| | | |
|---|---|---|
| **Institution: Queen Mary University of London** | | |
| **Unit of Assessment: 11** | | |
| **Title of case study: The Infer tool for automatic verification of memory violations in software systems** | | |
| **Period when the underpinning research was undertaken: 2006-31 Dec 2020** | | |
| **Details of staff conducting the underpinning research from the submitting unit:** | | |
| **Name(s):** | **Role(s) (e.g. job title):** | **Period(s) employed by submitting HEI:** |
| Dino Distefano | Professor | 2006-present |
| **Period when the claimed impact occurred: 2014 – 31 Jul 2020** | | |
| **Is this case study continued from a case study submitted in 2014? Y** http://impact.ref.ac.uk/CaseStudies/CaseStudy.aspx?Id=18590 | | |

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

Researchers at Queen Mary produced breakthrough algorithms for detecting computational memory violations. Using these algorithms, they developed the program analysis tool Infer, initially commercialised by the Queen Mary spin-off Monoidics, and since 2013, when Monoidics was acquired, by Facebook. Infer enables highly-scalable automatic verification of memory violations, which are a major cause of security breaches and operational flaws in software systems. Infer is open source and used by companies (including Amazon, Spotify, Uber, Mozilla) and developers around the world.

At Facebook, Infer analyses every change of infrastructure and mobile code. Every month, thousands of bugs are fixed before Facebook's apps (ie Facebook, Messenger, Instagram, WhatsApp, Oculus) are shipped to its 1,600,000,000 users. In the second half of 2017 alone, more than 25,000 bugs reported by Infer were fixed by Facebook developers saving tens of thousands of hours.

**2. Underpinning research** (indicative maximum 500 words)

Verifying that computer programs correctly and safely manipulate computer memory is imperative, as memory errors can cause operational flaws and have fatal consequences for the missions of systems (eg safety of a plane in flight). However, previous program verifiers had only been tested in toy programs of tens or hundreds of lines of code. Large systems need a reliable verifier for hundreds of thousands or even millions of lines of code. The program verifier, Infer, enables this level of verification.

Infer is based on research on logics for reasoning about memory use and a type of algorithm that gives partial decision procedures. Two areas of research were brought together to develop 'separation logic,' the first effective theory for reasoning about computer memory. Pym and O'Hearn at Queen Mary (both now at UCL) brought their research on substructural logic together with joint work (between researchers at Queen Mary and Carnegie Mellon) on a type of data and data structure – primitives to describe heap structure – to develop the theory of separation logic. Based on these theories, Distefano, at Queen Mary, produced the Space Invader tool [3.1], the first program analyser based on separation logic. Space Invader opened up, for the first time, the possibility of automating the theoretical ideas of separation logic.

To move on from the Space Invader tool to automatic proofs that can be used in real-world industrial code, several steps were necessary. The first milestone was reached when Distefano, in collaboration with Calcagno at Imperial College and other researchers at Microsoft, developed a technique where an analyser can tune the precision of its analysis on-the-fly to different data structures of the analysed program [3.2].

The second milestone was the development of automatic proofs concerning the use of pointers. Pointers are variables whose value is a location in the computer's memory. They are one of the major sources of error in memory manipulation in entire industrial programs (Linux and Microsoft device drivers of up to 10,000 lines of code) [3.3]. After this development, Distefano and researchers at Microsoft introduced a fully automated verification approach for object-oriented languages using separation logic [3.4].

Finally, Distefano proposed the notion of bi-abductive inference (a form of logical inference that automates the key ideas about local reasoning), a breakthrough that allowed partial properties of entire open-source projects in the hundreds of thousands or millions of lines of code to be shown [3.5]. To put these results in context, just a few years before, proofs of memory safety had been done only for toy programs in the tens or sometimes hundreds of lines of code.

Infer is based on these foundations and was initially commercialised by the Queen Mary spin-off, Monoidics Ltd, from 2009. Monoidics' Infer product attracted customers from across the world including government agencies and industrial clients such as Airbus, Mitsubishi, Toyota, Lawrence Livermore National Laboratory and ARM Holdings.

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

[3.1] Distefano. D., O'Hearn, P. W. & Yang, H. (2006). A Local Shape Analysis Based on Separation Logic. *TACAS* 287-302. https://doi.org/10.1007/11691372_19

[3.2] Berdine, J., Calcagno, C., Cook, B., Distefano, D., O'Hearn, P. W., Wies, T. & Yang, H. (2007). Shape Analysis for Composite Data Structures. *CAV*, 178-192. doi:10.1007/978-3-540-73368-3_22

[3.3] Yang, H., Lee, O., Berdine, J., Calcagno, C., Cook, B., Distefano, D. & O'Hearn, P. W. (2008). Scalable Shape Analysis for Systems Code. *CAV, 6.* 385-398. https://www.microsoft.com/en-us/research/publication/scalable-shape-analysis-for-systems-code/

[3.4] Distefano, D. & Parkinson, M. J. (2008). jStar: Towards practical verification of Java. *ACM Sigplan Notices*, *43* (10), 213-226. https://doi.org/10.1145/1449955.1449782

[3.5] Calcagno, C., Distefano, D., O'Hearn, P. W. & Yang, H. (2011). Compositional Shape Analysis by means of Bi-abduction. *Journal of the ACM*, *58* (10). https://doi.org/10.1145/2049697.2049700

**Evidence of quality of the research:**

[EQR. 1] Distefano, D. (2018). Most Influential OOPSLA Paper Award. *SIGPLAN*. The papers are judged by their influence over the past decade. The awarded paper is [3.5]. https://www.sigplan.org/Awards/OOPSLA/

[EQR. 2] Distefano, D. (2019). Most Influential POPL Paper Award. *SIGPLAN*. The papers are judged by their influence over the past decade. The awarded paper is the conference version of [3.4]. https://www.sigplan.org/Awards/POPL/#:~:text=Presented%20annually%20to%20the%20author,influence%20over%20the%20past%20decade.

[EQR. 3] Distefano, D. (2014). Silver Medal. *Royal Academy of Engineering.* For the commercialisation of Infer 'for an outstanding personal contribution to UK engineering by an early to mid-career engineer resulting in market exploitation'. https://www.raeng.org.uk/news/news-releases/2014/april/silver-medals-for-uks-up-and-coming-entrepreneurs

[EQR. 4] Distefano, D. (2007-2012). [DCSF1A3R] *Royal Academy of Engineering*. Research Fellowship. GBP448,963.

[EQR. 5] Distefano, D. (2009 to 2012). Adaptive Heap Analysis [EP/G006245/1]. *EPSRC*. Research Grant. GBP253,000.

[EQR. 6] Distefano, D. (2010 to 2013). jStar: making java verification practical [EP/H011749/1]. *EPSRC*. Research Grant. GBP219,000.

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

Infer detects memory violations in computer programs with hundreds of thousands or millions of lines of code. Before its development, proofs of memory safety had been done only for programs in the tens or sometimes hundreds of lines of code.

**Facebook acquisition of Monoidics**

Monoidics is a Queen Mary spin out company co-founded, in 2009, by Distefano, which grew to include offices in the UK, USA and Japan. Monoidics marketed Infer, an automatic program verification tool built on Distefano's proofs [3.1-3.4]. In July 2013, Monoidics was acquired by Facebook to improve the quality and the security of their code base.

Since the acquisition, the development of Infer has been carried out at Facebook, where Infer is now an established technology for software quality. The Facebook code is an evolving system, updated frequently and concurrently by many developers. It is not unusual for more than a thousand modifications to the mobile code to be submitted for review in a given day. Infer reports quickly on these code modifications, in the region of 10 minutes, to fit in with developers' workflow. Coping with this scale and velocity required the advanced mathematical techniques of separation logic and bi-abduction provided by Distefano's research. Bryan O'Sullivan, director of Developer Efficiency at Facebook, said [5.1]: "Today within Facebook, the verification technology originally designed by Prof. Dino Distefano and colleagues contributes to the quality of the products we serve to over 2,000,000,000 people every day. Currently Infer analyses every applicable code change within Facebook and its family of apps (Messenger, WhatsApp, Instagram, Oculus). Every month, thousands of issues are fixed before these apps are shipped to users. Thus, Infer has saved Facebook engineers innumerable hours that they would have had to spend debugging the problems it detected, had those problems reached production." In the second half of 2017 alone, more than 25,000 bugs reported by Infer were fixed by Facebook developers.

**Use of Infer as an Open Source tool**

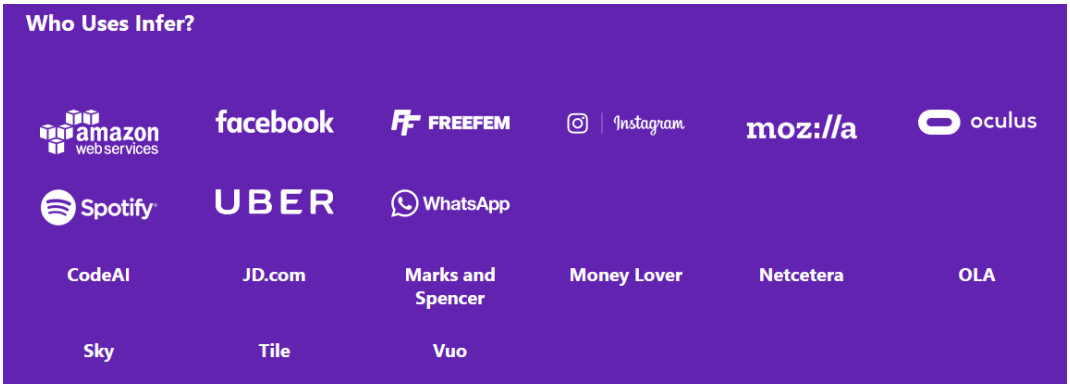


**Figure 1: Screen grab from fbinfer.com, showing some infer users. Copyright [2020] by fbinfer.com.**

Infer was open sourced in 2015 [5.2]. Since then, it has been used by many companies and developers around the world (including Amazon Web Services [5.3], Uber, Spotify, Mozilla, Sky, Mark & Spencer) [5.4]. Infer is used in diverse ways by these companies:

*Facebook and Spotify* use it for user generated content apps, and thus the development of their software as it is continuously released to users. Deniz Türkoglu, Software Engineer at Spotify in 2016 said [5.5], "At Spotify we are continuously working on making our codebase better, and in the Android infrastructure team we use a lot of tools: static analyzers, linters, thread/address sanitizers, etc. In our quest to make our code even better, we started using Infer. Infer found several legitimate issues that other tools had missed. Infer is a great add-on to a company's toolbox. It's not intrusive — you can simply add it to your flow and it will tell you where you forgot to close that cursor or leaked that context."

*Amazon and Uber* use Infer for their online marketplaces. Manu Sridharan, software engineer at Uber in 2016 said [5.6], "In 2016, Uber deployed the Facebook Infer tool for static detection of potential NullPointerExceptions (NPEs). Alongside our Runtime Annotation Validation Engine (RAVE) validation, the tool reduced the number of NPEs observed in our apps in production by an order of magnitude."

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

[5.1] O'Sullivan, B. Director of Engineering. *Facebook (*testimonial letter, 27 March, 2019). [Corroborator 1]

[5.2] O'Hearn, P. W., Distefano, D. & Calcagno, C. (2015, 11 June). *Open-sourcing Facebook Infer: Identify bugs before you ship*. Facebook Engineering. https://code.facebook.com/posts/1648953042007882/open-sourcing-facebook-infer-identify-bugs-before-you-ship/. 26 October 2020.

[5.3] Cook, B. (2018). *Computer aided verification*. H. Chockler and G. Weissenbacher (Eds.) *CAV, LNCS (10981)*, 38–47. https://doi.org/10.1007/978-3-319-96145-3_3.

[5.4] *A tool to detect bugs in Java and C/C++/Objective-C code before it ships*. Facebook Infer. https://fbinfer.com/. 26 October 2020.

[5.5] Villard, J. (2016, 17 March). *Collaboration with Spotify*. Facebook Infer. https://fbinfer.com/blog/2016/03/17/collaboration-with-spotify/. 20 March 2020.

[5.6] Sridharan, M. (2017, 19 October). *Engineering NullAway, Uber's Open Source Tool for Detecting NullPointerExceptions on Android.* Uber Engineering. https://eng.uber.com/nullaway/. 26 October 2020.