Alan Wassyng

Potential Collaboration between the Software Certification Consortium and the High Confidence Software and Systems Coordinating Group

McSCert

HCSS Jan 5, 2011

On behalf of SCC – thanks for this opportunity!

SCC History - 1

- SCC founded in 2007
  - Mark Lawford, Tom Maibaum, Alan Wassyng (McMaster)
  - Brian Larson (Boston Scientific)
  - Jo Atlee (Waterloo), Marsha Chechik (Toronto), Jonathan Ostroff (York)
- Steering Committee
  - Rick Chapman, Paul Jones (FDA)
  - John Hatcliff (Kansas State), Insup Lee (Pennsylvania)
  - Brian Larson (Multitude Corporation), Bran Selic (Malina Software)
  - Mark Lawford, Tom Maibaum, Alan Wassyng (McMaster)
  - Sushil Birla (NRC) should be included – we have been waiting for the Charter to be approved before changing the structure

SCC History - 2

- The idea
  - A group of researchers/practitioners from industry, regulatory/ government agencies and academia, getting together informally to see how they can improve the dependability and certification of systems that depend on software
  - Share knowledge, discuss approaches, encourage participation in/ liaison with standards organizations/committees to help develop more effective ways of building highly dependable software applications, and more effective ways of evaluating the dependability, efficacy, and especially safety of these applications

Early ideas on SCC and our view of the research topics of most importance can be found in a position paper produced as a result of the 2nd meeting of SCC in December 2007.

John Hatcliff, Mats Heimdahl, Mark Lawford, Tom Maibaum, Alan Wassyng, Fred Wurden, “A Software Certification Consortium and its Top 9 Hurdles,” In Proceedings of the First Workshop on Certification of Safety-Critical Software Controlled Systems (SafeCert 2008), Electronic Notes in Theoretical Computer Science, Vol. 238, No. 4, pp. 11-17, 2009.

Funding

- Current status
  - McMaster has funded many of the activities, and some support for SCC is called for in our Ontario funded certification project
  - Members hosted and (sometimes partially) funded meetings
  - Members pay their own way to meetings
  - There is no “SCC funded” research
- Immediate future
  - Applying to an Ontario grant program to support SCC directly
    - Admin and some support for meetings – requires in-kind support from members
  - There is a move to start a UK/EU version by people at York (UK)
- Ideally
  - Co-operative funding from Canada & US

Previous meetings

- August 2007, SEI Offices in Arlington Virginia
  - Original goals & objectives
- December 2007, University of Minnesota
  - Hurdles, SoftCert paper
- April 2008, SEI Offices in Arlington Virginia
  - Technical discussion, Direction for SCC
- May 2010, University of Pennsylvania
  - Draft Charter, Technical discussion
- August 2010, hosted by NRC, Rockville Maryland
  - Draft Charter, Plan for research, Technical discussion
- November 2010, 2-day workshop at IBM CASCON, Toronto
  - Charter, Technical papers

Members

- One of the remaining “issues” in the Charter
  - Want to make it easy to be a member, but also want some commitment
- SCC members are individuals, not institutions, companies, agencies, etc

SCC Objectives

- The SCC is organized to pursue the following objectives
  - To promote the scientific understanding of certification for Systems containing Software (ScS) and the standards on which such certification is based
  - To promote development and improvement of consensus standards supporting certifiable software-intensive systems and their certification, through transfer of knowledge to existing standards organizations
  - To promote public, government and industrial understanding of the concept of ScS certification and the acceptance of the need for certification standards for software related products
  - To co-ordinate software certification initiatives and activities to further the above objectives

Goals to Achieve SCC Objectives

- Primary Goal
  - Develop and document a generic framework for certification, supporting domain specific certification frameworks and criteria
- Detailed Goals
  - Use existing knowledge to develop appropriate evidence-based standards and audit points for critical software in specific domains, including hard real-time, safety-critical systems
  - Research and develop improved methods and tools for the development and certification of critical software, conforming to the above standards and audit points
  - Proof of concept: Develop and document software requirements and necessary system requirements and constraints that help developers and regulators in the realization of critical software applications in specific domains

Principle

- It is reasonably obvious (maybe) – but still needs to be said:

- There are two complementary aspects –
  - Need to determine how to build software applications that can be certified effectively
  - Need to determine how to certify software applications effectively

  - Eventually, as our knowledge about software applications improves, these will be more symbiotically related than they are now

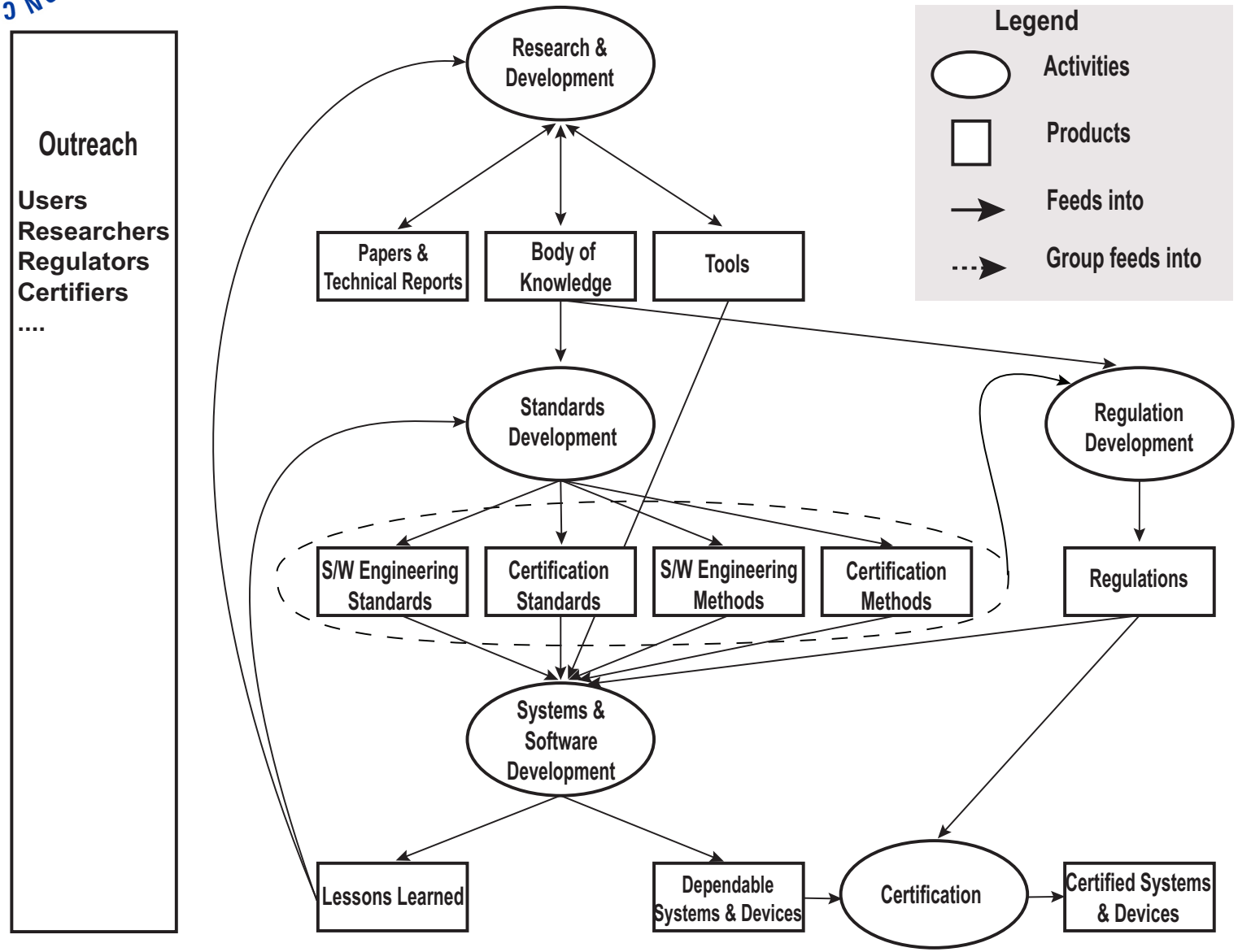
Scope & Deliverables - 1

- Requires coordination of the work program of SCC partners
- Research and Development
  - To produce research papers and technical reports focusing on approaches and techniques in software engineering for certifiable software-intensive systems and their certification
  - To develop a Body of Knowledge related to the development of certifiable software-intensive systems and their certification
  - To develop knowledge for evaluating tools supporting the development of certifiable software-intensive systems and their certification, including qualification of commercial tools to support development and evaluation of certifiable systems
  - Standards Development: Foster development and improvement of consensus standards supporting certifiable software-intensive systems and their certification, through transfer of knowledge to existing standards organizations

Scope & Deliverables - 2

- Experience in the usage of the Standards, Methods and Tools to document operating experience
  - Systems and software development
  - Certification
  - Licensing approval

Scope & Deliverables



SCC Meeting Schedule

- Three 2-day meetings per year
  - Two business oriented meetings with some technical discussion
    - These are likely to be held mainly in the US and probably most often in the Washington DC area, since we need regulatory agencies to be deeply involved – and it has been difficult for participants from these agencies to travel outside the US
  - One technical workshop (with minimal business sessions)
    - One idea is to attempt to run this as an IBM CASCON Workshop every year, as we did in 2010. This is always in October/ November, in Markham which is at the north end of the Greater Toronto Area

The Time is Right

- We started a software certification initiative in 2004
- Could not get people to take it seriously
- The fact that SCC started off successfully in 2007 was due to the fact that interest in software certification was starting to build
- There are now workshops and tracks at conferences dedicated to software certification
- Many people still think you can develop technology and then bolt on a certification aspect – it does not work (easily/effectively)

Major thrusts

- Product focused standards and certification
- Assurance cases
- Experimental aspects
- Dealing with complexity
- Example Challenges

Product focus

- Product focused standards and certification
  - Many of us strongly believe that process based approaches have failed, and that we do this simply because we do not yet understand how to examine the products adequately
  - Most other engineering disciplines evaluate the product
  - Process compliance grew out of concern for quality related to manufacturing the product. It is wider than that now, but there is a significant difference between manufacture of physical products and the manufacture of software products

Assurance cases

- Assurance/safety cases
  - They are being used more and more, and hold out promise for a much more evidence based, product focused approach
  - Some of us are worried about the lack of “prescription” in assurance/safety cases, in terms of certifying authorities being able to develop expertise in evaluating assurance cases
  - There is also concern that there is no underlying theory that helps us evaluate assurance cases in an objective and repeatable way

Experimental aspects

- Experimental aspects and metrics
  - Many of us agree that the state of the practice in experimental software engineering is disgraceful
  - Most of our “evidence” is anecdotal (including that used as the basis of standards), rather than experiment based
  - This topic also includes software metrics. After many years of work on software metrics, it is difficult to find metrics that are supported by experimental evidence, and will aid us in evaluating the dependability of software applications
  - If we are to have any success in product focused certification, all of this will have to change

Complexity

- Complexity – the root of all evil
  - Dealing with complexity is our biggest challenge
  - It is both a technical and political challenge
  - The US is deeply committed to R&D of cyber-physical systems. One of the defining characteristics of CPS is the inherent complexity of the system
  - Beyond a certain limit of complexity it becomes extremely difficult to demonstrate that the system is dependable
  - Traditional modularization that we teach and practice has a severe limitation. We end up moving the inherent complexity in the system, from inside modules, to the interface between modules
  - Separation of concerns, and in particular, separation of control from safety systems, needs to be better understood and is a potential golden principle – but not a silver bullet

Challenges

- The Pacemaker Challenge
  - Started in 2007. Brian Larson, who was at Boston Scientific, fought to get permission to present a requirements document of a real, ten-year-old pacemaker
  - The Wiki is at

    http://www.cas.mcmaster.ca/wiki/index.php/Pacemaker
  - Currently being revamped
- The Generic Infusion Pump
  - Discussion is on-going as to whether we can/should use this as an SCC challenge. It grew out of work at FDA, U Penn and Fraunhoffer CESE
  - The U Penn web site is at

    http://rtg.cis.upenn.edu/gip.php3

Research attitude

- Universities – tenure & promotion are barriers to making adequate progress on fundamental practical problems
- Industry – aware of the downside in academic research but often overlook the advances that have been made that could be useful
- "Certification" has not been exciting enough to get adequate funding – starting to see changes (2009 Ontario Research Fund – Research Excellence, $21M project on "Certification of Safety-Critical Software-Intensive Systems")

Collaboration: SCC & HCSS

- We have existing members who are also involved in HCSS
- This presentation was really about SCC
- Some thoughts:
  - SCC's main success has been to facilitate open, non-threatening, informal technical (and occasionally political) discussion on the essential problems involved in (Software) Certification
  - HCSS has the weight and influence of the US government behind it, and seems to have very similar goals and concerns to those of SCC
  - SCC members come from academia, industry, and government agencies in both the US and Canada
  - SCC has very modest funding requirements for a potentially large pay-off