EMBEDDED, REAL-TIME AND OPERATING SYSTEMS (ERTOS) PROGRAM

National ICT Australia

August 2003
National ICT Australia (NICTA)

- National research flagship for IT and Communications
- Established by Australian Government October 2002
- 4 core partners:
  - 2 universities: UNSW, ANU
  - 2 state governments: NSW, ACT
- Funding for first 4 years: A$200M (A$120M federal gov’t)
National ICT Australia (NICTA)

- National research flagship for IT and Communications
- Established by Australian Government October 2002
- 4 core partners:
  - 2 universities: UNSW, ANU
  - 2 state governments: NSW, ACT
- Funding for first 4 years: A$200M (A$120M federal gov’t)
- Steady-state federal funding: A$48M/a (indicative)
- Steady-state budget: A$100M/a (estimate)
NICTA: Four Pillars

- Research
- Education
- Commercialisation
- Linkages
NICTA: Four Pillars

- **Research**
  - Commitment to world class research across a wide range of ICT
  - Target: 300 PhD-qualified researchers

- **Education**

- **Commercialisation**

- **Linkages**
NICTA: Four Pillars

- Research
  - Commitment to world class research across a wide range of ICT
  - Target: 300 PhD-qualified researchers

- Education
  - Training of top-class PhD graduates
  - Target: 100 PhD graduates per year

- Commercialisation

- Linkages
NICTA: Four Pillars

- **Research**
  - Commitment to world class research across a wide range of ICT
  - Target: 300 PhD-qualified researchers

- **Education**
  - Training of top-class PhD graduates
  - Target: 100 PhD graduates per year

- **Commercialisation**
  - Attractive IP/commercialisation policies
  - Create a commercialisation culture

- **Linkages**
NICTA: Four Pillars

- Research
  - Commitment to world class research across a wide range of ICT
  - Target: 300 PhD-qualified researchers

- Education
  - Training of top-class PhD graduates
  - Target: 100 PhD graduates per year

- Commercialisation
  - Attractive IP/commercialisation policies
  - Create a commercialisation culture

- Linkages
  - International: top research institutions, MNCs
  - Domestic: SMEs
NICTA Overview: Research “Themes”

1. Infrastructure Technologies (InfT)

2. Software Engineering (SE)

3. Intelligent Systems (IntS)

4. Human-Machine Interaction and Usability (HMIU)

5. Foundations (Found)
NICTA Overview: Research “Themes”

1. Infrastructure Technologies (InfT)
   ➔ 6 Programs, 2 planned

2. Software Engineering (SE)
   ➔ 2 Programs, 2 planned

3. Intelligent Systems (IntS)
   ➔ 4 Programs, 1 planned

4. Human-Machine Interaction and Usability (HMIU)
   ➔ 1 Programs, 1 planned

5. Foundations (Found)
   ➔ 2 Programs, 2 planned
NICTA OVERVIEW: 3 NODES

- Sydney Research Lab — 2 locations:
  - UNSW Campus (4 Programs)
  - Australian Technology Park (3 Programs)

- Canberra Research Lab: ANU Campus (5 Programs)
**NICTA Overview: 3 Nodes**

- **Sydney Research Lab** — 2 locations:
  - UNSW Campus (4 Programs)
  - Australian Technology Park (3 Programs)

- **Canberra Research Lab**: ANU Campus (5 Programs)

- **NICTA Fellows**
  - Located at other Australian Universities
  - Part of NICTA’s commitment to the national interest
NICTA OVERVIEW: PRESENT RESEARCH PROGRAMS

- Sydney Research Lab, UNSW Site:

- Sydney Research Lab, ATP Site:

- Split, Sydney and Canberra Research Labs:
NICTA Overview: Present Research Programs

- Sydney Research Lab, UNSW Site:
  - Embedded, Real-Time and Operating Systems (Heiser, InfT)
  - Formal Methods (van der Meyden, SE)
  - Symbolic Machine Learning & Knowledge Acquisition (Sharma, IntS)
  - Knowledge Representation & Reasoning (Foo, IntS)

- Sydney Research Lab, ATP Site:
  - Networks and Pervasive Computing (Seneviratne, InfT)
  - Empirical Software Engineering (Jeffery, SE)
  - Humans Understanding Machines (Eades, HMIU)

- Split, Sydney and Canberra Research Labs:
  - Systems Engineering and Complex Systems (Anderson, Found)
- Canberra Research Lab:
Canberra Research Lab:

- Wireless Signal Processing (Kennedy, InfT)
- Autonomous Systems & Sensing Technology (Hartley, IntS)
- Logic & Computation (Lloyd, Found)
**Embedded System**

Computer system that is part of a larger system
**GENERAL-PURPOSE VS. EMBEDDED**

- Traditional model of embedded systems
General-Purpose vs. Embedded

- Traditional model of embedded systems
  - No longer true for complex and networked embedded systems!
CRITICAL ISSUES FOR EMBEDDED SYSTEMS

• Development cost
• Unit cost
• Time to market
• Size
• Performance
• Reliability
• Security
CRITICAL ISSUES FOR EMBEDDED SYSTEMS

- Development cost
- Unit cost
- Time to market
- Size
- Performance
- Reliability
- Security
ERTOS Vision

To develop methodologies, tools, components and systems that will deliver reliable, inexpensive system *software* meeting its requirements.
ERTOS VISION

To develop methodologies, tools, components and systems that will deliver reliable, inexpensive system software meeting its requirements.

ERTOS research will be driven by applications

- to identify common challenges
- to provide generic systems software
GRAND CHALLENGE: TRUSTWORTHY SYSTEMS

- Reliability of (embedded) systems is a major concern
- Can only really trust a system once mathematically proven correct
GRAND CHALLENGE: TRUSTWORTHY SYSTEMS

- Reliability of (embedded) systems is a major concern
- Can only really trust a system once mathematically proven correct
- Proofs for high-level parts of a system are of limited use
  - Need to assume that remaining parts are correct
- Essential to deal with hardware-software interface
- Difficult because:
  - Side effects of hardware
  - Complexity of operating system code
TRUSTWORTHY SYSTEM

- Must prove safety properties for *whole* system
TRUSTWORTHY SYSTEM

• Must prove safety properties for *whole* system
  ➔ Break system into small components of manageable size
  ➔ Develop formal models of each component
  ➔ Prove that each component satisfies requirements
  ➔ Prove that whole system satisfies requirements
TRUSTWORTHY SYSTEM

• Must prove safety properties for whole system
  ➔ Break system into small components of manageable size
  ➔ Develop formal models of each component
  ➔ Prove that each component satisfies requirements
  ➔ Prove that whole system satisfies requirements

• Must ensure that components interact via defined interfaces only
**Trustworthy System**

- Must prove safety properties for *whole* system
  - Break system into small components of manageable size
  - Develop formal models of each component
  - Prove that each component satisfies requirements
  - Prove that whole system satisfies requirements

- Must ensure that components interact via defined interfaces only

- Key issues:
  - components
  - encapsulation
LONG-TERM VS. INTERMEDIATE GOALS

• Work on the Grand Challenge is
  ➔ medium- to long-term
  ➔ high-risk
  ➔ potentially disruptive
LONG-TERM VS. INTERMEDIATE GOALS

● Work on the Grand Challenge is
  ➔ medium- to long-term
  ➔ high-risk
  ➔ potentially disruptive

● Will also perform research that is
  ➔ short- to medium-term
  ➔ medium-risk
  ➔ incremental
  ➔ addresses present challenges
LONG-TERM VS. INTERMEDIATE GOALS

• Work on the Grand Challenge is
  ➔ medium- to long-term
  ➔ high-risk
  ➔ potentially disruptive

• Will also perform research that is
  ➔ short- to medium-term
  ➔ medium-risk
  ➔ incremental
  ➔ addresses present challenges

• Outcomes:
  ➔ insights, design principles, methodologies
  ➔ software: kernels, compilers, frameworks, tools
  ➔ closing in on the Grand Challenge
**EMBEDDED SYSTEMS CONSTRAINTS**

- Technological Change
- Reliability
- Real-time
- Performance
- Security
- Size
- Power
- Requirement Changes

Life-cycle costs:
- design
- implementation
- maintenance
STRATEGIES FOR MEETING THE CHALLENGES

- Generic frameworks which can be specialised
- All-of-systems approach
- Open Source
STRATEGIES FOR MEETING THE CHALLENGES

• Generic frameworks which can be specialised
  ➔ application driven
  ➔ based on microkernel approach

• All-of-systems approach

• Open Source
STRATEGIES FOR MEETING THE CHALLENGES

• Generic frameworks which can be specialised
  ➔ application driven
  ➔ based on microkernel approach

• All-of-systems approach
  ➔ address challenges at all levels of system

• Open Source
STRATEGIES FOR MEETING THE CHALLENGES

• Generic frameworks which can be specialised
  ➔ application driven
  ➔ based on microkernel approach

• All-of-systems approach
  ➔ address challenges at all levels of system

• Open Source
  ➔ share infrastructure cost
  ➔ ease uptake
# ERTOS Overview

<table>
<thead>
<tr>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance/Power</td>
</tr>
<tr>
<td>Security/Reliability/Safety</td>
</tr>
</tbody>
</table>

- Operating Systems
- Languages & Compilers
- Architectures
# ERTOS Overview

<table>
<thead>
<tr>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance/Power</td>
</tr>
<tr>
<td>Security/Reliability/Safety</td>
</tr>
</tbody>
</table>

- Operating Systems
- Languages & Compilers
- Architectures

- Biomedical
- Solar Car
- Satellite
- Biodiversity Monitoring
ERTOS OVERVIEW

<table>
<thead>
<tr>
<th>Cost</th>
<th>Performance/Power</th>
<th>Security/Reliability/Safety</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Operating Systems</td>
<td>Languages &amp; Compilers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Biomedical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solar Car</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Satellite</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Biodiversity Monitoring</td>
</tr>
</tbody>
</table>
ERTOS OVERVIEW

- Cost
- Performance/Power
- Security/Reliability/Safety

- Operating Systems
- Languages & Compilers
- Architectures

- Biomedical
- Solar Car
- Satellite
- Biodiversity Monitoring

- Robotics
- Automotive
- Sensor Networks
- Games
INDICATIVE PROJECTS

• Formal modeling of low-level system code
  ➔ with Formal Methods Program
  ➔ first step towards meeting Grand Challenge
INDICATIVE PROJECTS

- Formal modeling of low-level system code
  ➔ with Formal Methods Program
  ➔ first step towards meeting Grand Challenge

- Light-weight high-performance distributed robot OS
  ➔ with ARC Centre of Excellence for Autonomous Systems
  ➔ with Symbolic Machine Learning Program
  ➔ with Autonomous Systems & Sensor Technologies Program
INDICATIVE PROJECTS

• Formal modeling of low-level system code
  ➔ with Formal Methods Program
  ➔ first step towards meeting Grand Challenge

• Light-weight high-performance distributed robot OS
  ➔ with ARC Centre of Excellence for Autonomous Systems
  ➔ with Symbolic Machine Learning Program
  ➔ with Autonomous Systems & Sensor Technologies Program

• Sensor networks software infrastructure
  ➔ with Networks & Pervasive Computing Program
  ➔ also, personal area network demonstrator
  ➔ industrial client desirable
**INDICATIVE PROJECTS...**

- Embedded systems on a chip
  - with Wireless Signal Processing Program
  - with potential Circuits & Systems Program
INDICATIVE PROJECTS...

- Embedded systems on a chip
  ➔ with Wireless Signal Processing Program
  ➔ with potential Circuits & Systems Program

- High-performance switching systems
  ➔ with Optical Processing & Monitoring Program
INDICATIVE PROJECTS...

- Embedded systems on a chip
  → with Wireless Signal Processing Program
  → with potential Circuits & Systems Program

- High-performance switching systems
  → with Optical Processing & Monitoring Program

- Real-time systems design
  → with Formal Methods Program
  → possible applications: robotics, automotive
INDICATIVE PROJECTS...

- Embedded systems on a chip
  - with Wireless Signal Processing Program
  - with potential Circuits & Systems Program

- High-performance switching systems
  - with Optical Processing & Monitoring Program

- Real-time systems design
  - with Formal Methods Program
  - possible applications: robotics, automotive

- Embedded systems product line
  - with Empirical Software Engineering Program
  - possibly with Fraunhofer Software Engineering Institute
Education

- Global shortage of graduates/PhDs with good “systems” skills

- UNSW is only place left in Australia where students get to:
  - get real experience with low-level systems code
  - build sizable systems from the ground up
  - get trained with real-live systems (Linux)
**EDUCATION**

- Global shortage of graduates/PhDs with good “systems” skills
- UNSW is only place left in Australia where students get to:
  - get real experience with low-level systems code
  - build sizable systems from the ground up
  - get trained with real-live systems (Linux)
- Undergraduate student achievements:
  - 3 × winners CISRA Project Prize
  - 2 × NSW winners Siemens Prize for Innovation
  - 4 × winners Aurema Operating Systems Prize
  - winner AUUG Open Source Prize
  - 2 × winners AUUG John Lions Award
  - 3 placed as interns at IBM Watson (6–12 months each)
    - latest round of applications just closed (3 applied)
EDUCATION

- Advanced-Level Courses, coverage unique in Australia:
  - Advanced Operating Systems
  - Advanced Compilers
  - Advanced Functional Languages
EDUCATION

- Advanced-Level Courses, coverage unique in Australia:
  - Advanced Operating Systems
  - Advanced Compilers
  - Advanced Functional Languages

- PhD students
  - presently 20 students
  - 7 commenced in 2003
EDUCATION

- Advanced-Level Courses, coverage unique in Australia:
  - Advanced Operating Systems
  - Advanced Compilers
  - Advanced Functional Languages

- PhD students
  - presently 20 students
  - 7 commenced in 2003

- Summer Scholarships for Undergraduates:
  - 18 students in 2002–3
  - similar number expected next summer
STRATEGIC LINKAGES

- IBM T J Watson Research Center and OzLabs (HPCS)
- HP Labs
- Microsoft Cambridge Lab
- Xilinx, Intel
- CMU, UIUC, Waterloo, Karlsruhe and Dresden U, Barcelona
- Partner in EU FP6 Project
  ➔ ST Microelectronics, Dresden, Prague
COMMERCIALISATION

• Presently all work is in open-source domain
  ➔ will continue to open-source generic infrastructure
  ➔ client-specific code subject to standard commercialisation arrangements
COMMERCIALISATION

- Presently all work is in open-source domain
  - will continue to open-source generic infrastructure
  - client-specific code subject to standard commercialisation arrangements

- Reasons:
  - little profit in basic software infrastructure
  - impact is more important to NICTA than cash
  - unrestricted accessibility (BSD License) encourages uptake
COMMERCIALISATION

- Presently all work is in open-source domain
  - will continue to open-source generic infrastructure
  - client-specific code subject to standard commercialisation arrangements

- Reasons:
  - little profit in basic software infrastructure
  - impact is more important to NICTA than cash
  - unrestricted accessibility (BSD License) encourages uptake

- Goal: Create the BSD of Embedded Systems!
**SUMMARY**

**ERTOS WILL:**

- Make conceptual contributions to software frameworks and methodologies for the development of embedded systems
  
  ➔ A concrete outcome will be kernels, systems and tools which will be widely available and used, and will enhance the NICTA brand

- Produce concrete applications of these systems in specific domains which will lead to commercialisable outcomes

- Build capabilities to overcome a lack of systems expertise — a critical resource for the future of Australia