Transferring Real-Time Systems Research into Industrial Practice: Four Impact Case Studies

Robert I. Davis
University of York, York, UK

Iain Bate
University of York, York, UK

Guillem Bernat
Rapita Systems Ltd., York, UK

Ian Broster
Rapita Systems Ltd., York, UK

Alan Burns
University of York, York, UK

Antoine Colin
Rapita Systems Ltd., York, UK

Stuart Hutchesson
Rolls-Royce PLC., Derby, UK

Nigel Tracey
ETAS Ltd. York, UK

Abstract

This paper describes four impact case studies where real-time systems research has been successfully transferred into industrial practice. In three cases, the technology created was translated into a viable commercial product via a start-up company. This technology transfer led to the creation and sustaining of a large number of high technology jobs over a 20 year period. The final case study involved the direct transfer of research results into an engineering company. Taken together, all four case studies have led to significant advances in automotive electronics and avionics, providing substantial returns on investment for the companies using the technology.

2012 ACM Subject Classification Computer systems organization → Real-time systems

Keywords and phrases real-time systems, industrial impact, automotive, avionics

Digital Object Identifier 10.4230/LIPIcs.ECRTS.2018.7

Funding The work that went into writing this paper was funded, in part, by the ESPRC grant MCCps (EP/P003664/1). EPSRC Research Data Management: No new primary data was created during this study.

1 Introduction

This paper describes four impact case studies where real-time systems research has been successfully transferred into industrial practice. The four studies relate to:

1. Volcano: Guaranteeing the real-time performance of in-vehicle networks (Section 2).
2. RTA-OSEK and RTA-OS: The world's smallest commercial automotive real-time operating systems (Section 3).

3. RapiTime: A tool suite for analyzing the timing behaviour of real-time software (Section 4).
4. Visual FPS: The first CAA certified use of a fixed priority scheduler in an avionics system of the highest criticality (Section 5).

Preliminary versions of the first three impact case studies were informally published as white papers [30], [38], and [31]. Some of the content was also used in the University of York submissions to the Research Excellence Framework[1] (REF) assessment of UK Universities.

Each of the impact case studies is organised into the following subsections:

- *Impact Summary*: An executive summary of the impact achieved.
- *Background*: An overview of the industrial technology and practice prior to the research and development taking place.
- *Research*: An overview of the underpinning research, outlining the key insights and findings that contributed to the industrial impact.
- *Route to Impact*: The story of how the academic research was translated into viable commercial products.
- *Impact*: How the technology has been used, and by whom. (Note, due to non-disclosure agreements and commercial sensitivities, it is not always possible to give full details).
- *Beneficiaries*: Lists the beneficiaries and describes the benefits they have obtained by using the technology.
- *Future Challenges*: Sets out the main challenges in the specific technological area today.

The paper ends with a discussion of the key success factors and potential roadblocks. It is hoped that this information will be useful to others taking the exciting entrepreneurial step of trying to commercialise their research.

2 Volcano: Guaranteeing the Real-Time Performance of In-Vehicle Networks

2.1 Impact Summary

Controller Area Network (CAN) is a digital communications bus used by the automotive industry for in-vehicle networks. During 1994, research from the Real-Time Systems Research Group at the University of York introduced techniques that enable CAN to operate under high loads (approx. 80% utilisation) while ensuring that all messages meet their deadlines [65], [68], [67], [69]. This research led directly to the development of commercial products, now called Volcano Network Architect (VNA) and the Volcano Target Package (VTP). This Volcano technology (VNA and VTP) is now owned by Mentor Graphics. In recent years, VNA has been used to configure CAN communications for all Volvo production cars, with VTP used in the majority of Electronic Control Units (ECUs) in these vehicles, including the S40, S60, S80, V50, V70, XC60, XC70, XC90, C30, and C70; total production volume rising from 330,000 in 2008 to 530,000 vehicles per year in 2016. This Volcano technology is also used by Jaguar, LandRover, Aston Martin, Mazda, and the Chinese automotive company SAIC. It is used by the world's leading automotive suppliers, including Bosch and Visteon. It is also used by Airbus.

[1] http://www.ref.ac.uk/2014/

2.2 Background

Prior to the 1990s, cars used point-to-point wiring. This was expensive to manufacture, install and maintain. From 1991, the automotive industry began to use Controller Area Network (CAN) [21] to connect ECUs such as engine management and transmission control. Using this approach dramatically reduced the size, weight and complexity of the wiring harness, for example with CAN, a door system in a high-end car typically requires 4 wires, compared to 50+ with point-to-point wiring. The adoption of CAN led to significant cost savings and reliability improvements. It has supported a revolution in the complexity of automotive electronics, with the number of ECUs in a typical mainstream car increasing from 5-10 in the mid to late 1990's to 25-35 today.

CAN supports communications at typical bus speeds of 500Kbit/sec for powertrain applications and 125Kbits for body electronics. In a typical application, over 2000 individual signals (e.g. switch positions, wheel speeds, temperatures etc.) are sent in hundreds of CAN messages. There are deadlines on the maximum time that these messages can take to be transmitted on the bus. If a message fails to meet its deadline, then the reliability and functionality of the electronic systems can be compromised. This can lead to intermittent problems, and high warranty costs associated with 'no fault found' replacement of ECUs.

Messages queued by ECUs connected to a CAN bus compete to be sent on the bus according to their IDs, which represent their priority. Higher priority messages are sent in preference to those with lower priority. In the early 1990's, CAN messages were typically assigned IDs according to the data in the message, with a range of message IDs assigned to each supplier. Further, extensive testing was the only way of trying to verify that the messages would meet their deadlines. This was effective up to bus utilisations of about 30%; however, higher bus loads would result in deadline failures and intermittent problems.

2.3 Research

In 1994, three members of the Real-Time Systems Research Group at the University of York; Ken Tindell, Alan Burns, and Andy Wellings, introduced schedulability analysis of messages on CAN. This research [65], [68], [67], and [69] computed the longest time that each message could take from being queued by an ECU to being successfully transmitted on the bus and therefore received by other ECUs, referred to as the worst-case response time. This analysis enabled system designers to determine offline if all of the messages on a CAN bus could be guaranteed to always meet their deadlines during operation. This systematic approach was a significant improvement over the methods previously used in the automotive industry, which involved extensive testing, followed by hoping that the worst-case response time of every message had been seen.

This work also showed how to obtain optimal priority assignments for CAN messages. The research in [65] provided the fundamental analysis of message response times. This was extended in [68] to account for errors on the network, and integrated in [67] with information about the timing behaviour of the sending and receiving software. The analysis provided in [65], [68], [67], does not apply to all CAN hardware, some specific CAN Controller designs were shown in [69] to have relatively poor real-time performance, while others matched the requirements of the theory well. In 2007 research published by Davis et al. [33] corrected some flaws in the original analysis of CAN message response times, and was used by Mentor Graphics to check their Volcano Network Architect implementation. More recent research has addressed areas where the CAN controller hardware and the communications stack depart from the assumptions of the original research, such as non-abortable transmit buffers [54],

and the use of FIFO [35], [36] and other work-conserving [37] queuing policies, as well as systems where peak network load is reduced using offset message release times [73]. Further research has studied robust priority assignment policies [32], including the case where some messages are constrained to have specific IDs [34].

2.4 Route to Impact

The initial research on schedulability analysis for CAN [65] was disseminated at the 1st International CAN Conference in 1994. As a direct result of this Ken Tindell was approached by Antal Rajnak, then working for Volvo Car Corporation. In April 1995, Ken Tindell and Robert Davis founded a start-up company called Northern Real-Time Technologies Ltd. (NRTT) to exploit the research in [65], [68], [67], and [69]. This company was contracted by Volvo Car Corporation to develop a CAN software device driver library and associated configuration tools [47], now referred to as the Volcano Target Package. Over the next two years, NRTT developed the Volcano Target Package through 4 major versions, and ported it to more than 10 different microprocessors used in the Volvo S80 and other automotive applications. At the same time, the message priority assignment policies and schedulability analysis techniques first introduced in [65], [68], [67], [69] were implemented in a CAN message configuration and analysis toolkit called Volcano Network Architect (VNA). The initial versions of VNA were developed by Kimble AB (a Swedish company founded by Antal Rajnak), working in conjunction with NRTT. Rights to the initial versions of the Volcano Target Package were transferred to Volcano Communications Technologies AB (a Swedish company founded by Antal Rajnak) which subsequently developed fully commercial versions of the Volcano technology (VNA and VTP), before being acquired by Mentor Graphics in 2005 [45]. From 1997 onwards the Volcano technology was used in the Volvo XC90, S80, S/V/XC70, S60, S40, and V50. When Volvo was bought by Ford in 1999, this technology was adopted by Ford Premier Automotive Group (Jaguar, Land Rover, and Aston Martin).

As part of its work on the Volcano technology, in 1995/6 NRTT consulted with Motorola, strongly influencing the hardware design used in the on-chip peripheral MSCAN controller [47], [43] (Section 4.2 of that document). This design used a 3 transmit buffer solution to ensure that the MSCAN controller can send out a stream of high priority CAN messages without releasing the bus – essential in achieving high bus utilisation without deadline failures. The 3 transmit buffer solution reduced the silicon area, and hence the unit cost of the hardware, compared to a 'full' CAN controller with 15 or 16 transmit buffers. This gave Motorola a competitive advantage, and reduced unit production costs for Volvo. Since 1997, microprocessors using MSCAN have been used in the door modules and other ECUs in a wide range of Volvo cars. In 2007, the analysis in [33] was used by Mentor Graphics to verify that the analysis provided by VNA [44] was correct. Further details of the Volcano Target Package and Volcano Network Architect can be found on Mentor Graphics' website [43], [44] with detailed descriptions given in [56].

2.5 Impact

The initial research [65], [68], [67], and [69] from 1994 was exploited in the design of CAN network layer software, called the Volcano Target Package (VTP), and network schedulability analysis tools, called Volcano Network Architect (VNA). The Volcano Target Package is deployed in ECUs, while Volcano Network Architect is used to configure networks and to ensure that the configurations obtained result in all messages meeting their time constraints. The research was initially exploited by a start-up company called Northern Real-Time

Technologies Ltd. (NRTT) that developed the first versions of the Volcano Target Package for Volvo Car Corporation (VCC) and worked in conjunction with Kimble AB to develop the first versions of Volcano Network Architect. Fully commercial versions of the Volcano technology (VNA and VTP) were later produced by Volcano Communications Technologies AB, which was sold to Mentor Graphics in 2005 [45].

In 2018, the Volcano Target Package is available for more than 30 different ECU microcontrollers [48], including: Fujitsu 16LX, FR Series; Hitachi H8S, SH7055, SH7058; Infineon C16x, TC179x, TC176x, XC800, XC2000; Renesas M16C, R32C/M32C; Freescale HC08, HC12, MC683xx, MPC5xx, MAC71xx; S12, S12X, MPC55xx, MPC 56xx; Mitsubishi M32R, MC32C; PowerPC; National CR16; NEC V85x, 78K0; ST Microelectronics ST9, ST10; Texas Instruments TMS470; Toshiba TMP92/TMP94.

Since the introduction of the Volvo S80 in 1998, Volcano Network Architect has been used to configure CAN communications in all new Volvo production cars, with the Volcano Target Package used in the majority of Electronic Control Units (ECUs) in these vehicles. During the period 2008 – 2016, this includes the S40, S60, S80, V50, V70, XC60, XC70, XC90, C30, and C70; total production volume 330,000 to 530,000 vehicles per year [70].

The Volcano technology (VNA and VTP) is also used by Jaguar, LandRover and Aston Martin. Since 2007, this technology has been used in its own branded vehicles by the Chinese automotive giant SAIC [46]. In 2012, Mazda announced that they would be using Volcano technology in order to make more efficient and reliable use of CAN in vehicles featuring their "Skyactiv Technology" [49]. The Volcano Target Package is also used by the world's leading automotive suppliers, including Bosch and Visteon.

2.6 Beneficiaries

Volcano Network Architect, and the Volcano Target Package software that conforms to its assumptions, enable system architects at automotive manufacturers to configure in-car networks using CAN such that all of the messages are guaranteed to meet their deadlines at bus loads (utilisations) of up to approx. 80%. This compares with a maximum of approx. 30% using the approach otherwise prevalent in industry, where message IDs (priorities) are assigned in groups according to ECU supplier, and extensive testing and a large engineering margin for error is used to gain some confidence that message deadlines will be met. Achieving higher bus utilisation enables far more functionality to be supported using the same bus speed and communications hardware, providing those automotive manufacturers that adopt this technology with a key competitive advantage. With higher bus utilisations, more ECUs can be connected to the same network, and the network can support a larger number of signals and messages. Wiring complexity can be reduced, with fewer connectors, increased reliability, and improved brand image. Further, there is enhanced support for the addition of lucrative 'software-only' options.

These benefits are summarised in the Volvo Technology Report [25]:

"The advantages to Volvo of the development and application of Volcano include: Production cost benefits due to high bus efficiency (four times as many signals can be transmitted at half the baud rate). Development cost benefits (in the form of a single, proven implementation which is much cheaper than multiple implementations by suppliers and conformance testing by Volvo). Improved network reliability, resulting in higher product quality. Reduction in Volvo´s test load. Reduction in supplier´s test load. High degree of flexibility (useful in many situations). Recognition of the real-time problem (Volvo developed solutions before the problem had been recognised generally)".

Although [25] was written in 1998, the benefits of using this technology remain the same today. They are highlighted in 2006 [46] in relation to the Chinese automotive giant SAIC:

"By using Volcano, network design is made easy and predictable, guaranteeing data communication, which reduces the verification effort to almost zero and eliminates warranty and change costs caused by networking issues."

Similarly, in 2012 [49]:

"Mazda's use of VNA has enabled significant improvements in network efficiency and reliability." ... "This procedure increased the network utilization and significantly reduced the testing requirements and time".

The research on CAN also led directly to the design by Motorola (now Freescale) of a low-cost on-chip CAN peripheral MSCAN [25], [55] that requires less silicon area than a 'full' CAN controller, and so reduces unit costs in production.

In summary, car manufacturers and their sub-suppliers have benefited from the research in terms of reductions in development, production, and warranty costs. Development costs have been reduced via improvements in the time taken to verify network timing behaviour, reducing the cost of testing, and time-to-market. Production costs have been reduced via the ability to run in-vehicle networks at high loads while ensuring that all message deadlines are met. This has enabled increasing amounts of functionality to be accommodated using the same low cost CAN hardware. Improvements in network reliability, via off-line guarantees that messages will always meet their deadlines, have reduced warranty costs, in particular, costly 'no fault found' ECU replacement. In a competitive marketplace, benefits to the car manufacturers have been passed on to the consumer, in terms of vehicles that are less expensive, yet have more functionality, and better reliability.[2]

2.7 Future Challenges

The future challenges in this area originate from:

- The use of multiple networks, often of different types, with signals and messages transferred between them. Here, gateway policies, signal packing, and prioritization all influence end-to-end response times.
- The need to efficiently utilise network bandwidth. Message priority assignment, offset assignment and signal packing all influence the useful bandwidth that can be employed before messages begin to miss their deadlines. The recent CAN-FD protocol increases network speed during data transmission.
- The need to deal with legacy applications and ECUs. It is rare that any automotive system begins with a clean sheet design.
- Security issues. Connection of in-vehicle networks to the internet raises significant security concerns.

3 RTA-OSEK and RTA-OS: The World's Smallest Automotive Real-Time Operating Systems

3.1 Impact Summary

Research [2], [3], [24], [4], [66] published by the Real-Time Systems Research Group at the University of York from 1993 to 1995 was exploited in 1997 to design an exceptionally efficient Real-Time Operating System (RTOS), used in automotive Electronic Control Units (ECUs), and its associated schedulability analysis tools . By 2017, the RTOS had been deployed

[2] Accounting for inflation, the average car purchased in the USA in 2015 was less expensive than the average car purchased during 1990 ($25,300 versus $27,300).

in over 1.25 billion ECUs. It has been standardised upon by many of the world's leading automotive powertrain systems and chassis electronics suppliers, and is used in cars produced by nearly all of the world's major car manufacturers.

3.2 Background

In real-time embedded systems, such as the ECUs used in vehicles, system functionality is decomposed into multiple software tasks running on a microprocessor. The system requirements place time constraints on these tasks. Hence a task may be required to execute every 10 milliseconds, read and process data from sensors, and output its results within a specific time constraint or deadline. When there are multiple tasks with different periods and deadlines running on the same microprocessor, an RTOS is needed to schedule when each task should execute. It is essential that all of the tasks are guaranteed to meet their deadlines during operation; otherwise the system may suffer from intermittent timing faults that compromise its functionality and reliability.

Given the complex behaviour of these systems, it is impossible to obtain a 100% guarantee that tasks will always meet their deadlines via testing. Instead, a rigorous scientific and systematic solution to this problem is schedulability analysis; a set of techniques used to determine off-line if each task can be guaranteed to meet its deadline under a specific scheduling policy. Schedulability analysis is used to compute the worst-case response time, the longest time that can elapse from a task being released to it outputting its results and completing execution. If this is less than the deadline, then the task can be guaranteed to always meet its time constraints.

3.3 Research

In the early 1990's seminal research into schedulability analysis [2], [3], [24], [4], and [66] for fixed priority pre-emptive scheduling, originally called Deadline Monotonic Schedulability Analysis but now widely referred to as Response Time Analysis, was introduced by the Real-Time Systems Research Group at the University of York.

This analysis is applicable to fixed priority scheduling, and a task model that accurately accounts for the detailed timing behaviours of tasks in automotive systems. These timing behaviours include: tasks that are invoked sporadically (i.e. with minimum inter-arrival times, but not necessarily strictly periodically in time); tasks with deadlines that are less than their periods and prior to completion [2], [3] — accounting for tasks that need to make a response prior to their next invocation to avoid buffer overruns, and to carry out further computations after a response has been made, in preparation for the next cycle; tasks with offset release times [4] – used as a means of avoiding peak load in short time intervals; tasks with jittered released times [66] – that are triggered by the arrival of messages that can take a variable amount of time to be transmitted, and tasks that share resources [2], [3] – such as data structures and peripheral devices used for communication. The analysis also accounts for the overheads of a well-designed RTOS [24].

This research therefore introduced for the first time, schedulability analysis that could be applied in practice to commercial real-time systems, providing a rigorous approach to obtaining timing correctness. This was recognised in the EPSRC International Review of Computer Science undertaken in 2002: *"These researchers are credited with a significant body of research in static real-time scheduling theory. They have also demonstrated how to employ these theoretical results in practice, by accounting for networking and operating system overheads. This combination of theory and practice has resulted in important and practical applications of their work."*

The techniques developed also built upon other important research contributions such as the Stack Resource Policy [7] for resource locking.

3.4 Route to Impact

In 1997, Robert Davis and Ken Tindell co-founded a company called Northern Real-Time Applications (NRTA) Ltd., with the aim of developing an RTOS and schedulability analysis tools specifically tailored to automotive applications that use low cost microcontrollers.

There were two fundamental design goals:

1. The real-time behaviour of systems built using the RTOS must be fully analysable using schedulability analysis tools. In other words the behaviour of the RTOS must match the assumptions of the underpinning schedulability analysis techniques.
2. The memory and execution time overheads of the RTOS must be significantly less than those of any other RTOS available for use in automotive applications.

Robert Davis led the team that developed the SSX5 RTOS and associated schedulability analysis tools (originally called the "Time Compiler", later "Real-Time Architect (RTA)" and "RTA-OS Analysis Visualizer"). The schedulability analysis tools implemented Response Time Analysis as introduced in [2], [3], [24], [4], and [66]. The SSX5 RTOS was developed precisely to meet the assumptions of this analysis. The execution time overheads (of preemption, task termination, interrupt service routine entry and exit, and all system calls that can cause context switches) were minimised and made constant, independent of the number of tasks, allowing them to be accurately measured and integrated into the schedulability analysis implemented in Real-Time Architect.

The memory overheads of applications built on SSX5 were radically reduced by comparison with other automotive RTOSes. This was achieved via the use of single-stack execution and compile time, i.e. off-line, configuration of the RTOS data structures to minimise RAM usage. NRTA attracted significant venture capital funding in 1998 (£1 million from 3i) and again in 2000 (£9.2 million from 3i and TecCapital). In 2001, the company changed its name to LiveDevices Ltd.[3]

In March 2003 LiveDevices was sold to ETAS GmbH, a wholly owned subsidiary of Robert Bosch GmbH. The reason for the trade sale was that Robert Bosch had benchmarked RTA-OSEK and found it to be significantly more efficient than its subsidiary's Ercos RTOS. Rather than attempt to write a new OSEK RTOS from scratch and compete with LiveDevices, ETAS chose to buy the company, bringing the RTA-OSEK technology and the 20+ LiveDevices engineering team in-house.

3.5 Standards

During the development of the SSX5 RTOS, the automotive industry was working on standards via the OSEK organisation. As a Technical Committee Member of OSEK, NRTA influenced the OSEK OS standard [52] ensuring that the basic conformance classes (BCCx) could be achieved with a single-stack RTOS, leveraging the execution time and memory savings which that approach facilitates [29]. NRTA modified the SSX5 RTOS to comply with the OSEK standard, in the process renaming the product: RTA-OSEK.

[3] This name change was marketing led as the company was also developing Internet-of-Things technology, including a very small TCP-IP stack. This technology was not commercially successful; in hindsight it was around 10 years ahead of its time.

Subsequently, ETAS, as a premium partner of the AUTOSAR (AUTomotive Open System ARchitecture) partnership [6], have been heavily involved in specifying the AUTOSAR operating system standard [5], which extends the OSEK operating system standard. ETAS derived an AUTOSAR compliant RTOS called RTA-OS from RTA-OSEK [42].

3.6 Impact

As of 2018, ETAS sells two versions of the RTOS, RTA-OSEK and RTA-OS compliant with the OSEK and AUTOSAR operating system standards respectively.

The RTOS is currently available for more than 50 different ECU microcontrollers [42] including: Renesas: V850E, SH2, SH2A, H8S, H8SX, M16C; Xilinx Microblaze, PPC405 Core; Texas Instruments TMS470P, TMS570P; Infineon Tricore TC17x6, C166, XC2000; Freescale Star12, MPC555, MPC55xx, S12X, MPC56x, HC12X16, HC08, HCS12; Fujitsu 16LX; Analog Devices Blackfin, STMicroelectronics ST30, ST7, ST10.

RTA-OS is also available for the following multi-core processors: Infineon Aurix, Freescale MPC57xx, Renesas RH850, STMicroelectronics SPC57x, and the Xilinx Zynq-7000 family.

ETAS customers for the RTOS cover a wide range of application areas within Automotive Electronics. It has been standardised upon (used by default in all ECUs) by many of the world's leading automotive powertrain systems and chassis electronics suppliers, and is used in cars produced by nearly all of the world's major car manufacturers. By 2017, the RTOS had been deployed in over 1.25 billion ECUs. This number is increasing at a rate of between 1 and 2 million new ECUs per week.

3.7 Beneficiaries

Use of the RTOS and its associated schedulability analysis tools has benefitted automotive manufacturers and their Tier 1 suppliers in the following ways:

(i) A reduced memory footprint means that cheaper microcontroller variants with smaller on-chip RAM / Flash memory can be used. (The code size of RTA-OS is typically in the range 1 Kbytes to 1.5 Kbytes depending on the processor – making it, to the best of our knowledge, world's smallest commercial AUTOSAR OS.[4] This has reduced unit costs in production.

(ii) The very low execution time overheads[5] of the RTOS mean that more functionality can be included on a given low cost microprocessor reducing costs by avoiding the need for hardware upgrades to more capable but expensive devices.

(iii) A reduction in the time spent debugging intermittent timing issues. Schedulability analysis and appropriate use of proven real-time mechanisms have enabled off-line analysis of task response times, reducing system integration time and testing effort, and improving reliability.

For these reasons the world's major ECU suppliers and car manufacturers have adopted this technology. In a competitive market, some of these benefits will have been passed on to their customers in the form of cheaper, more reliable vehicles.

The Automotive Electronics market is both huge and highly competitive, with electronics now contributing 15-30% of overall vehicle production costs. For the reasons given above, the world's leading Automotive OEMs and Tier-1 suppliers have adopted the RTA-OSEK and RTA-OS operating systems. They have done so for the substantial benefits it brings to them and to their customers. The technology has led directly to the creation and sustaining, over a period of more than 15 years, of a large number of high technology jobs in York, UK.

[4] See section 8 of [41] for an example of the RTA-OS ROM and RAM usage.

[5] See section 8.5.1 of [41] for an example of the overheads in CPU cycles and nano-seconds for different types of context switches, along with diagrams explaining the precise measurements.

3.8 Future Challenges

Automotive systems are now moving towards implementations on multi-core hardware. This leads to the following challenges:

- The use of high performance multi-core hardware, with shared interconnects and other shared hardware resources makes it significantly more difficult to obtain an accurate understanding of the execution time behaviour of tasks, due to issues of cross-core interference. In some cases this interference can be so severe that guaranteed performance using multiple cores may be no better than could be obtained by utilising just one core.
- Synchronization via non-preemptive execution is no longer effective in a multi-core environment, creating significant difficulties in porting legacy applications. More complex and potentially substantially less efficient synchronization and locking mechanisms need to be employed, and large amounts of code potentially re-factored.
- High performance multi-core hardware means that it is cost effective to integrate different applications that would otherwise have run on independent ECUs onto the same hardware platform. These different applications have different criticality levels which leads to a host of interesting problems. Mixed criticality systems are currently a hot topic in real-time systems research [22].

4 RapiTime: A Tool Suite for Analyzing the Timing Behaviour of Real-Time Software

4.1 Impact Summary

Research [18], [16], [28], [27], [17] from the Real-Time Systems Research Group at the University of York published in 2002-2005 resulted in a measurement-based Worst-Case Execution time (WCET) analysis technology now called RapiTime, which was transferred to industry via a spin-out company, Rapita Systems Ltd, founded in 2004. The technology enables companies in the aerospace, space and automotive industries to reduce the time and cost required to obtain confidence in the timing correctness of the systems they develop. The RapiTime technology has global reach having been deployed on major aerospace and automotive projects in the UK, Europe, Brazil, India, China, and the USA. Key customers include leading aerospace companies as well as major automotive suppliers.

4.2 Background

Determining the longest time that software components can execute on a microprocessor, referred to as the Worst-Case Execution Time (WCET), is a key issue in the development of real-time embedded systems in the aerospace and automotive industries. Here, intermittent timing failures caused by software exceeding its budgeted execution time can lead to operational problems, reliability issues, and in some cases catastrophic consequences. In these applications the WCET of software components needs to be tightly bounded to avoid the need to over-provision hardware in terms of faster, but more costly processors.

Prior to this research, there were two main approaches to WCET estimation; end-to-end measurement and static analysis. End-to-end measurement techniques insert profiling code into the software. During testing this profiling code records the end-to-end execution time of each invocation of each software component. End-to-end measurement alone typically under-estimates the WCET, and provides little confidence that timing constraints will always be met during operation. Static analysis techniques analyse the software object code and compute the WCET using a model of the timing behaviour of the microprocessor. This

is done without running the code. Using static analysis alone has the disadvantage that the computed WCETs depend on the model of the processor and its hardware acceleration features; as processor technology advances this becomes more complex, and expensive to determine, and in some cases may not be possible due to a lack of detailed information.

4.3 Research

During the NextTTA project (2002 to 2004), Guillem Bernat, Antoine Colin, Stefan Petters, and Alan Burns developed a set of hybrid and probabilistic techniques for WCET analysis [18], [16], [28], [27], and [17], now referred to as RapiTime. The RapiTime approach combines static analysis of the structure of the source code with timing measurements taken during testing, which record the execution time of short sub-paths through the code. RapiTime recognises that the best possible model of an advanced microprocessor is the microprocessor itself and therefore uses online testing to measure the execution time of short sub-paths in the code. By contrast, offline static analysis is the best way to determine the overall structure of the code and the paths through it. Therefore RapiTime uses path analysis techniques to build up a precise model of the overall code structure and determine which combinations of sub-paths form complete and feasible paths through the code. Finally the measurement and path analysis information are combined using mathematical techniques to compute WCETs in a way that accurately captures the execution time variation on individual paths due to hardware effects.

This novel approach combines the advantages of both measurement and static analysis techniques while avoiding the majority of their drawbacks. Unlike static analysis, it does not require the expensive and time consuming production of a precise timing model for each new microprocessor variant and its hardware acceleration features, and so is portable to a wide range of different microprocessors. RapiTime is also viable when the only accurate timing model that is available is the microprocessor itself. Further, RapiTime does not require the manual annotations that static analysis alone needs to establish essential information about control flow. This reduces the amount of engineering time required before meaningful results can be obtained, and removes a potential source of errors. Compared to measurement, RapiTime is able to identify the worst-case path and compute the overall WCET of software components from the WCETs of sub-paths when not all of the complete paths through the code have been executed. This significantly reduces the amount of testing required to verify timing correctness. For a detailed discussion of the advantages / disadvantages of static and measurement based approaches to timing analysis, the interested reader is referred to [71].

4.4 Route to Impact

During the EU FP5 NextTTA project, Guillem Bernat, Antoine Colin, Stefan Petters, and Alan Burns, introduced research on hybrid measurement-based WCET analysis. This approach combined both measurement and static analysis techniques to accurately estimate the WCET of complex software components running on advanced microprocessors. As part of the project, they also developed a prototype WCET analysis tool called pWCET [17]. This tool was evaluated on an Audi drive-by-wire system. Audi was an industrial partner in the NextTTA project. Audi's expression of interest in pWCET and its capabilities led directly to the formation of a spin-out company to transfer this technology into industry.

In 2004, Guillem Bernat, Ian Broster, Antoine Colin, and Robert Davis founded a spin-out company called Rapita Systems Ltd. (www.rapitasystems.com) to commercialise the technology and bring it to market. All rights to the technology and prototype tools were transferred to the company by the University of York in exchange for shares in the company.

In 2005, Rapita Systems received £200k of funding from Viking Investments Ltd. and an associated group of Business Angels [72]. Following the initial technology transfer, the pWCET prototype was re-implemented as a commercial quality tool and re-branded as RapiTime. RapiTime has since been extended to support analysis of systems written in C++ as well as the C, and Ada programming languages. RapiTime has been complemented by RapiCover, an on-target structural code coverage tool, and RapiTask, a tool which enables users to visualize high-level system scheduling, locate rare timing events such as race conditions, and verify actual timing behaviour. Both RapiCover and RapiTask use the underpinning RapiTime technology for code instrumentation and analysis. RapiTime, RapiCover, and RapiTask form part of the Rapita Verification Suite (RVS).

In 2006, BAE Systems used RapiTime on the Hawk Advanced Jet Trainer project [60]. Here, RapiTime was used to identify opportunities for WCET reduction, thus creating headroom for new functionality to be added to the system, while avoiding the need for a costly hardware upgrade. Using RapiTime, BAE identified that just 1% of hundreds of thousands of lines of code contributed 29% of the overall WCET. By focusing optimisation efforts on this 1% of the code, they were able to reduce the WCET by 23% [19]. Further, RapiTime was quantified as being able to identify timing problems with less than 10% of the effort of previous approaches, potentially saving months of work. As a result Rapita received a BAE chairman's award for Innovation in the category Transferring Best Practice.

In April 2016, Rapita Systems Ltd. was sold to Danlaw Inc. in a trade sale [64]. (Danlaw is a global connected vehicle, automotive electronics and embedded engineering enterprise with facilities in USA, Europe, India, and China).

4.5 Impact

As described in the previous section, research from the Real-Time Systems Research Group at the University of York was exploited in the development of an innovative Worst-Case Execution time (WCET) analysis technology called RapiTime. This technology was transferred to industry via the formation in 2004 of a spin-out company; Rapita Systems Ltd.

RapiTime has been deployed on, and is in continuous use on, a number of major long-term space, aerospace and automotive projects world-wide, examples include: Flight Control Computers [61] and FADECs (Full Authority Digital Engine Control); Alenia Aermacchi (Italy) Flight Control System for the M-346 military transonic trainer [62] (since 2010), and various projects for the European Space Agency (ESA) (since 2008). RapiCover has also been qualified for MC/DC coverage of the DO-178B DAL A Flight Control System of the M-346 [63]. Rapita has also won significant export orders to China via its distributor Cinawind.

4.6 Beneficiaries

RapiTime enables companies in the aerospace and automotive electronics industries to reduce the time and cost required to obtain confidence in the timing correctness of the systems they develop. It provides a cost-effective means of targeting software optimisation, such that new functionality can be added to existing systems without the need for expensive hardware upgrades. Further, RapiTime is portable across a wide range of different microprocessors, meaning that companies can use the same technology across multiple projects without the need for re-training or adoption of multiple solutions.

A major aerospace supplier described the benefits of using RapiTime to identify timing problems during continued development of a Flight Control System as follows: *"The biggest benefit that RapiTime brought to our development process was just how quickly we could get comprehensive timing measurements from our tests. Not only did we reduce our effort*

requirements for the testing, but we could use our results in ways that were infeasible before. It is now significantly faster for us to identify a timing issue, update the software to resolve the issue, test the updated software and verify that it's fixed" - Wayne King, Engineering Fellow – 30th July 2009.

Without RapiTime, the timing measurement and analysis process needed to determine WCETs has to be done manually. This is a painstaking and error prone process that takes considerable time and effort. It also needs to be repeated when changes are made to the application software. Further, the manual process provides no information about the worst-case path, or the contribution of different sections of code to the WCET. This makes code optimisation an ad-hoc, ineffective and inefficient process, as optimising for the worst-case is very different from optimising for the average case.

Alenia Aermacchi engineers working on the M-346 Flight Control System in 2010 said, *"the main advantage [of using RapiTime] is the possibility to identify software bottlenecks that can be subject to optimisation. Without RapiTime the mandatory code optimisation would have been done without the knowledge of where to concentrate the efforts."* [62]. Overall, *"Using RVS, customers have cut the worst-case execution time of large scale, legacy applications by up to 50% with only a few days effort, and significantly reduced unnecessary testing and instrumentation overheads"* [61].

Embraer used Rapita's RVS tool suite to capture WCET and stack usage data for DO178B level A Flight Control Systems (FCS) [58]. Because it was not necessary to manually design a test case for the worst-case path, significant effort was avoided, saving time and money. *"We have successfully shown the viability of using RapiTime to measure WCET. It was able to support our hardware platform and once the system was set up, the analysis method could be repeated with relative ease. With the WCET results, time partitioning was easily configured in the platform for the FCS application. Processing resources could be optimized by tightening the time window, even leaving some room for future expansion"*, Felipe Kamei, Embraer [58].

Infineon asked Rapita Systems to use RapiTime to look for optimization opportunities to reduce the execution times of the SafeTCore drivers which form part of Infineon's PRO-SIL concept. (These drivers are functionally independent of micro-controller hardware and can run on all micro-controllers in Infineon's TriCore family). The timing analysis part of the case study focused on 5 Tricore functions, giving up to 43.9% reduction in the WCET [59].

4.7 Future Challenges

In the next 5 to 10 years, complex multi-core and many-core systems will present an extreme challenge in terms of the difficulty involved in obtaining tight worst-case execution time estimates.

- The use of high performance multi-core hardware, with shared interconnects and other shared hardware resources means that execution times can be heavily impacted by contention over shared resources by co-running tasks on other cores. WCETs obtained in isolation can thus be substantially optimistic, when compared to the values for an operational system that runs applications on multiple cores.
- Obtaining context independent WCETs presents a significant challenge, since it is not obvious what pattern of co-runner execution will produce the most interference on shared resources. Even if a fully context independent WCET can be found, then it may be substantially pessimistic, compared to the actual WCET in the context of the deployed system.

Hardware and software techniques which ensure isolation from the effects of co-runner contention may be effective here. Another promising approach, is to use measurement-based probabilistic timing analysis techniques.

5 Visual FPS: The First CAA Certified use of a Fixed Priority Scheduler in an Avionics System of the Highest Criticality

5.1 Impact Summary

Fixed Priority Scheduling (FPS) research from the Real-Time Systems Research Group at the University of York was exploited to make the design and maintenance of the software in Rolls-Royce's Full Authority Digital Engine Controllers (FADEC) more efficient in terms of resource usage and cost [51]. The most notable benefit to Rolls-Royce was that they did not need to procure new more powerful processing hardware for a project where the processor which they normally used had run out of capacity. Unlike conventional systems, a processor board for a safety-critical avionics system costs thousands of pounds and has lead times of between one and two years. Changing the hardware would have meant that the software team would not have had access to the actual target until very late in the development, and the project would most likely have been late incurring significant penalties. Overall, the risk of such a change was unacceptable.

5.2 Background

FADECs are responsible for the control and monitoring of aircraft engines. They play a vital role in not only the reduction of hazardous events related to the aircraft engine, but also the overall safety and certification of the aircraft. FADECs do much more than inject fuel and control the engine. They help keep both the aircraft's cabin and fuel at the right temperature, receive information and commands from the cockpit and send back information, they also log information about the engine for future maintenance, and play other vital roles such as helping the aircraft brake on landing via the use of thrust reversers. Over time, this has led to an increase in the amount of software in the system, most of which is hard real-time. The timing requirements that have to be guaranteed span not only deadlines, but also tight jitter requirements. These requirements have to be guaranteed for both independent tasks and precedence constrained tasks, referred to as transactions.

For many years, the avionics industry used static scheduling to try and meet the timing requirements. Despite the use of automated tools, e.g. search-based algorithms for choosing task attributes [23], a number of issues remained unresolved. Firstly, the static scheduler places restrictions on the timing requirements, for example minimising the number of periods used, and making them harmonics of a single period. Secondly, and more importantly, the schedules become hard to maintain as the number of tasks and their execution times change as the system's build progresses. Here, the software in the FADEC is slowly integrated through a number of carefully considered phases; however, most approaches to designing the schedule do not consider maintenance and ensuring the minimum change between synthesised schedules [40] nor the similarity of schedules between functional modes [39]. This is significant as changes to the order in which tasks execute changes both the timing and functional characteristics which makes regression testing a much larger and hence more costly activity. Finally, as with many systems the processor was almost fully utilised (approaching 100%), making meeting the timing requirements difficult. Therefore as part of a University Technology Centre based at the University of York, a significant body of research was initiated to consider how fixed priority scheduling could be migrated into the development of the FADEC software. This work needed to be performed in the context of DO-178B [57] (which was later replaced by DO-178C), and the software written in SPARK 95, which is a subset of Ada supported by verification tools [8].

5.3 Research

In the early 1990s there was significant work by a number of research groups on fixed priority preemptive scheduling, including by the Real-Time Systems Research Group at the University of York. This led to a number of approaches to both priority assignment and schedulability analysis [2] (see also the material referenced in Section 3.3). This analysis largely covered independent tasks and systems without overheads. Similar to the majority of works in this area, it did not consider how the scheduling policies used affect software development.

The first consideration, in 1995/6, was to establish a detailed understanding of why the software was currently developed the way it was and what the implications of any change would be [14]. This work was undertaken by Iain Bate, under the guidance of Alan Burns, as part of a long-term project funded by Rolls Royce., The most obvious and important conclusion of this work was that a non-preemptive approach should be used. There were three main reasons for this. Firstly, the existing software was written in a non-preemptive fashion, therefore the minimal change and the one that carried least risk was to stay with that approach. This also gave the easiest reversion path in case the Aviation Authorities, who regulate the certification of systems, rejected the final safety case. Secondly, fixed priority scheduling was already causing a significant increase in the number of paths through the software, since for the majority of tasks there was no longer a deterministic order of execution. As functional verification is much more costly than timing verification then managing its financial cost was deemed more important. Finally, the potential overheads and the complexity of the Real-Time Operating System (RTOS) was higher with a preemptive scheduler.

Given the decision to employ a non-preemptive scheduler, the lack of substantial work in the academic literature, and a need to keep the overheads as low as possible, the next piece of work looked at how the RTOS should be designed and analysed. This led to a detailed assessment of how the existing RTOS was designed, the minimal migration path possible while reusing the existing mechanisms for timing watchdogs, and how the timing overheads could be analysed. This assessment resulted in a new task release mechanism that ensured the overheads were O(1) [1]. The final technical challenge was how to take the complex timing requirements of the FADEC and map these onto a set of task attributes. The approach taken was based on the use of offsets to control the jitter within the system [11] and setting independent task deadlines such that the transactional (precedence) requirements were met [12]. The overall research and strategy [9], [13] were published in 1998.

5.4 Route to Impact

A key aspect of the work was engaging with Rolls-Royce's technical staff to understand how they develop systems and how the adoption of fixed priority scheduling would affect their work. This meant Iain Bate spending extensive periods of time within Rolls-Royce not only on fixed priority scheduling for FADECs but also gaining their trust by helping out with other immediate technical concerns [15]. As part of this strategy, a champion within the company was created who could not only guide the research but would help pull it into the organisation and then own it after the research was complete [51]. Four other important activities were undertaken. Firstly, FADECs have a need for regulatory approval and hence once some key decisions were taken a Preliminary Safety Case was established in 1997 which could then be discussed with both Rolls-Royce's engineers as well as representatives from the Civil Aviation Authority (CAA) [53]. The result of this step was a clear picture of the implications of the technology and company approval to continue the investigation. Secondly,

a cost-benefit analysis was undertaken to not only understand the financial implications for the whole engine development but also the risks. Thirdly, a RTOS was written in SPARK and a qualified tool, *VisualFPS*, produced for task attribute assignment and schedulability analysis. Finally, Rolls-Royce placed patents [10] on the work in 1997 and the University of York's legal team addressed potential litigation issues that could emerge.

After the technology was "adopted" by the project team for its first project, the technology was abandoned as the project fell behind schedule and any unnecessary risk was cut. The links with Rolls-Royce then went quiet until the next project reached a point at which a hardware re-design would be necessary without VisualFPS. This led to its adoption in 2003 and its subsequent on-going use. Notably during this time there was little contact between the University of York and Rolls-Royce due there being no reason to change the adopted approach. This demonstrates that a robust future-proofed approach had been developed.

The FADEC software including the fixed priority non-preemptive scheduler were certified to DAL-A by the CAA in December 2002 for the Tay 611-8C engine. (Note, the Honeywell Digital Engine Operating System (DEOS), which uses fixed priority preemptive scheduling is noted as being *"contracted for use in 6 FAA certified jet products"* in 2001 [20]).

5.5 Standards

As previously stated, the FADEC software is produced in accordance with DO-178C/ED-12C. This provides a set of objectives for the software development and verification process, and requires evidence to be produced by the development organisation to demonstrate compliance as part of the engine/aircraft certification activity. However, DO-178C/ED-12C is a process-based guidance document, and the certification objectives focus on compliance to requirements and conformance to standards. There is little guidance on specific product performance aspects, for example. A common myth is that the standards and the regulatory authorities demand that timing requirements are always met and static analysis is mandatory [50]. Instead current best practice, arguments and evidence for acceptable safety, and graceful degradation when the inevitable failures occur is what matters. Predictability of system-level performance is the overriding principle. Safety experts including the regulatory authorities also provide some steer towards achieving these things in the presence of new technologies through position papers by the Certification Authorities Software Team. For example, CAST 20 [26] gives guidance to those considering using processors with caches. For the FADEC software, this meant supporting the relevant parts of the safety argument through a No Less Safe Than Before approach, i.e. that the new technology did not introduce new hazardous events or make existing ones more likely or more severe. It is worth noting that a significant influence in the regulatory authority's decision was that the scheduling approach came with mathematical analysis that had been peer reviewed in top international conferences by specialists in the field.

A further complication was that the certification regime differs between Europe and the US. The European Aviation Safety Agency (EASA) tend to regulate civil aviation certification centrally, using a team of experts employed by EASA. The Federal Aviation Agency (FAA) in the USA operate a "Designated Engineering Representative" (DER) scheme, where DERs are licensed by the FAA but employed by the applicant companies. This can lead to variation in how the certification rules and compliance evidence requirements are applied.

5.6 Impact

The impact of this work was easy to gauge as millions of pounds were saved by avoiding the need to change the hardware platform and hence the attendant risk of delivering the aircraft engine late. Since then, the technology has been used within Rolls-Royce without the need

for any updates. The benefits of this are much harder to quantify. Outside the FADEC, fixed priority scheduling has now been widely adopted in avionics and other critical systems including automotive.

5.7 Beneficiaries

The beneficiaries of this successful technology transfer are many fold. External to Rolls-Royce, its questionable whether Rolls Royce adopting fixed priority scheduling made it easier for others, or whether the change of scheduling practice was inevitable, but it certainly didn't harm. Internal to Rolls-Royce, the benefits are clear. Financially it saved a significant amount of money some of which was easy to quantify, i.e. the immediate saving of not procuring a new processing platform. Further, there is the longer term benefit of an automated tool for synthesising the scheduler and producing analysis results. The change of scheduler has also given Rolls Royce's engineers the freedom to specify different timing requirements, which has allowed both better control of jitter and more flexible choices of task periods. Both of these have enabled improvements in engine performance and reductions in processor utilisation.

5.8 Future Challenges

The future challenges for Rolls Royce are significant.[6] As of 2018, they are currently undertaking their most ambitious engine re-design in more than 30 years. This re-design is targeted at dramatic improvements in engine efficiency, towards the industry wide Clean Skies initiative, while at the same time reducing costs. This has led to significant interest in mixed-criticality scheduling [22], cheaper to implement and maintain communications, and more advanced control and monitoring systems. In response, Rolls Royce has started a number of research projects including ones to derive a new scheduling and timing analysis strategy. There are a number of major challenges to tackle including:

1. Where do the values for the high-criticality WCET and low-criticality WCET estimates come from?
2. If low-criticality services can be dropped or degraded for a period of time, then how regularly and for how long?
3. How to generate the test vectors to support timing analysis?
4. How to create an equivalent No Less Safe Than Before argument?
5. How to move to preemptive, as apposed to non-preemptive, scheduling?
6. How to implement a predictable scheduler with minimal overheads complemented by appropriate timing analysis?
7. How to allocate tasks and assign task attributes so that the timing requirements are met?

6 Key Success Factors and Roadblocks

Below, we list some of the key success factors in transferring real-time systems research into industrial practice. First we consider the experience of developing the three start-up companies discussed above. With the benefit of hindsight, these were the main factors in ensuring that the companies succeeded, growing from less than 5 employees to more than 20, and culminating in successful trade sales.

[6] See the Keynote presentation at WMC (RTSS) in 2017 – `https://github.com/CPS-research-group/WMC2017/raw/master/keynote.pdf`

1. *Having an idea and then a product that made a step change for customers, providing a return on their investment.* Each commercial product provided this step change. Volcano increased network utilisation from 30% to 80% with improved reliability, and reduced development, production and warranty costs. The reduced memory footprint and overheads of the RTA-OSEK/RTA-OS operating systems resulted in lower production costs, while the use of proven real-time policies and mechanisms as well as schedulability analysis improved reliability resulting in lower warranty costs. Finally, RapiTime provided an efficient WCET analysis process, which was portable across different platforms, providing a significant reduction in testing and optimisation effort and costs.
2. *A core team of smart and hardworking people.* The founders of each company and the first few employees worked very hard (6 days per week 12+ hours per day) over many years to ensure that the company was a success.
3. *A product that was not easy to replicate: barrier to competition.* This was important in obtaining funding and getting a foothold in the market. It was particularly evident with the RTOS since the company was subsequently bought by one of its competitors.
4. *Extremely high product quality and outstanding customer support.* When a company is small and has only been around for a year or two it needs to build an excellent reputation. Quality is absolutely essential at this time, since it is make or break in terms of winning the trust of major companies who are considering adopting the technology.
5. *A balanced team of people.* On the technical side, it was not sufficient to just have technologists and software engineers who worked in the back office. Field application engineers and support staff who could do an exceptional job at customer sites / handling customer issues were also needed. Marketing and sales staff who actually understood the technology and could therefore talk effectively to both engineers and managers at customer sites were essential.
6. *Previous experience.* Having someone on board who has previous experience in a successful start-up company in the same field can be hugely advantageous, as they will understand what is needed to grow a company successfully and help avoid all manner of pitfalls.
7. *Attracting an acquisition.* An acquisition can lead to scaling up of the success of the technical transfer. In all cases the speed of adoption accelerated after acquisition. Therefore structuring the company not only for standalone success, but also acquisition was a common success factor.

There were also a number of major roadblocks and difficulties in turning promising research results into commercial reality.

1. *Funding the initial development from academic ideas and prototypes to saleable product.* A high quality industry ready product is very different from academic prototypes. It needs to be robust, with full error handling; easy to use, (since users will typically not be experts) and supported by full documentation including internal documents, e.g. requirement and test specifications, as well as external documentation such as user guides, tutorials, and marketing material. It also needs to be of extremely high quality; fully tested against its specification, and as far as possible the code needs to be bug free. The difficulty arises because considerable effort is needed in this area when the company first starts and has few sales. This effort has to be funded somehow. Self-funding by the founders can be effective if they can afford not to be paid for a while, or they can get one or two early contracts from a benevolent customer. Business angel or venture capital funding is also effective but comes at a cost of giving up some proportion of the equity (shares) in the company. Assistance from the host University or institution in terms of providing time to cover initial development efforts is also greatly beneficial at this stage.

2. *Adapting academic research to cater for industrial realities.* It is rarely if ever the case that academic theories and prototypes cover all of the details that need to be catered for in real industrial systems. There are inevitably different behaviours, aspects that are left out of models, and extra functionality that is required in commercial tools. The analysis of CAN used in the Volcano technology came close to a direct transfer. Even so, it required that the Volcano Target Package was carefully designed and developed to meet the assumptions of the theory, which was itself extended to account for specific implementation behaviours (e.g. polling input and output). Substantial engineering work was also needed to support key commercial requirements, such as the ability to re-configure signal packing and message IDs post-production. Each of the start-ups described in this paper undertook substantial engineering efforts as well as further adaptation and extension to academic results to produce commercially viable products. Again the difficulty arises because much of this effort is needed at a time when the company first starts and may have little funding and few staff.
3. *Finding the right sales staff.* In each of the three start-up companies discussed in the previous sections, it proved remarkably difficult to find people who were both good at sales and really understood the technology. In each company, sales were led by someone with a strong technical background who had the right personality and turned themselves into an excellent salesman via appropriate training. Bringing in "high flying" sales staff without a strong technology background was an expensive mistake. Beware that sales staff can be very good at selling themselves!
4. *Convincing major companies to adopt a new technology.* This is problematic due to the conservative approach often taken to purchasing from small companies. Major companies rightly have the following concerns: (i) Will the start-up be around in a year's time? (ii) Can it handle the volume of support that may be needed? (iii) Is the product really of a high enough quality to rely upon for future production? The main factors in addressing these questions were product and customer service quality, and simply time; it becomes easier to make larger sales once a company has been established for a few years.

For the final case study, where the technology transfer was into the company that directly benefited from its adoption, the existence of a long term link between the company and the research group was crucial. Simple things such as developing a common vocabulary, terms, and concepts take time. Also important are champions in both the industrial partner and the academic group. The simplistic notion that 'industry has the problem, and academia the solution' is far from true. Academics have a crucial role in understanding the problem, and experienced engineers are essential in shaping the solution. For example, moving to fixed priority scheduling meant that engineers now had greater flexibility in setting the timing requirements of the system, e.g. not just being restricted to a harmonic of the minor cycle rate. This raised the question of what the real timing requirements were. As part of the technology transfer the academics worked with engineers from Rolls-Royce across multiple disciplines, (e.g. software, hardware and control systems) to establish what the limits of the timing requirements were. This allowed significant extra benefits to be gained. Within the company there must be pull, and of course within academia, push. Research benefits from extensive use of abstraction to get to the core of the issues being addressed; however, to deploy this research the *devil is in the detail*, which takes both time and commitment.

Once a new technology, analysis method or design approach is adopted then it must be transferred completely. The academic cannot be part of the day-to-day application of the new ideas. A successful partnership has periods of deep interaction and periods of separation. New challenges may lead to a rekindled partnership.

7 Conclusions

In this paper, we described how real-time systems research has been successfully transferred into industrial practice via three start-up companies, and by direct application. Each of these impact case studies represents a success story for the real-time community. Each start-up company has developed commercially viable products, which are in use today by many of the world's leading automotive and aerospace companies and their tier 1 suppliers.

Volcano technology is used for in-vehicle communication between Electronic Control Units (ECUs) in millions of cars produced by Volvo since 1998, as well as in vehicles from a number of other major automotive manufacturers in Europe, America and Asia. The RTA-OS and RTA-OSEK real-time operating systems are used by the overwhelming majority of the world's car makers; with the number of deployed units exceeding 1.25 billion in 2017, and continuing to increase by 50-100 million per year (i.e. 1-2 million per week!). RapiTime is in use on a wide range of aerospace projects where customers need to understand the detailed execution time behaviour of their systems. The company, Rapita Systems, has recently undergone a successful trade sale (April 2016) and continues to employ a large number of graduate and post-graduate staff with expertise in real-time systems gained in the Real-Time System Research Group at the University of York. Rolls Royces' use of Visual FPS continues and has demonstrated that scheduling ideas from the research community can be exploited in the most safety-critical application domain.

Other real-time systems research groups have also succeeded in transferring their research into commercial products via start-up companies, examples include: Symptavision Gmbh (acquired by Luxoft in 2016) and Absint Gmbh, while others are just beginning.

It takes some excellent research and ideas, a willingness to take a risk and start a company or commit to a long term relationship with an industrial partner, a great deal of hard work and persistence, and perhaps an element of luck to succeed in transferring research into world-class commercial products and systems. We hope that these impact case studies will inspire others in the community to take this entrepreneurial step.

References

1 Neil C. Audsley, Iain J. Bate, and Alan Burns. Putting fixed priority scheduling into engineering practice for safety critical applications. In *Real-Time Technology and Applications Symposium (RTAS)*, pages 2–10, 1996.

2 Neil C. Audsley, Alan Burns, Mike M. Richardson, Ken Tindell, and Andy J. Wellings. Applying new scheduling theory to static priority pre-emptive scheduling. *Software Engineering Journal*, 8(5):284–292, 1993.

3 Neil C. Audsley, Alan Burns, and Andy J. Wellings. Deadline monotonic scheduling: Theory and application. *Control Engineering Practice*, 1(1):71–78, 1993.

4 Neil C. Audsley, Ken Tindell, and Alan Burns. The end of the line for static cyclic scheduling? In *5th Euromicro Workshop on Real-Time Systems*, pages 36–41, 1993.

5 AUTOSAR. Specification of operating system v3.1.1. Technical report, AUTOSAR, 2009. URL: `https://www.autosar.org/fileadmin/user_upload/standards/classic/3-0/AUTOSAR_SWS_OS.pdf`.

6 AUTOSAR. Premium partners. `https://www.autosar.org/about/current-partners/premium-partners/`, 2018. Accessed: 2018-02-27.

7 Theodore P. Baker. Stack-based scheduling of realtime processes. *Real-Time Systems*, 3(1):67–99, 1991.

8 John Barnes. *High Integrity Ada: The SPARK Approach*. Addison-Wesley, 1997.

9 Iain J. Bate. *Scheduling and Timing Analysis for Safety-Critical Systems*. PhD thesis, Department of Computer Science, University of York, 1998.

10 Iain J. Bate and Alan Burns. Flexible scheduling for engine controllers – uk patent application number 9710522.5 and us patent number 6,151,538, 1997.

11 Iain J. Bate and Alan Burns. Timing analysis of fixed priority real-time systems with offsets. In *9th Euromicro Workshop on Real-Time Systems*, pages 153–160, 1997.

12 Iain J. Bate and Alan Burns. An approach to task attribute assignment for uniprocessor systems. In *11th Euromicro Conference on Real-Time Systems*, pages 46–53, 1999.

13 Iain J. Bate and Alan Burns. An integrated approach to scheduling in safety-critical embedded control systems. *Real-Time Systems Journal*, 25(1):5–37, 2003.

14 Iain J. Bate, Alan Burns, John A. McDermid, and Andrew J. Vickers. Towards a fixed priority scheduler for an aircraft application. In *8th Euromicro Workshop on Real-Time Systems*, pages 34–39, 1996.

15 I.J. Bate, A. Burns, T.O. Jackson, T.P. Kelly, W. Lam, P. Tongue, J.A. McDermid, A.L. Powell, J.E. Smith, A. J. Vickers, A. J. Wellings, and B.R. Whittle. Technology transfer: An integrated 'culture-friendly' approach. In *Briefing Document Technology Transfer Workshop*, 1996.

16 Guillem Bernat, Alan Burns, and Martin Newby. Probabilistic timing analysis: An approach using copulas. *J. Embedded Computing*, 1(2):179–194, 2005.

17 Guillem Bernat, Antoine Colin, and Stefan M. Petters. pWCET, a Tool for Probabilistic WCET Analysis of Real-Time Systems. In *3rd International Workshop on Worst-Case Execution Time Analysis*, pages 21–38, 2003.

18 Guillem Bernat, Antoine Colin, and Steffan M. Petters. Wcet analysis of probabilistic hard real-time systems. In *23rd IEEE Real-Time Systems Symposium*, pages 279–288, 2002.

19 Guillem Bernat, Robert I. Davis, Nicholas Merriam, John Tuffen, A. Gardner, Michael Bennett, and D. Armstrong. Identifying opportunities for worst-case execution time reduction in an avionics system. *Ada User Journal*, 28(3):189–194, 9 2007.

20 Pam Binns. A robust high-performance time partitioning algorithm: the digital engine operating system (DEOS) approach. In *20th Digital Avionics Systems Conference (DASC)*, volume 1, pages 1B6/1–1B6/12 vol.1, Oct 2001. doi:10.1109/DASC.2001.963309.

21 Bosch. Can specification version 2.0. Technical report, Robert Bosch GmbH, Postfach 30 02 40, D-70442 Stuttgart, 1991.

22 Alan Burns and Robert I. Davis. A survey of research into mixed criticality systems. *ACM Computer Surveys*, 50(6):1–37, 2017.

23 Alan Burns, N. Hayes, and M.F. Richardson. Generating feasible cyclic schedules. *Control Engineering Practice*, 3(2):151–162, 1995.

24 Alan Burns and Andy J. Wellings. Engineering a hard real-time system: From theory to practice. *Softw., Pract. Exper.*, 25(7):705–726, 1995. doi:10.1002/spe.4380250702.

25 Lennart Casparsson, Antal Rajnak, Ken Tindell, and Peter Malmberg. Volcano - a revolution in on-board communications. Technical report, Volvo, 1998.

26 Certification Authorities Software Team CAST. Addressing cache in airborne systems and equipment – cast-20, June 2003.

27 Antoine Colin and Guillem Bernat. Scope-tree: A program representation for symbolic worst-case execution time analysis. In *14th Euromicro Conference on Real-Time Systems (ECRTS)*, 2002.

28 Antoine Colin and Stefan M. Petters. Experimental evaluation of code properties for WCET analysis. In *24th IEEE Real-Time Systems Symposium (RTSS)*, pages 190–199, 2003.

29 Robert Davis, Nick Merriam, and Nigel Tracey. How embedded applications using an RTOS can stay within on-chip memory limits. In *Work in Progress and Industrial Experience Sessions, 12th EuroMicro Conference on Real-Time Systems.*, 2000.

30 Robert I. Davis. Impact case study: Guaranteeing the real-time performance of in-vehicle networks. Technical report, University of York, 2015. URL: https://www-users.cs.york.ac.uk/~robdavis/papers/ImpactCaseStudyVolcano.pdf.

31 Robert I. Davis, Guillem Bernat, Ian Broster, and Antoine Colin. Impact case study: How long does your real-time software take to run? Technical report, University of York, 2015. URL: https://www-users.cs.york.ac.uk/~robdavis/papers/ImpactCaseStudyRapiTime.pdf.

32 Robert I. Davis and Alan Burns. Robust priority assignment for messages on controller area network (CAN). *Real-Time Systems*, 41(2):152–180, 2009.

33 Robert I. Davis, Alan Burns, Reinder J. Bril, and Johan J. Lukkien. Controller area network (CAN) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems*, 35(3):239–272, 2007.

34 Robert I. Davis, Alan Burns, Victor Pollex, and Frank Slomka. On priority assignment for controller area network when some message identifiers are fixed. In *23rd International Conference on Real Time Networks and Systems, RTNS*, pages 279–288, 2015.

35 Robert I. Davis, Steffen Kollmann, Victor Pollex, and Frank Slomka. Controller area network (CAN) schedulability analysis with FIFO queues. In *23rd Euromicro Conference on Real-Time Systems (ECRTS)*, pages 45–56, 2011.

36 Robert I. Davis, Steffen Kollmann, Victor Pollex, and Frank Slomka. Schedulability analysis for controller area network (CAN) with FIFO queues priority queues and gateways. *Real-Time Systems*, 49(1):73–116, 2013.

37 Robert I. Davis and Nicolas Navet. Controller area network (CAN) schedulability analysis for messages with arbitrary deadlines in FIFO and work-conserving queues. In *9th IEEE International Workshop on Factory Communication Systems, (WFCS)*, pages 33–42, 2012.

38 Robert I. Davis and Nigel Tracey. Impact case study: The world's smallest automotive real-time operating system. Technical report, University of York, 2015. URL: https://www-users.cs.york.ac.uk/~robdavis/papers/ImpactCaseStudyRTOS.pdf.

39 Paul Emberson and Iain J. Bate. Minimising task migrations and priority changes in mode transitions. In *13th IEEE Real-Time And Embedded Technology and Applications Symposium*, pages 158–167, 2007.

40 Paul Emberson and Iain J. Bate. Stressing search with scenarios for flexible solutions to real-time task allocation problems. *IEEE Transactions on Software Engineering*, 36(5):704–718, 2010.

41 ETAS. RTA-OS RH850/WR Port Guide. https://www.etas.com/download-center-files/products_RTA_Software_Products/RTA-OS_RH850WR_Port_Guide_V2.0.5.pdf, 2017. Accessed: 2018-02-27.

42 ETAS. RTA software products. https://www.etas.com/en/products/rta_software_products.php, 2018. Accessed: 2018-02-27.

43 Mentor Graphics. Volcano in-vehicle embedded software. http://www.mentor.com/products/vnd/in-vehicle_software/. Accessed: 2018-02-27.

44 Mentor Graphics. Volcano Network Architect (vna). http://www.mentor.com/products/vnd/communication-management/vna/. Accessed: 2018-02-27.

45 Mentor Graphics. Mentor graphics strengthens its automotive solutions portfolio with the acquisition of volcano communications technologies. https://www.mentor.com/company/news/volcano_acquisition, 2005. Accessed: 2018-02-27.

46 Mentor Graphics. Shanghai automotive industries adopts mentor graphics volcano automotive network design tools. http://www.mentor.com/products/vnd/news/saic_sdopts_volcano, 2006. Accessed: 2018-02-27.

47 Mentor Graphics. Volcano target package datasheet. `http://www.mentor.com/products/vnd/communication-management/vna/upload/VNA_Datasheet.pdf`, 2006. Accessed: 2018-02-27.

48 Mentor Graphics. Volcano target package datasheet. `http://www.mentor.com/products/vnd/in-vehicle_software/volcano_target_package/upload/vtp-ds.pdf`, 2010. Accessed: 2018-02-27.

49 Mentor Graphics. Volcano network architect from mentor graphics verifies and improves network bandwidth usage at mazda. `http://www.mentor.com/products/vnd/news/mentor-vnd-mazda`, 2012. Accessed: 2018-02-27.

50 Paul Graydon and Iain J. Bate. Realistic safety cases for the timing of systems. *The Computer Journal*, 57(5):759–774, 2014.

51 S. Hutchesson and N. Hayes. Technology transfer and certification issues in safety critical real-time systems. In *Digest of the IEE Colloquium on Real-Time Systems*, page 98/306, 1998.

52 ISO. ISO 17356-3:2005 preview road vehicles – open interface for embedded automotive applications – part 3: Osek/vdx operating system (os). Technical report, ISO, 2005. URL: `https://www.iso.org/standard/40079.html`.

53 Tim Kelly, Iain J. Bate, John McDermid, and Alan Burns. Building a preliminary safety case: An example from aerospace. In *Australian Workshop on Industrial Experience with Safety Critical Systems and Software*, 1997.

54 Dawood Ashraf Khan, Robert I. Davis, and Nicolas Navet. Schedulability analysis of CAN with non-abortable transmission requests. In *16th IEEE Conference on Emerging Technologies & Factory Automation, (ETFA)*, pages 1–8, 2011.

55 Motorola. MSCAN block guide. Technical report, Motorola, , Document No. S12MSCANV2/D., 2004. URL: `http://application-notes.digchip.com/314/314-67565.pdf/`.

56 Antal Rajnak. Volcano technology: Enabling correctness by design. In *The Industrial Communication Technology Handbook*, chapter 32. CRC Press, 2009.

57 RTCA-EUROCAE. *Software Considerations in Airborne Systems and Equipment Certification DO-178B/ED-12B*. RTCA, Inc, December 1992.

58 Rapita Systems. Capturing worst case timing and stack usage data for do-178b level a embraer flight control systems. `https://www.rapitasystems.com/downloads/do-178b-level-embraer-fcs`. Accessed: 2018-02-27.

59 Rapita Systems. Verifying the timing correctness of infineon's safetcore safety drivers. `https://www.rapitasystems.com/downloads/infineon-safetcore-drivers`. Accessed: 2018-02-27.

60 Rapita Systems. Rapitime worst-case execution time optimization on the bae systems hawk mission computer. `https://www.rapitasystems.com/downloads/bae-systems-hawk-mission-computer`, 2006. Accessed: 2018-02-27.

61 Rapita Systems. Flight control system execution timing analyzed cheaper, faster with rapitime. `https://www.rapitasystems.com/downloads/wide-body-jet-flight-control-system`, 2009. Accessed: 2018-02-27.

62 Rapita Systems. Proving and Improving Worst-Case Execution Times on the Alenia Aermacchi M-346. `https://www.rapitasystems.com/downloads/alenia-aermacchi-m-346`, 2010. Accessed: 2018-02-27.

63 Rapita Systems. Qualification of RapiCover for MC/DC coverage of DO-178B level-A software. `https://www.rapitasystems.com/system/files/downloads/mc-cs-009_alenia_aermacchi_m346_case_study_v2.pdf`, 2010. Accessed: 2018-03-26.

64 Rapita Systems. Danlaw acquires rapita systems. `https://www.rapitasystems.com/news/danlaw-acquires-rapita-systems`, 2016. Accessed: 2018-02-27.

65 Ken Tindell and Alan Burns. Guaranteeing message latencies on controller area network (CAN). In *1st international CAN conference*, pages 1–11, 1994.

66 Ken Tindell, Alan Burns, and Andy J. Wellings. An extendible approach for analyzing fixed priority hard real-time tasks. *Real-Time Systems*, 6(2):133–151, 1994.

67 Ken Tindell, Alan Burns, and Andy J. Wellings. Analysis of hard real-time communications. *Real-Time Systems*, 9(2):147–171, 1995.

68 Ken Tindell, Alan Burns, and Andy J. Wellings. Calculating controller area network (CAN) message response times. *Control Engineering Practice*, 3(8):1163–1169, 1995.

69 Ken Tindell, H. Hanssmon, and Andy J. Wellings. Analysing real-time communications: Controller area network (CAN). In *15th IEEE Real-Time Systems Symposium*, pages 259–263, 1994.

70 Volvo. Volvo annual reports. `https://group.volvocars.com/sustainability/publication-list`, 2018. Accessed: 2018-02-27.

71 Reinhard Wilhelm, Jakob Engblom, Andreas Ermedahl, Niklas Holsti, Stephan Thesing, David Whalley, Guillem Bernat, Christian Ferdinand, Reinhold Heckmann, Tulika Mitra, Frank Mueller, Isabelle Puaut, Peter Puschner, Jan Staschulat, and Per Stenström. The worst-case execution-time problem; overview of methods and survey of tools. *ACM Trans. Embed. Comput. Syst.*, 7(3):36:1–36:53, 2008. `doi:10.1145/1347375.1347389`.

72 Yaba. Rapita systems flies high. `http://www.rapitasystems.com/system/files/yabawinter05news.2.pdf`, 2005. Accessed: 2018-02-27.

73 Patrick Meumeu Yomsi, Dominique Bertrand, Nicolas Navet, and Robert I. Davis. Controller area network (CAN): response time analysis with offsets. In *9th IEEE International Workshop on Factory Communication Systems, (WFCS)*, pages 43–52, 2012.