Recent results with Correctness by Construction and SPARK

Rod Chapman
Praxis High Integrity Systems

Goals

- Don’t say the same as last year (or the year before…)
- Present some new techie stuff.
- Make sure everyone gets home on time.

Contents

- What are C-by-C and SPARK?
- Projects and Results
- SPARK technical update…
- C-by-C and the SEI PSP/TSP
- The Correctness-by-Construction Challenge

What is Correctness by Construction

- A systems and software development approach.
- Key principles:
  - Make it hard to introduce defects in the first place.
  - Detect and correct defects as soon as possible after their introduction

- Easy huh? Easier said than done…

Correctness by Construction(2)

- Observation
  - We can't rely on testing alone as the primary verification activity - much too expensive and risk prone.
  - Also, for the most critical systems, testing can **never** generate sufficient evidence.
- So what else can we do?

Correctness by Construction(3)

- Therefore, C-by-C is a design approach characterized by:
  - Use of static verification to **prevent** defects at all stages.
  - Small, verifiable design steps.
  - Appropriate use of formality (aka “Maths”).
  - “Right tools and notations for the job” approach.
  - Generation of certification/evaluation evidence as a side-effect of the development process. E.g. for a safety-case.

So what's SPARK?

- SPARK embodies the principles of C-by-C in a programming language and verification system.

- Languages *really do* matter.
  - They affect the way we think about the world, the problem we're solving etc. etc.

The Catch…

- Our ability to perform static verification critically depends on the language or notation under analysis.
- In particular, **ambiguity** in the definition of the language severely limits what is achievable.
- Ideally, languages and notations should be as unambiguous as possible.

SPARK vs X, Y, or Z…

- SPARK is **not** about the retrospective, “bug finding” of existing code.
- It aims for a sound, practical, efficient, constructive verification environment that is appropriate for the highest assurance levels.
  - SPARK **does** go for the “last 10 percent”
- We **can** make the “hard problems” go away (or at least render them easy) through careful control and annotation of the language subset.

Contents

- What are C-by-C and SPARK?
- Projects and Results
- SPARK technical update…
- C-by-C and the SEI PSP/TSP
- The Correctness-by-Construction Challenge

C-by-C Projects

Project	Year	Size (loc)	Productivity (loc/day)	Defects (per kloc)
CDIS	1992	197000	12.77	0.75
SHOLIS	1997	27000	7.0	0.22
MULTOS CA	1999	100000	28.0	0.04
SAWCS	2001	39000	11.0	0.05
Tokeneer	2003	10000	38.0	0.0

More recent C-by-C Projects

- Tokeneer 2
  - Summer 2004 - Interns learnt C-by-C and SPARK.
  - Spring 2005 - Making deliverables fully CC compliant.
- Jet Engine Health Monitoring
- ARINC ACAMS (Annapolis, MD)
  - Aircraft health monitoring

Results…

- C-by-C productivity and defect rate is as good or better than data reported for TSP, CMM 5, CleanRoom.
- C-by-C provides the means to also meet the most stringent regulatory requirements and standards *without* undue additional pain and/or expense.
- We find that *better can be cheaper* - ultra-reliable does *not* mean ultra-expensive!

Contents

- What are C-by-C and SPARK?
- Projects and Results
- SPARK technical update…
- C-by-C and the SEI PSP/TSP
- The Correctness-by-Construction Challenge

SPARK Technical Update

- Many, Many things…far too many to mention in detail.
- Main focus: proof, proof, proof
  - Improving the completeness of the theorem prover.
  - Improving the VC Generator to produce more “easily provable” VCs.
  - Performance and parallel proof.
- This year: usability, usability, usability.

Theorem Proving Performance

Test Data: SAWCS Project. 39kloc. 18655 VCs.

Toolset	Hardware	Time/mins	Hit rate %	VCs left
6.3 (Dec 2002)	1.8GHz P4 Mobile	111	94.5	1025
7.0	1.8GHz P4 Mobile	109	94.69	990
7.0	2.4 GHz P4 Xeon	73	94.69	990
7.1	2 * 2.4 GHz P4 Xeon	49	95.75	791
7.2 (Jan 2005)	2 * 2.4 GHz P4 Xeon	82	97.24	515

Contents

- What are C-by-C and SPARK?
- Projects and Results
- SPARK technical update…
- C-by-C and the SEI PSP/TSP
- The Correctness-by-Construction Challenge

SEI, PSP and SPARK…

- SEI have discovered Correctness-by-Construction and SPARK…
- Praxis have discovered PSP/TSP.
- SPARK projects can deliver <= 0.1 defects per kloc.
- So can PSP/TSP projects…
- What happens if you put the two together?

SEI, PSP and SPARK…

- PSP emphasis on personal practice.
  - Measurement of performance
  - Statistical analysis of data
  - Use of data to aim future planning.
  - BUT – technology neutral…
- C by C takes a strong technical stance
  - Well-defined languages
  - Strong static verification

Contents

- What are C-by-C and SPARK?
- Projects and Results
- SPARK technical update…
- C-by-C and the SEI PSP/TSP
- The Correctness-by-Construction Challenge

The Correctness-by-Construction Challenge

- For regulators:
  - Regulate!

The Correctness-by-Construction Challenge

- For procurers:
  - Your next high-integrity software procurement…
    - Demand better than 0.1 defects per kloc, **and**
    - Pay less than $100 per line, **and**
    - Get a warranty
  - If it doesn't work, send it back!

The Correctness-by-Construction Challenge

- For contractors:
  - Have the nerve to bid such a project.
  - Bid it **cheaper** than a traditional software process
    - you might just win!
  - Deliver the above.

Final Quote

"There is still no silver bullet, but dramatic improvements in software quality can be achieved through the rigorous and systematic application of *what we already know…*"

Martyn Thomas

Professor of Software Engineering, Oxford University