Institution:	**University of Oxford**
Unit of Assessment:	**11 Computer Science and Informatics**
Title of case study:	**Automated Software Design and Verification (1)**

1. Summary of the impact (indicative maximum 100 words)

Analytical Software Design (ASD), based on Communicating Sequential Processes (CSP) and the Failures Divergences Refinement tool (FDR), has been developed and patented by the specially created Dutch company Verum. The new software, based on research in the UoA, was released in 2009 and has allowed customers to build rigorous, error-free software systems automatically by specifying state machines. ASD, using FDR as its verification engine, has produced many millions of lines of verified code for customers including Philips Medical Systems, Ericsson, FEI and ASML, who typically report at least a 50% reduction in costs and a 90% reduction in errors.

2. Underpinning research (indicative maximum 500 words)

Verum's technology incorporates Oxford's Failures Divergences Refinement (FDR). FDR is a model checker which itself uses Communicating Sequential Processes (CSP), a process algebra designed to help understand and analyse how systems interact with each other [2]. While the initial development of CSP dates back to the 1970s, it continues to be researched and developed at the University of Oxford [4, 5, 6], and it is these later works that underpin the impact.

The second, heavily updated and completely re-written version, FDR2, became available in 1994. The key designer of FDR and its algorithms has been Professor Bill Roscoe from the UoA throughout its history, though until 2007 the program was released and maintained by Oxford Spin-out Formal Systems. FDR development has been entirely the responsibility of the UoA since 2008 with releases of FDR2 up to 2.94 in 2012, and the completely re-written FDR3 in 2013. FDR's main function is to verify or refute refinement relations over a variety of semantic models—essentially it establishes that abstract specifications are satisfied by concrete programs. FDR is an extremely powerful process-algebra based model checker, and as such has been widely used in research and industry.

Between 1998 and 2001 at the University of Oxford, Guy Broadfoot studied the part-time MSc in Software Engineering. This included courses on CSP and FDR, the latter with Bill Roscoe. Broadfoot then investigated the possibility of combining the theory behind the Box Structure Development Method (BSDM), a technique created by IBM to develop commercial software, with CSP and FDR in order to use the power of formal verification in the context of a standard development process used for practical software engineering.

The concept was built upon between 2001 and 2005 by Dr Philippa Hopcroft, while a Research Fellow and Research Assistant at Oxford, funded latterly by Verum. This research took Broadfoot's insight and built on it, exploiting CSP's compositional properties and stepwise refinement to create software verification systems that could work on an industrial scale. This laid the foundations for systems capable of automated construction and verification of software - in theory providing commercial software developers with the power of formal verification tools, without having to learn anything about CSP or FDR [1]. This idea lies at the core of ASD.

Verum makes considerable use of CSP and FDR innovations developed by Roscoe and his group in the UoA since 1994, including the stable failures model [5], priority [5], lazy abstraction [3], the theory of expressibility in CSP [5] and compression [4,5]. In addition Verum has inspired new theoretical work such as the discovery of connections between priority, and a form of abstraction related to fairness [6].

Verum currently supports research on FDR3, which it plans to adopt by the end of 2013, and regularly consults with Roscoe and other UoA researchers about its use of CSP and FDR.

3. References to the research (indicative maximum of six references)

The three asterisked outputs best indicate the quality of the underpinning research.

[1] G.H. Broadfoot and P.J. Hopcroft. Combining the box structure development method and CSP. Proc IEEE conference on Automated Software Engineering 2004.
DOI 10.1109/ASE.2004.1342760
The key paper on ASD. Written when Hopcroft was an RA at Oxford.

[2] A.W. Roscoe. Model-Checking CSP. In Essays in Honour of C.A.R. Hoare,
Prentice-Hall 1994.
http://scholar.google.co.uk/scholar_url?hl=en&q=http://www.cs.ox.ac.uk/bill.roscoe/publications/50.ps&sa=X&scisig=AAGBfm1TKIoaJFcSjd2W-k9S3Fqn8Ix81Q&oi=scholarr&ei=j5hOUpXNJOfE7AbKgoH4Dw&ved=0CCwQgAMoADAA
The key paper on FDR.

[3]* A.W. Roscoe. CSP and determinism in security modelling. IEEE Security and Privacy Symposium 1995.
DOI http://dx.doi.org/10.1109/SECPRI.1995.398927
Introduced the concept of lazy abstraction, crucial in Verum's models.

[4]* A.W. Roscoe. The Theory and Practice of Concurrency, Prentice-Hall 1997.
http://dl.acm.org/citation.cfm?id=550448
Heavily cited book on CSP, described compressions, implementation of FDR2, specification techniques crucial to Verum.

[5]* A.W. Roscoe. Understanding Concurrent Systems. Springer 2010.
http://www.springer.com/computer/swe/book/978-1-84882-257-3
Sequel to [4], described priority, new compressions, built foundations for FDR3, introduced compilation-and-compression model for imperative state, all crucial to Verum.

[6] P. Hopcroft and A.W. Roscoe, Slow abstraction through priority. In Theories of Programming and Formal Methods. Springer LNCS volume 8051, 2013.
http://dx.doi.org/10.1007/978-3-642-39698-4_20
Reports theoretical research inspired by Verum, and its implementation and its use within ASD.

4. Details of the impact (indicative maximum 750 words)

Attempts to use automated formal verification in industrial-scale examples go back several decades, the motivation being that bringing verification into the software engineering process should ultimately replace most or all testing, create more reliable systems, and save both costs and time. This has proved a more challenging activity than was first hoped, though improvements both in verification technology such as model checkers (tools which explore a system's reachable states and report any errors found) and the computers available for running this technology are now enabling real progress.

The **path to impact** began with Guy Broadfoot's learning UoA research on his SEP MSc and formulating a plan to use it. After Hopcroft in the UoA researched and developed this idea with him [1], the path continued in 2004 when Broadfoot and a business partner set up Verum to develop this research into a product now known as Analytical Software Design (ASD). The company's goal was to convert the theoretical principles into a commercial system that could mathematically verify all possible outcomes during software design, and go on to automatically generate defect-free code. In turn that would save money, cut development time, and improve accuracy—attractive to businesses because software consumes over 50% of the development costs of technological products, and of that more than 40% is spent on testing and defect removal. [A]

The **impact since 2008** is represented by Verum's product: ASD:Suite, released in 2009, and by this product's impact on other organisations as detailed below

ASD:Suite is a design automation tool that combines industrial model-driven design with automated formal verification and code generation. In simple terms, developers use the high level ASD language to develop models which represent embedded systems as networks of state machines that are linked hierarchically; the network of components are wired together automatically to reflect the semantics of the runtime code generated. Verum has received US, European and Hong Kong patents for ASD.

The verification technology central to ASD:Suite is completely based on the UoA research set out above. The models are automatically translated into the process algebra CSP, enabling properties such as well-formedness of the original specification, deadlock and livelock freedom, correctness with respect to the specification, and responsiveness to be automatically verified or refuted using the model checker FDR. Once verification is complete, semantically equivalent source program code is automatically generated, in any one of a number of commercial software languages. Crucially, because the software system translates the state machines into CSP and the verified code is automatically generated, clients never have to interact with either CSP or FDR, but can take advantage of their power to test the design models automatically. Broadfoot (CTO at Verum) writes (to Bill Roscoe) as follows [F]:

"The ability to express a sufficiently broad class of systems and the necessary correctness properties, as well as our ability to push the boundaries of scalability in practice have resulted from your research, including the abstraction operations, compression and chase functions, and advanced specification techniques based for example on one-to-many renaming. We see our continued involvement with the on-going development of FDR and your development of new CSP/FDR techniques for us as vital to the successful creation and ongoing advances of our ASD:Suite product."

Verum sells licenses to use ASD:Suite, in which models developed by clients' engineers are processed and verified on servers operated by Verum. Between 2009 and 2013, ASD:Suite's users have included Philips Medical Systems, Ericsson, electron microscopy company FEI, and lithography systems firm ASML. The benefits of using ASD are reduced cost, reduced time-to-market, and improved accuracy. These have been demonstrated practically by its clients:

- FEI has reported a 4x reduction in cost per line of code [B].

- Ericsson has produced essentially error free software, with a sevenfold increase in productivity and up to 50% cost saving over conventional software development techniques. They also report being able to use only 'three or four people' when using Verum, rather than the 20 that would have been needed previously [B, G]. Ard-Jan Moerdijk, manager of the Dutch subsidiary of Ericsson Telecommunications, says that Verum's work is 'faultless' and T-Mobile and other of Ericsson's clients have not come across a single error in the programs created using Verum's package [G].

- The Technical University of Eindhoven has shown that use of the ASD:Suite at Philips Healthcare led to a 10x decrease in software defects [C].

Initially set up with funding supplied by the founders, in 2010 Verum secured an additional €1.5M to continue developing ASD [D]. In 2012 Verum was ranked 5th in the SMB (MKB) Innovation Top 100 in the Netherlands, being labelled the most innovative ICT technology company in the country [E]. In the period July 2009 – July 2013, ASD:Suite was used to create more than 250 million lines executable lines of code in the languages C, C++, C#, and Java [F], with individual generated systems frequently being over 500,000 lines of code. All of the models from which this code was generated had been verified on FDR. Each month in January – July 2013, ASD:Suite had on average over 74 active users. In July 2013, Verum had a workforce of 23 people.

While there have been many impressive applications of formal verification technology by verification experts, and a few generally usable applications for extremely restricted applications, such as Microsoft's SLAM applied to device driver compliance, we are not aware of any other application which, like ASD:Suite, has made verification of a very wide range of software accessible to software engineers who are not verification experts.

5. Sources to corroborate the impact (indicative maximum of 10 references)

[A] http://www.vdcresearch.com/webcasts/?id=168, Searching for the Total Size of the Embedded Software Engineering Market, VDC Research Inc.
Presentation assessing the size of the embedded software market in 2011.
[B] Verum ASD Suite Introduction, document held by University of Oxford.
Description of Verum's product ASD:Suite.
[C] http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6081983, Analyzing the effects of formal methods on the development of industrial control software.
Paper describing the application of ASD to an X-Ray machine.
[D] http://fd.nl/entrepreneur/young-entrepreneur/452776-1206/doorbraak-voor-softwarebedrijf-verum,
Translated here
http://www.verum.com/company/news-and-press/12-06-04/Breakthrough_for_Software_Company_Verum.aspx
'Breakthrough for Software Company Verum' article from "Het Financieele Dagblad" on 4 June 2012, by Hans de Jongh.
Newpaper article describing adoption of ASD by ASML.
[E] http://www.syntens.nl/innovatietop100/top-100-2012/Nummer-5.aspx, MKB Innovation Top 100.
Winners of Dutch innovation awards.
[F] Email correspondence with CTO of Verum and other Verum employees.
Testimonials about the role of the UoA's research in Verum's product and various aggregated statistics about the use of ASD by Verum's clients.
[G] http://fd.nl/entrepreneur/wereldveroveraars/634621-1211/brabantse-vinding-verslaat-indiase-softwaremakers, article from "Het Financieele Dagblad" on 26 November 2012, by Hans de Jongh.
Translated:
http://www.verum.com/company/news-and-press/12-11-26/Netherlands_invention_beats_Indian_Software_Developers.aspx
Article on Ericsson study demonstrating cost-effectiveness of using ASD in Europe versus outsourcing conventional development to India.