agents [3]. There are many Software Agent Platforms available, supporting the development of agents. Recently there have been attempts to exploit the overlaps between Agent and Grid technologies to provide a unified, heterogeneous, distributed computing environment for agents [4].

The major difficulty, as for most current agent systems, is ensuring the environment is capable of providing the requisite information for the agent to act upon. In closed domains such as tax or accounting, knowledge and information repositories could be constructed to provide such an environment (see Pellucid [5] for an example in another context). Pellucid also exemplifies another way of using agents; rather than the agent doing (external) things on behalf of the user, the agent could be providing help, support and advice to the user.

The notion of cooperation leads on to another technology hinted at but not explicitly mentioned in the scenario; the use of workflow or business process management systems to coordinate agents, systems or people who need to collaborate to achieve a shared goal. The Theorem Prover seems to be a peculiarly solitary worker; in practice most jobs are part of a larger process and involve understanding of the bigger picture and collaboration with others working on related parts of the overall problem.

The Theorem Prover would probably identify most strongly with formally defined processes, where he could be sure he has all the necessary inputs, that all necessary interactions have been defined and that his results will be properly applied by others. Work on formal business processes specification languages and infrastructures has not progressed much since the early HICOS system [6] proposed formal mapping of business requirements through process descriptions to executable processes and supported reasoning about process properties. Recent work in the Web Services arena is reintroducing the notion of formal specifications of processes, e.g. BPMI [7], but reasoning about processes is still in its infancy.

At the other end of the business process spectrum are the flexible dynamic processes typical of management, design or customer-facing tasks. In tackling the Praxis job, the Theorem Prover is presumably following one of the templates or methodologies he has built up through experience (but adapting it on-the-fly to suit the details of the job in hand). However, there is no evidence in the scenario of project management tools, let alone a process support tool for his work. For larger or more complex jobs, especially those involving other people (or agents?), tools to support sub-task allocation, inter-task dependencies and recovery from errors or failed proofs would become essential (see Proflex [8] for an example in another domain). The move towards distributed, independent Web/Grid services will exacerbate the

problem, but initiatives such as BPEL [9] and BPMI (both Web service choreography languages) are already tackling this

One outstanding area of concern in both agent and process support systems is trust, security and privacy, particularly where multiple agents representing different entities are cooperating. (It is interesting that the Theorem Prover had not insisted on a certified accountant until after he got burned!). After years of being treated as "just a security problem", recently several projects, e.g. TrustCom[10] have started to develop Trust management frameworks in which Services can safely cooperate as dynamic virtual organizations created for the purpose of solving one problem and then disbanding.

2.2 Semantic Interchange

> *The NCC library has now interrogated the TP's personal environment to find out where he likes to start navigating from, and the Library window shows this position. He navigates through, spawning off a few useful theorems and tactics in their own windows and comparing them with parts of the job's formal statement*

One feature of the Theorem Prover's house which is almost taken for granted is that the terminology and sense of the terms that the Theorem Prover is using is shared seamlessly between the Theorem Prover and his clients, and also with external services such as the large-scale on-line theory libraries, which the Theorem Prover can set his agents to search and integrate with his tools. This in general is not trivial and requires a shared knowledge representation describing resources; services can publish their definitions and properties of shared terms and exchange and merge these descriptions so that the end user may not see the interaction. The technologies underlying this interchange were well advanced at the time of the original Theorem Prover's House, but their large scale deployment was not foreseen. Now with the emergence of the Semantic Web, knowledge representation formalisms to describe resources (e.g. RDF [11]) and Ontologies to describe their properties (e.g. OWL [12]) have become common currency. There is now the real potentiality that semantically meaningful terms can be shared across the network; the Theorem Prover should now be able to get "*the details showing deadline, originator's informal description, originator id and reference, agent's id and reference*" of jobs in a common framework from anywhere in the world on his incoming Jobs Channel. Similarly, the terms used to search the *NCC library, and to establish* his user preferences can be shared much more easily. The beginning of this process of using the Semantic Web has been started in [13,14]. Incidentally, the emphasis placed in the Semantic Web on the use of logic and proof as building blocks to establishing the properties of web resources and providing a surety on their trustworthiness may lead to a wider appreciation of the value of formal definition and proof.

2.3 Web and Grid Services

> *He drags one over towards the Library's socket and as soon as they are brought together, they recognize their type-compatibility and the connection is made with a flash and a beep:*

Clearly, the emergence of Web Services [15] and the Grid [16] can be seen as a step towards many of the remote-service and distributed-processing aspects of the Theorem Prover's working environment, most notably the ability to browse available proof services according to a range of selection criteria, to choose several proof services, and to interact with them. In the case of the TP using the NCC library of theorems and tactics, the interaction is relatively fine-grained; whereas the interaction with the chosen automatic prover is more akin to a batch-process model (proof by tea-break – with a slice of lemma?) Let us not forget that the Theorem Prover himself is acting as a service, and may appear as such to his potential clients;

perhaps those "incoming jobs" were directed at him by clients, or their agents, in response to his own advertised service. We can envisage the Theorem Prover registering a UDDI description [17] of his service with one or more business repositories. Thus perhaps our Theorem Prover is more akin to a "proofbroker".

Grid technology in particular opens up the possibility of applying massively parallel computing power to proof strategies; perhaps proof by brute-force will become the dominant form. However, for this to become genuinely feasible, a number of hurdles remain.

Firstly, in order to be adopted, proof services will have to agree on a common representation format for propositions, and almost certainly for proofs; or at least use one of a small set of commonly-agreed formats. If the Theorem Prover wishes to use multiple proof services that use different representations, then he will have to justify transformations between them. (Of course, such transformations could be provided by another form of service; indeed, some transformations will probably require proof themselves.) Though significant, this is a surmountable problem, given sufficient will to agree on formats and transformations. Work in this area includes [18,19].

Another issue that is perhaps more difficult to solve is that of trust and integrity. How can the Theorem Prover be certain (or, second-best, be confident) that a particular proof service really can deliver the goods? What mechanisms have to be in place to prevent the appearance of shady "verify-by-night" proof shops that take our TP's money and return a random truth value? For many other kinds of services, it may be the case that an eBay-style "customer rating" scheme is adequate (given an initial head of steam); but this would hardly be sufficient for a service claiming to provide formally verified proofs. If all proof services were required to supply an LCF-style validation or similar form of independently verifiable proof, then this would (probably) be sufficient; but the onus is on the user of the service to perform the validation in order to verify the result, and this itself might be a time-consuming task. This would be even worse if this verification has to be performed by each of a hierarchy of services and users; if you don't trust your (proof) client, would you want to pass that lack of trust on to your customers? Might our Theorem Prover be undercut by a competitor who forgoes this step, and thus has a higher proofs-per-hour metric, albeit at the risk of reduced accuracy? In the cut-and-trust world of proof obligation generation, it's pog-eat-pog, and if proof-time is money (as it almost certainly is), a service that is fast because it avoids the overhead of generating a formally-reproducible proof may win out. History suggests that we are not likely to see legislation requiring every proof service to provide a verification; at least not until some spectacular public failure (perhaps that date-dependent bug in the accountant).

The functional description of a web service, as, say, a function from proposition to proof-or-failure, says little about how the service is actually implemented: is a particular proof service fully automated, semi-automated, mechanically assisted, or entirely hand-driven? Presumably these characteristics, together with performance claims and charging models, could be part of the proof service metadata. However, who or what will verify these claims? The web of proof may well be built upon the web of service, but it will have to be underwritten by the web of trust.

2.4 Large-scale displays

> *This he expands to a long scroll over the desk, then splits off some interesting bits into separate windows. He spends an hour or so just reading it, crawling over it on his hands and knees, often juxtaposing various parts, or revealing elided detail (and sometimes getting irritated when a window gets 'stuck' to his knee as he moves around).*

The most noticeable aspect of the Theorem Prover's House is that it has a sophisticated user interface with large-scale touch-screen displays forming the surface of the desk, the walls, even the floor of the Theorem Prover's work environment; this is reminiscent of the touch screen glass panels in the recent science fiction film "*Minority Report*". This is an area which has perhaps developed more slowly than some of the others in the report. Desk size displays are possible, but still very much in the research realm; whilst flat-screen technology has advanced, large size Plasma screen technology is still prohibitively expensive. Nevertheless, there are other approaches which open options which are not discussed here, such as voice recognition, which has now become usable, and Virtual Reality offers new opportunities for interaction.

Furthermore, little discussed in the original piece are the opportunities for collaborative working and visualisation, which are now becoming integrated with the work on developing the Grid. These range from relatively cheap multi-user conferencing systems such as Access Grid [20]; the Theorem Prover would almost certainly consult with his clients, or attend meetings of the Royal Logical Society via such a system; to visualisations which integrate a variety of tools and databases across the Grid in a shared visualisation. Thus the Theorem Prover may generate an animation from the specification and its properties derived in the proof and allow customers remotely to integrate the animation into the visualisations of their business and then change the parameters and visualising the results, allowing a shared collaborative software engineering environment. The gViz project amongst others is working towards realising this vision within the Grid [21].

2.5 Wireless technology

> *Ursula Martin envisages the even more remote cellphone model, in which the TP works whilst jogging along the clifftops.*

One obvious area where the technology has become established in a fashion which was just not predictable at the time of writing is in the pervasiveness of wireless and mobile technology. Wireless interaction is mentioned only in the more fanciful "variations" section. These ideas in any modern update to the Theorem Prover's House would be considered the least of the capabilities! It would now be considered almost natural to monitor progress and control the progress of the tools of the House from any location in range of a radio mast or Bluetooth connection.

3. LEARNING GRID OF EXCELLENCE

Thus we can see that much of the technology to realise the Theorem Prover's House is now available or is near to practical use. The question is raised how we might put such environments together and how they might affect the learning and teaching opportunities. Whilst remote learning is now becoming a commonplace (albeit one which is often not done well), the implications on learning of the Grid has hardly even been contemplated. One initiative which is exploring how the Grid might be used to support education is the European Commission supported Learning Grid of Excellence Working Group [22].

The basic objective of LeGE-WG is to facilitate the establishment of a European Learning Grid Infrastructure supporting the systematic exchange of information and creating opportunities for collaboration between the different actors. The WG brings together actors with complementary interests in Grid computing and e-Learning from technological disciplines, pedagogy, government regulating bodies and of course students, so as to achieve in-depth understanding of fundamental issues underpinning the application of Grid computing for e-

Learning. A common theme emerging from the WG, and forming the basis of a FP6 project eLeGI, is "experiential learning", i.e. learning by interacting with (models of) the subject matter. In many fields, these models are computationally expensive, and if provided with experiential interfaces (such as VR), necessitate Grid technologies to collect, organize and deliver the "experience". Theorem proving could be one such domain, and it can be envisaged how Learning Grids, visualisation techniques and knowledge-based tutoring systems could transform theorem proving education from abstract study towards a more engineering "direct manipulation" approach. While aspects of this have been explored in various projects, the emergence of the Learning Grid infrastructure starts to make it practical. Furthermore, given the computational resources and distributed nature of a Learning Grid infrastructure, Continuing Professional Development could also become much more interactive experience, with a background tutoring system suggesting new avenues to learn about, and quietly monitoring experts at work to build up "best practice" methodologies that can be passed on to others. Learning, and teaching, would therefore become part of the fabric of the Theorem Prover's House.

4. CONCLUSIONS

Perhaps the interesting thing about the Theorem Prover's House is not only that only twelve years after publication the vision of the future now looks feasible, but also the areas which would now be seen as feasible which are not discussed. The advances in networking and connectivity which have been seen in the last twelve years have made a layer of interaction and collaborative working between people, and between people and software agents much more natural. The Theorem Prover may live alone in his house, but now is unlikely to be working alone. Also, the advances in Wireless technology have meant that the Theorem Prover will be able to work anywhere and any time.

The possibilities opened up by the Grid have yet to be fully considered for Formal Methods. There are projects which are concerned with applying formal methods to the Grid, particularly in the distribution of verified code [23], and also attempt to solve individual theorems or mathematical problems, such as the Great Internet Mersenne Primes Search [24] using Grid like techniques, but little attempt has been made to link the power of the Grid to provide access to high power theorem proving systems, certified searchable libraries, high-power visualisations and animation, collaborative software design spaces or other applications which could be delivered over the Grid. Now is the time to make the Theorem Prover's House a reality.

REFERENCES

[1]. C.B. Jones, K.D. Jones, P. A. Lindsay and R Moore (1991). *mural: A Formal Development Support System,* Springer-Verlag ISBN 3-540-19651. Appendix E on the Theorem Prover's House is attributed to A.C. Wills.

[2]. *MASIF* http://www.omg.org/technology/documents/formal/mobile_agent_facility.htm

[3]. *FIPA* http://www.fipa.org/

[4]. M. Bradshaw, N. Suri, A. J. Cañas, R. David, K. Ford, R. Hoffman, R. Jeffers, and T. Reichherzer. *Terraforming cyberspace*. Computer, 34(7):48–56, July 2001.

[5]. Pellucid http://www.sadiel.es/Europa/pellucid/default.htm

[6]. Kenneth Robinson, Damian Mac Randal *Business Case Processing - Rationale, Survey, and Trends* SOFSEM'96: Theory and Practice of Informatics 1175 p.143 - 160 K G Jeffery, J Král, M Bartosek (Eds), Lecture Notes in Computer Science, Springer, Berlin, (November 1996)

[7]. J. Koehler, G. Tirenni, S. Kumaran: *From Business Process Model to Consistent Implementation: A Case for Formal Verification Methods*, EDOC 2002, pp. 96-106.
[8]. R. Nachtsheim, D. Lippert *Proflex - Professional Management of Flexible, Customer Driven, Responsive Processes* EMMSEC'99 http://www.cms.livjm.ac.uk/library/EMMSEC/Part-03/054-Nachtsheim.pdf
[9]. *BEPL* http://www-106.ibm.com/developerworks/library/ws-bpel/
[10]. *TrustCom* http://www.bitd.clrc.ac.uk/Activity/ACTIVITY=TRUSTCOM
[11]. *Resource Description Framework (RDF)* http://www.w3.org/RDF
[12]. *The Web Ontology Language OWL* http://www.w3.org/WebOnt
[13]. Jin Song Dong, Jing Sun, Hai Wang: *Semantic Web for Extending and Linking Formalisms*. FME 2002: 587-606
[14]. Jin Song Dong, Jing Sun, Hai Wang: *Z Approach to Semantic Web*. ICFEM 2002: 156-167
[15]. *W3C Web Services Activity*, http://www.w3c.org/2002/ws/
[16]. Ian Foster (Editor), Carl Kesselman (Editor), *The Grid: Blueprint for a New Computing Infrastructure*, 2nd Edition, Morgan Kaufmann, 2003.
[17]. Oasis: *Universal Description, Discovery and Integration of Web Services*, http://www.uddi.org/
[18]. Brian M. Matthews *vdmML: using XML to represent VDM on the Web. Some initial thoughts 2nd Workshop on VDM* York,, (September 2000)
[19]. Jing Sun, Jin Song Dong, Jing Liu and Hai Wang. *A Formal Object Approach to the Design of ZML.* Annals of Software Engineering: An international journal, 13:329-356, Kluwer Academic Publishers, 2002
[20]. *Access Grid* http://www.accessgrid.org
[21]. *gViz* http://www.visualization.leeds.ac.uk/gViz/
[22]. Learning Grid of Excellence http://www.lege-wg.org/
[23]. The ConCert Project: Certified Code for Grid Computing http://www-2.cs.cmu.edu/~concert/
[24]. *Great Internet Mersenne Prime Search*, http://www.mersenne.org/prime.htm

APPENDIX A: THE FULL TEXT OF THE THEOREM PROVER'S HOUSE

"The Theorem Prover's House[1] is on a high headland, above the seagulls' nests in steep cliffs above a pebbly seashore. The Theorem Prover wakes soon after the sun has appeared over the mountains behind the few miles of pasture inland, and sits up in his bed. He stares out to sea, of which he takes pleasure in observing the variations from one day to the next. To the north, only fields and mountains touch the coastline. Some distance down the coast to the south, he can see the little town where he goes, every Thursday, to pick up groceries and to spend an hour or two with the vet's wife while the vet is out on his rounds.

"He descends from the transparent dome enclosing his bed, to the central space of his house. He makes a mug of tea in the peripheral alcove he calls his kitchen; thinks about shaving, but does not because it is not Monday – the vet's wife, somewhat passé, likes a three-day stubble – not Monday, he reasons (in his head), because it is Tuesday, as he determines from the calendar on the wall in his workspace under the bed.

"The calendar is functional but unaesthetic, so he touches it, slides his finger down the resulting menu to the 'today's picture' slot, and stands back to admire the result. It is Turner's The Fighting Temeraire. He has been pleased with his subscription to this service, even it if is a shade pricey; anyway, he has few worries about money. He drags one corner with his finger, until the picture fills much of the wall; but he takes a cursory glance at the noticeboard before covering it: one or two new

results advertised in the category theory section, and some ongoing political wranglings in the board of the Royal Logical Society.

"He has had some ideas overnight for the problem, bits of whose proof are still littered all over the opposite wall; but before pulling it down to his desk, at which he now kneels he will see what has arrived to be done today. Once yesterday's clutter has been moved to the wall, the desk is relatively tidy. A touch at a clear space pops up in his role panel, and he chooses to unfurl the BUSINESS ADMIN cartouche. In it, the informal mailbox is glowing red: he presses at it and reveals a letter from his agent, who must have been up early. He listens to the message: would he review a new hyperbook? He drafts a brief acceptance, signs and despatches it, watching the route indicator show its arrival at the agent's wristerminal. The other ADMIN cartouches he ignores: they mostly give access to tax and VAT stuff, which he is content to leave to his accountant, as he is unable to comprehend such complexities. (Which was a considerable problem for himself and many others, when a date-dependent bug appeared in the accountant last April. Since then, he has changed to a formally-certified accountant; the old one was later convicted of unverified behaviour , and descheduled.

"Collapsing the Business Admin browser back into its cartouche, he chooses instead the Theorem Proving role. The summary of the Incoming Jobs Channel shows the originators, subject matter and fees of a couple of small jobs and one big one: "STL, Group Theory, 4000kW; Jones, prog transformation, 3500kW; Praxis, spec lang consistency, 32000kW". He fingers the last, getting the details showing deadline, originator's informal description, originator id and reference, agent's id and reference, expenses incurred (initialised to 0), sundry other administrative and commercial details, and of course the formal description of the problem itself. This he expands to a long scroll over the desk, then splits off some interesting bits into separate windows. He spends an hour or so just reading it, crawling over it on his hands and knees, often juxtaposing various parts, or revealing elided detail (and sometimes getting irritated when a window gets 'stuck' to his knee as he moves around).

"Going back to the Theorem Proving Rôle, he opens the Resources cartouche, and within that the NCC Theorem Proving Library access, which first reveals a welcoming green and yellow logo and a money socket (female), which is flashing because it isn't connected to anything. He prods at the Praxis job's expenses cartouche, revealing the empty 'outputs' list and a picture of a stack of money plugs (male). He drags one over towards the Library's socket and as soon as they are brought together, they recognize their type-compatibility and the connection is made with a flash and a beep: the Library's socket shows the Praxis job's logo and name, and the job's expense supplies list shows the NCC library logo, together with the charges as they clock up. The NCC library has now interrogated the TP's personal environment to find out where he likes to start navigating from, and the Library window shows this position. He navigates through, spawning off a few useful theorems and tactics in their own windows and comparing them with parts of the job's formal statement. The expenses clock up as he goes.

"Finally, he sets up some short propositions to do some preliminary checks. These are very boring: he decides to set an automatic theorem prover at them while he has some tea. The Theorem Proving Library contains an Automatic TP section, to which he navigates; a number of whimsically-named ATPs are offered by various commercial concerns. They have various strengths and weaknesses, some listed in the blurb attached to each, and some which he knows from experience or reputation. He selects a couple which advertise 'no fee without termination' and applies them to his propositions, setting an audio alarm triggered by termination before making his tea and going to sit on his bed to watch the seagulls.

"Variations Various reviewers have suggested alternatives to this basic scenario. Mark van Harmelen suggests a cordless keyboard is an essential accessory, hanging from a long shoulder-strap like those used by rock musicians; objects displayed on the floor are pointed to with the toes. Rather than crawling all over the theorems, Peter Lindsay would use a remote

control for armchair theorem-proving, using a light-gun to zap propositions from a distance. Ursula Martin envisages the even more remote cellphone model, in which the TP works whilst jogging along the clifftops.