Institution: Durham University

Unit of Assessment: Computer Science and Informatics

Title of case study: From Formal Methods to Software Migration

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

The spin-out company CSM Ltd. was set up in 1991 to commercially develop Durham research on program transformation. Up until 1999, this company (which in the mid-90's became Durham Software Engineering Ltd. and subsequently Software Migrations Ltd.) and researchers at Durham University developed the *FermaT Workbench*: an industrial-strength assembler re-engineering workbench for program comprehension, migration and re-engineering. In 1999, Software Migrations Ltd. relocated to St. Albans and now has an extensive list of national and international clients. All its products (software and services) are built on the FermaT Workbench and has generated considerable revenue with this revenue strongly expected to rise steeply in the near future.

2. Underpinning research (indicative maximum 500 words)

Throughout the 1990's, there was considerable activity at Durham in relation to building program transformation systems involving mainly the staff members Professor Keith Bennett (since 1991; now emeritus professor) and Dr. Martin Ward (Senior Research Assistant 1987-90 and Research Fellow 1991-95). In chronological order (see [5]) the systems were:

- *The Maintainer's Assistant* (a software maintenance tool)
- *GREET* (a generic reverse engineering tool)
- *FermaT* (an interactive program transformation engine).

The resulting transformation engine *FermaT* forms the core of the *FermaT Workbench*: a collection of tools and databases based around the core technology of program transformations in the WSL language [5] (embedded in all the above systems). In turn, the *FermaT Workbench* forms the core of the commercial *FERMAT Migration and Comprehension Workbench* around which all products and services offered by Software Migrations Ltd. are based (see material downloadable from the company's web-site http://www.smltd.com and [5] where the opening sentence of the abstract states:

> "*The FermaT transformation system, based on research carried out over the last twelve years at Durham University and Software Migrations Ltd., is an industrial-strength formal transformation engine with many applications in program comprehension and language migration.*").

WSL (the "heart" of the *FermaT Workbench*) was developed at Durham in a collaboration between Durham's researchers and Software Migrations Ltd. so that by 1999 it had been transformed "*from a simple and tractable kernel language to a complete and powerful programming language*" [5]. By 1999, the WSL language included "*constructs for loops with multiple exits, action systems, side-effects, etc., and the transformation theory included a large catalogue of proven transformations for manipulating these constructs*" [5]. Moreover, by 1999 many of these transformations had been implemented in the *FermaT Workbench* [5].

Research typical of the evolution of the *FermaT Workbench* follows.

- In [1], the efficacy of the approach taken with *FermaT* is demonstrated by the reverse engineering of a range of challenging codes (written in JOVIAL and Assembler). The research successfully obtains a spectrum of requirements, ranging from using *FermaT* for code comprehension and simple restructuring, to reverse engineering from code to a high level of abstraction. In [2], the efficacy of FermaT was extended to software of an industrial strength.
- In [3], the flexibility of *FermaT* was demonstrated when a well-known and complex program presented by Knuth and Szwarcfiter to undertake topological sorting was successfully reverse engineered into a more structured program (without changing the semantics).
- Research in [4] involved an extensive analysis of a particular class of algorithms which present certain difficulties to formal verification. The approach in [4] is based on applying

- proven semantics-preserving transformation rules in a wide spectrum language. As is stated in [4,Sect.1.1.6], the tool *FermaT* is underpinned by the theory in [4].
- In [6], techniques to strengthen the reverse engineering of recursive programs were developed. These techniques extend those of Knuth and Bird (see [6]) and involve a powerful recursion removal and introduction operation which can be applied to a great many iterative programs which will put them in a suitable form for recursion introduction.

There was other underpinning Durham research including the formal proof (using Coq) of some transformations within WSL by Professor Malcolm Munro (emeritus professor).

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

[1] M.P. Ward and K.H. Bennett, Formal methods for legacy systems, *Journal of Software Maintenance*: *Research and Practice*, 7 (3) (1995) 203-219

- [ISSN: 1532-0618, DOI: 10.1002/smr.4360070305; 41 citations on Google Scholar as of 21st October 2013]

[2] M.P. Ward and K.H. Bennett, Formal methods to aid the evolution of software, *International Journal of Software Engineering and Knowledge Engineering* 5 (1) (1995) 25-47

- [ISSN: 0218-1940; DOI: 10.1142/S0218194095000034; 37 citations on Google Scholar as of 21st October 2013]

[3] M.P. Ward, Program analysis by formal transformation, *The Computer Journal* 39 (7) (1996) 598-618

- [ISSN 0010-4620; DOI 10.1093/comjnl/39.7.598; 25 citations on Google Scholar as of 21st October 2013]

[4] M. Ward, Derivation of data intensive algorithms by formal transformation: the Schnorr-Waite graph marking algorithm, *IEEE Transactions on Software Engineering* 22 (9) (1996) 665-686

- [ISSN: 0098-5589, DOI: 10.1109/32.541437; 37 citations on Google Scholar as of 21st October 2013]

[5] M.P. Ward, Assembler to C migration using the FermaT transformation system, *Proc. IEEE Conf. on Software Maintenance*, 30 Aug-3 Sep 1999, pp. 67-76

- [ISBN 0-7695-0016-1; DOI: 10.1109/ICSM.1999.792571; 59 citations on Google Scholar as of 21st October 2013]

[6] K.H. Bennett and M.P. Ward, Recursion removal/introduction by formal transformation: an aid to program development and program comprehension, *The Computer Journal* 42 (8) (1999) 650-673

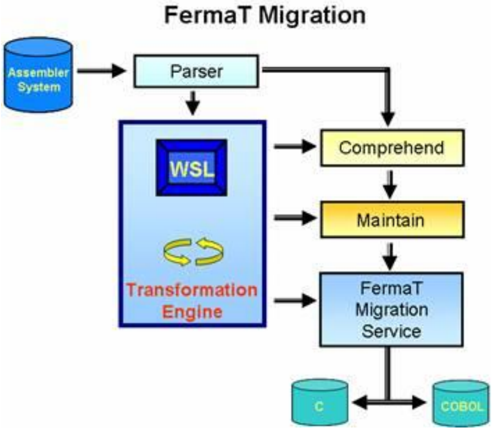
- [ISSN 0010-4620; DOI 10.1093/comjnl/42.8.650; 18 citations on Google Scholar as of 21st October 2013]

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

Up until 1999, whilst Software Migrations Ltd. (SML) was located at Durham, the company employed around 6 people. In early 2002, SML was bought by the current owner. The company is based in St. Albans and at this moment in time employs around 24 people [S4,S5] (20 of whom are technical experts).

All of SML's business is based on two core products [S1]:

- The FermaT Workbench, a set of software tools that can manage and maintain Assembler-based systems. The product is available for purchase from SML or can be used as part of a joint SML-customer project.
- The FermaT Migration Service, which uses technology that can transform Assembler into a more manageable language, is a service performed by SML consultants working with the customers' programmers.



SML's skill, expertise and product portfolio enable the re-engineering of crucial legacy codes so that these codes might be translated into a more structured style in a more maintainable programming language without affecting the program semantics. From [S4]:

> "*Therefore, we can confidently state that our core competency is detailed and precise analysis of Assembler and migration of Assembler to modern languages (C, COBOL and Java), based on a sound mathematical framework supported by considerable peer-reviewed research*."

From [S1]:

> *"Software Migrations Limited [SML] is the world's leading expert in legacy Assembler comprehension and migration. We help companies to get their core systems working for them, enabling clients to regain control of the technology which supports their business, improving the clarity, simplicity and flexibility of legacy Assembler systems."*

SML is uniquely placed in this regard and as such is in high demand. SML's customer base is composed of a combination of very large public corporations and government agencies in Europe and the USA (examples follow).

SML saves its clients vast amounts of money by protecting them from potentially catastrophic failures and deriving structured codes, from their legacy codes, in a form that is amenable to improvement and extension and so that these more structured codes are much easier to maintain. Whilst the financial savings of such protection are notoriously difficult to quantify, it is widely accepted that failure to protect can result in monumental costs (financial and otherwise). From [S1]:

> "*Forward thinking companies who address their core Legacy Assembler systems are finding that the benefits include:*
>
> - *Easier compliance with corporate governance regulation*
> - *Improved response to business development demands*
> - *Reduced IT costs and improved productivity*
> - *Improved new product time to market*
> - *Increased system flexibility*"

The largest project SML has undertaken comprised of over 15 million lines of Assembler listings representing over 9000 modules. SML has performed modernization tasks for over 50 customers ranging from deep analysis of their Assembler code to full conversion to COBOL and C (which is in live production). SML is actively engaged in a project to produce a pseudo-Java conversion so that full O-O Java might be produced [S5,S6].

Since 2008, clients of SML have included US Social Security Administration and Inland Revenue Services, US State Farm Insurance, California Department of Motor Vehicles, American Finance Credit Union, Standard Life, HP, IBM, HBOS, EDS, Microfocus, Royal Bank of Scotland, Tenovis and Tesseract [S2,S5,S6]. For example:

- State Farm Insurance [S2,S5], USA (www.statefarm.com; ranked 37th in the 2011 Fortune 500 list and widely regarded as the world's largest property and casualty insurance company): the work with State Farm Insurance, to transform Assembler code to Cobol, was done in collaboration with HP and the payment made directly to SML by State Farm Insurance was around $1.4 million.
- American First Credit Union [S2,S5], USA (www.americafirst.com; 7th largest credit union in the US in terms of total membership and 11th largest in assets in the US): the work with AFCU, to transform Assembler code to Cobol, was done in collaboration with IBM and the payment made directly to SML by AFCU was around $0.4 million.
- Department of Motor Vehicle Licensing in California, USA (www.dmv.ca.gov) [S2,S3,S5]: the work with DMV was done in collaboration with HP and the payment made directly to SML by DMV was around $6 million. The work was to transform 2.8 million lines of Assembler code into Cobol but within the 2.8 million lines of code were 1 million lines of EDL code (an assembler language developed by IBM). The workbench was also successfully used to deal with the EDL code too.

SML's 2012-2013 revenue was expected to be extremely significant with future expectations considerably in excess of this (see below) [S4]. SML has an American office at 3 Weybridge Court, Severna Park, MD 21146, USA [S5].

SML regularly works in collaboration with companies such as HP and IBM to the point that when either HP or IBM are working with a client with legacy code involving Assembler, their only real option (world-wide) is to use SML and this is what generally happens. No large integrator (like HP, IBM, etc., nor any smaller concern) has the specific technical Assembler skills that SML has, and they certainly do not have the unique products that SML has developed and own (and all of which are based on Durham research) [S2,S5,S6].

SML is currently responding to an IRS contract to transform 20 million lines of Assembler code. As of June 2013, SML has signed a partner agreement with IBM so that SML will receive 25% of the total revenue of the project which will amount to around $50 million. Negotiations are currently on-going between the IRS, SML and IBM with the eventual go-ahead strongly expected in early 2014 [S2,S4,S5]. This contract will stretch over 3-5 years. Whilst any agreement with the IRS will be outside the REF impact window, the agreement with IBM has been signed (within the REF impact window); this testifies to the tremendously positive impact that IBM believes SML can have on the whole project as well as the confidence that IBM has in SML's technology.

SML has funded PhD studentships in the past and its enlightened approach to funding academic research has meant that it is currently (widely recognised as) the outstanding global company as regards assembler code migration. It plans to continue its support for academic research so that it might develop this research into commercial tools for the analysis, restructuring and maintenance of code written in Java and other high level programming languages. SML is a real success story testifying to the efficacy of formal methods applied to modern commercial software development. Not only is SML generating financial profit (with this profit highly likely to rise steeply in the very near future), it serves as an outstanding exemplar as to the commercial benefits and rewards of employing the principles of and research in software engineering.

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

[S1] SML's website http://www.smltd.com (last accessed 21st October 2013)

[S2] Series of emails from the owner of SML and the Chief Technical Officer of SML relating to data supplied above and a telephone conversation between the owner of SML and Professor Iain Stewart, Durham University.

[S3] Two reports from SML to California Department of Motor Vehicles, 2012.

[S4] Response from SML to the IRS, 2012.

[S5] The auditors can contact Owner, SML for verbal confirmation.

[S6] The auditors can contact Chief Technical Officer, SML for verbal confirmation.