DoD Data & Analysis Center for Software

DACS Survey on the State of the Practice of Development Tools to Support Embedded Software Development

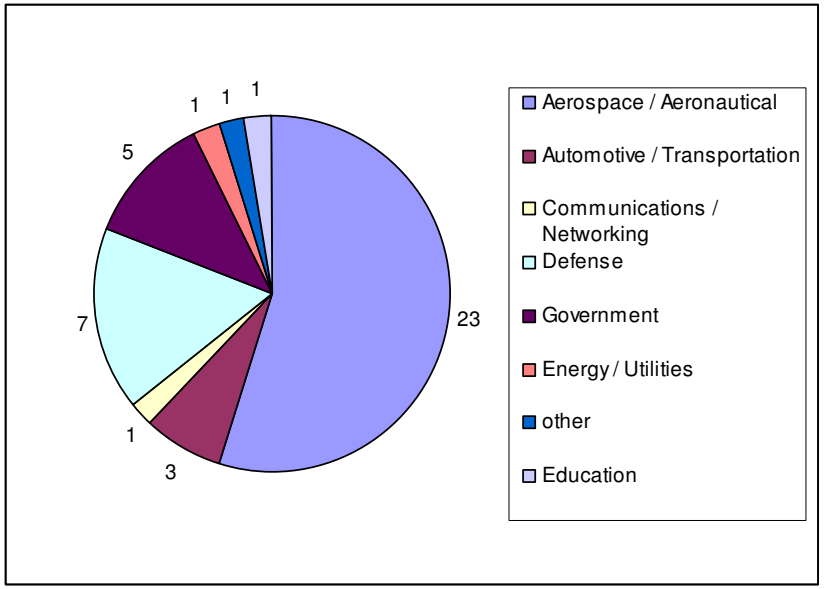
Tom McGibbon, CSDP
DACS Director
tom.mcgibbon@itt.com
315.334.4933

Survey Objectives

- Increased complexity within net-centric and SoS embedded software
- New tools are being utilized
- Many believe tools are not adequate
- Survey
  - State of tools for embedded software engineering
  - Identify benefits received from tools utilized
  - Identify shortcomings and gaps
- Taxonomy developed based on tool chains to aid in survey development

Survey Demographics

- Internet based survey
- 44 responses
  - 36 respondents directly solicited
  - 8 respondents responded to DACS user-wide solicitation
  - Engineers Developing Embedded Components of:
    - F-22,
    - Global Hawk,
    - JSF,
    - F16 RWR,
    - ECM,
    - Satellite Software,
    - Boeing 767,
    - GTTA HDC,
    - Radar
  - Types of Software
    - Control
    - Real Time
    - Diagnostic
    - Flight Software



Embedded Systems

DoD Data & Analysis Center for Software

- Biggest Challenges in Designing:

29 out of 44 Respondents answered this question.

This open ended question received a variety of responses from survey participants. In order to uncover any trend among the responses each response was categorized in the following categories below.

Response Category	Number
Change in Scope/Requirements Creep/Poor Requirements	13
Problems discovered in Testing/Integration	5
Poor Estimatation/Scheduling/Aggressive Schedule	5
Not enough staff/resources	3
Hardware Changes/Issues/Bugs	3
Fuding Changes/Unrealistic Budget	3
Dependent systems/software not ready	2
Conflicting Program Priorities	2
Unique Challenges not faced before/Unforseen technical Challenges	2
Lack of Stable Infrastructure/Middelware	1
Lack of Planning/Poor Management	1
Performance Issues	1
	1

Embedded Systems

DoD Data & Analysis Center for Software

- Design Approaches

32 out of 44 Respondents answered this question.

The following is a rank order list of software design methodologies from most to least frequently selected.

Object Oriented Design	24
Model Based Development	20
Rapid Prototyping	20
Simulation and Modeling	21
Automatic Code Generation	14
ADARTS/CODARTS	9
Design Verification and Validation	8
Agile	7
Coverification	5
Other	3
JSD (Jackson System Development)	1
Product Engineering	1

20 out of 44 Respondents answered this question.

The following is a list of the most notable **drawbacks** identified.

- Both Object Oriented with C++ and automatic code generation produce significantly more code.

- Tools all force users to follow their process and methodologies even if they don't make sense from a business perspective.

- OO design has been somewhat problematic. It is not always applied properly to our systems (which have severe real-time constraints & memory limitations). It is difficult to gain visibility into the execution of the OOD components & also difficult to prove that the code has actually been tested (code coverage).

- Tools are costly and difficult to learn and use. Buggy. Don't support my processor of choice.

- Automatic code generation works for GUI's, but not ready for prime-time for anything real-time or large.

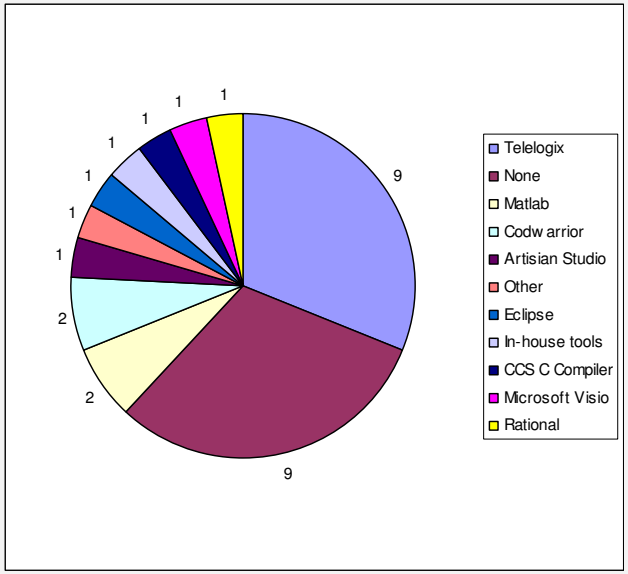
- The tool needed for effective MBD are very expensive. In addition, DO-178B does not yet fully address the use of MBD methods

Embedded Systems

DoD Data & Analysis Center for Software

- Software Design & Modeling Tools:

29 out of 44 Respondents answered this question.



12 out of 44 Respondents answered this question.

The following is a partial list of **lacking capabilities**.

- Having the ability to import/export models from other tools (or have a COMMON file format among the tools). Ease of set up for different environments and the ability to easily switch between them. More/better on line tutorials.

- Customizability. Higher modeling languages. More robustness in simulation. More advanced analysis techniques. More support for co-simulation.

- Reverse code generation, also known as round-trip engineering. We need to be able to build a model, have code generated from the model, change the code and have the midel updated automatically.

- Tools should be as simple to use as pencil and paper.

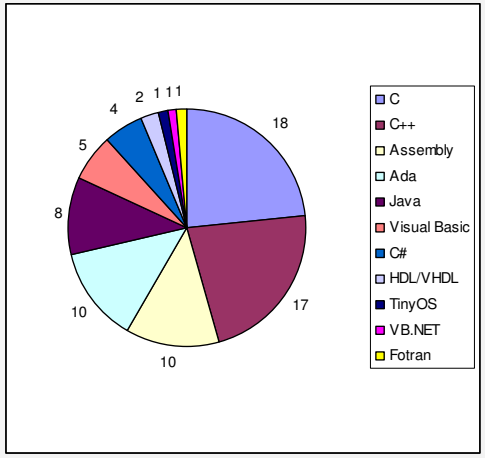
- Perfect code import / reverse-engineering, better predictive heuristics to reduce the amount of design data that must be entered before generating code.

Other Questions

DoD Data & Analysis Center for Software

- Why projects late/cancelled?
- Biggest Challenges Faced…
- Drawbacks from use of design methodologies
- Have you developed any in-house tools? Why?
- Missing capabilities/problems from
  - RM tools
  - CM tools
  - Formal Analysis Tools
  - Testing Tools
  - Programming Languages
  - IDEs
  - RTOSs
- Unique Challenges with
  - Systems of Systems
  - Net Centric Systems
- What challenges do you see on the horizon for your embedded system projects?

27 out of 44 Respondents answered this question.



Other Languages mentioned:

- Tiny OS

Conclusions (Partial)

DoD Data & Analysis Center for Software

Statement of Problem	Better embedded systems/software development technology is needed for…	Number Of Observations from Survey	Need Supported In Literature
1. Software development technology is immature for embedded system development	Total end-to-end/requirements-to-release development environment	15	
	A common software development foundation with plug-in components	6	√
	Maintaining consistency and compliance between requirements, design, and code	5	
2. Observability during Test and Debug is a Significant Challenge	Observing and analyzing system behavior	12	√
	Observing system QoS performance	6	√
	Observing system behavior and interaction in embedded & distributed systems	4	√
	Observing memory utilization	2	√
3. Developing and testing software when hardware or other parts of the architecture are missing or not available or TBD is difficult. 3.1 Doing hardware and software tradeoffs during design is difficult	Developing executable behavioral models and simulations of hardware, software, and interfaces	12	√
4. Software development technology for systems with constrained memory and hard real time requirements is lacking.	Better programming languages for embedded systems that addresses constrained memory and performance	10	

Further Information

- Download the Document from the DACS Website:

  http://www.thedacs.com

- Contact Information

Tom McGibbon

tom.mcgibbon@itt.com

315.334.4933