

COTS Product Selection for Safety-Critical Applications

Fan Ye, Tim Kelly
Department of Computer Science
University of York, York, UK
E-mail: fan.ye, tim.kelly@cs.york.ac.uk

Outline

- Background – COTS, COTS-based system, safety-critical system
- COTS use in safety-critical context
- Existing methods for COTS selection
- Proposed method
- Beyond selection
- Conclusion

COTS & COTS-Based System

- COTS = *Commercial-Of-The-Shelf*
- Related terms: OTS (COTS, MOTS, GOTS), NDI, SOUP
- Standard commercial software developed without any particular application in mind [*McDermid 98*]
- A COTS product is a product that is [*Mayer & Oberndorf 01*]
 - sold, leased, or licensed to the general public
 - offered by a vendor trying to profit from it
 - supported and evolved by the vendor, who retains the intellectual property rights
 - available in multiple, identical copies
 - used without modification of the internals

COTS & CBS (Cont'd)

- Examples of COTS Products
 - Application level, e.g., Stores Management System, Word Processor
 - Infrastructure level, e.g., Operating Systems, libraries
 - Supporting tools, e.g., Compilers, testing tools
- **COTS-Based Systems (CBS)** *are composed of off-the-shelf parts integrated to achieve new/expanded system functionality (SEI)*
- Examples of COTS-Based Systems
 - DoD's Global Transportation Network (GTN) (>50 COTS)
 - U.S. Navy's Radar Tracking System
 - NASA's EPOCH2000 Satellite Ground System

Benefits & Pitfalls

- Cheaper
 - ✓ Cost shared among all consumers
 - ✗ Increased continual investment
- Faster
 - ✓ Functionality can be accessed immediately
 - ✓ Time-to-market
 - ✗ Integration effort
- Better
 - ✓ Better tested
 - ✓ Hands-on experience/prototype
 - ✗ Malicious code
- Other benefits & pitfalls
 - ✓ Up-to-date technology
 - ✓ Increased portability
 - ✗ Support problems

Safety-Critical System

- A **safety-critical system** is *a system whose incorrect state (failure) may have serious consequences such as loss of human life, severe injuries, or large-scale environmental damage*

[Hauge 2001]

Safety-Critical System (Cont'd)

- Certification requirements
 - Need safety case
- Principal objective of safety case
 - Safety case presents the argument that a system will be acceptably safe in a given context.
 - *"The software safety case shall present a well-organised and reasoned justification based on objective evidence, that the software does or will satisfy the safety aspects of the Statement of Technical Requirements and the Software Requirements specification."* (U.K. Defence Standard 00-55)
- A safety case requires two elements:
 - *Supporting Evidence* (e.g. analysing and testing results, simulation)
 - *High Level Argument*

Use of COTS in SC Context

- Economic necessity
- What standards say about COTS use in SC context
- Challenges
 - Evidence to support safety
 - Limitation on access to: *[Jones, et al 2001]*
 - ✘ Descriptions of development processes
 - ✘ Design documentation
 - ✘ Source code
 - ✘ Fault histories
 - Safety assurance under change (e.g. upgrades)
 - Lack of systematic approach taken application specific safety considerations into account

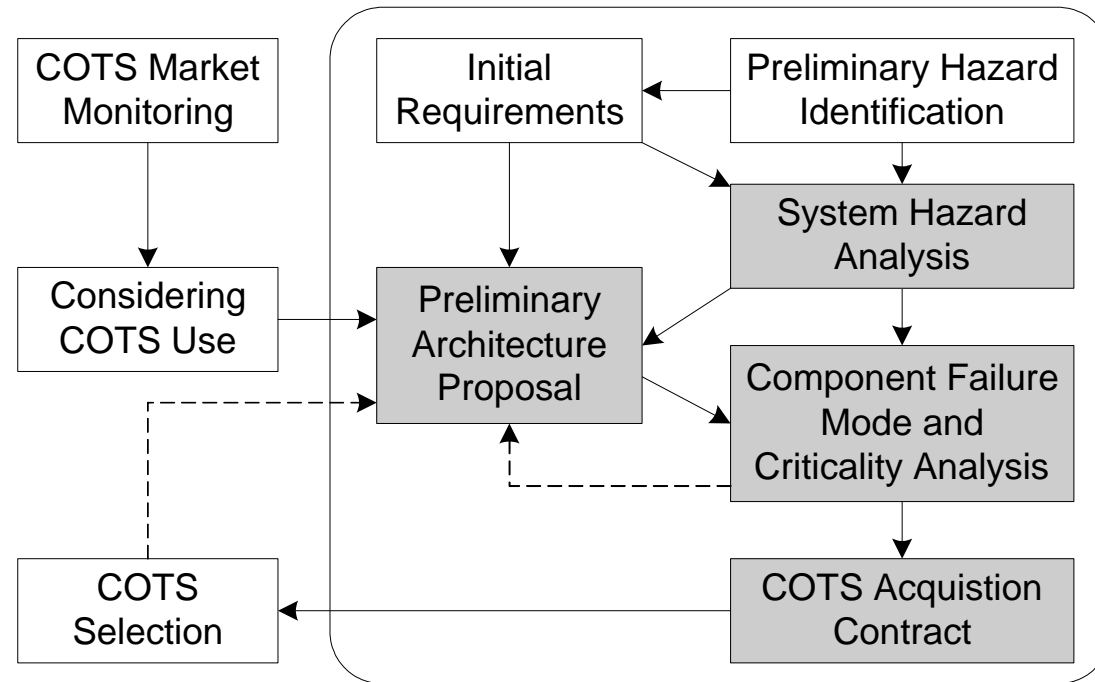
Existing COTS Selection Methods

- Methods
 - Procurement-Oriented Requirement Engineering (PORE), Off-The-Shelf Option (OTSO), COTS Acquisition Process (CAP), Social-Technical Approach to COTS Evaluation (STACE)
- Problems
 - Not dedicated to safety application
 - Lack of consideration of safety

Proposed Method

- Principles
 - Take safety consideration early, seriously, and systematically
 - Careful safety analysis used to define acquisition contract
 - Contract terms used as selection and evaluation criteria
 - Safety Assurance Level (SAL) dictates required evidence [*Weaver 03*]
 - SAL assigned to each contract term is commensurate to its criticality level

Proposed Method – CBCPS



- Method overview – major activities
 - Shaping required COTS functionality
 - Criticality analysis for COTS functionality
 - Contract establishment
 - COTS product selection

Phased Activities – #1

- Shaping required COTS component
 - Initial system requirements
 - Functional decomposition
 - COTS market monitoring
- OSC with COTS DB Example
 - Needs to store system safety case
 - Major functions
 - store, add, remove, edit, retrieve, enquire

Phased Activities - #2

- Criticality analysis for COTS functionality – to establish application specific safety requirements for COTS functionality
 - System hazards analysis
 - COTS failure modes analysis
 - HAZOP [*UK MOD 00-58*]
 - Relationship – fault tree analysis
 - Criticality determination

Phased Activities - #2 (Cont'd)

- Criticality analysis example

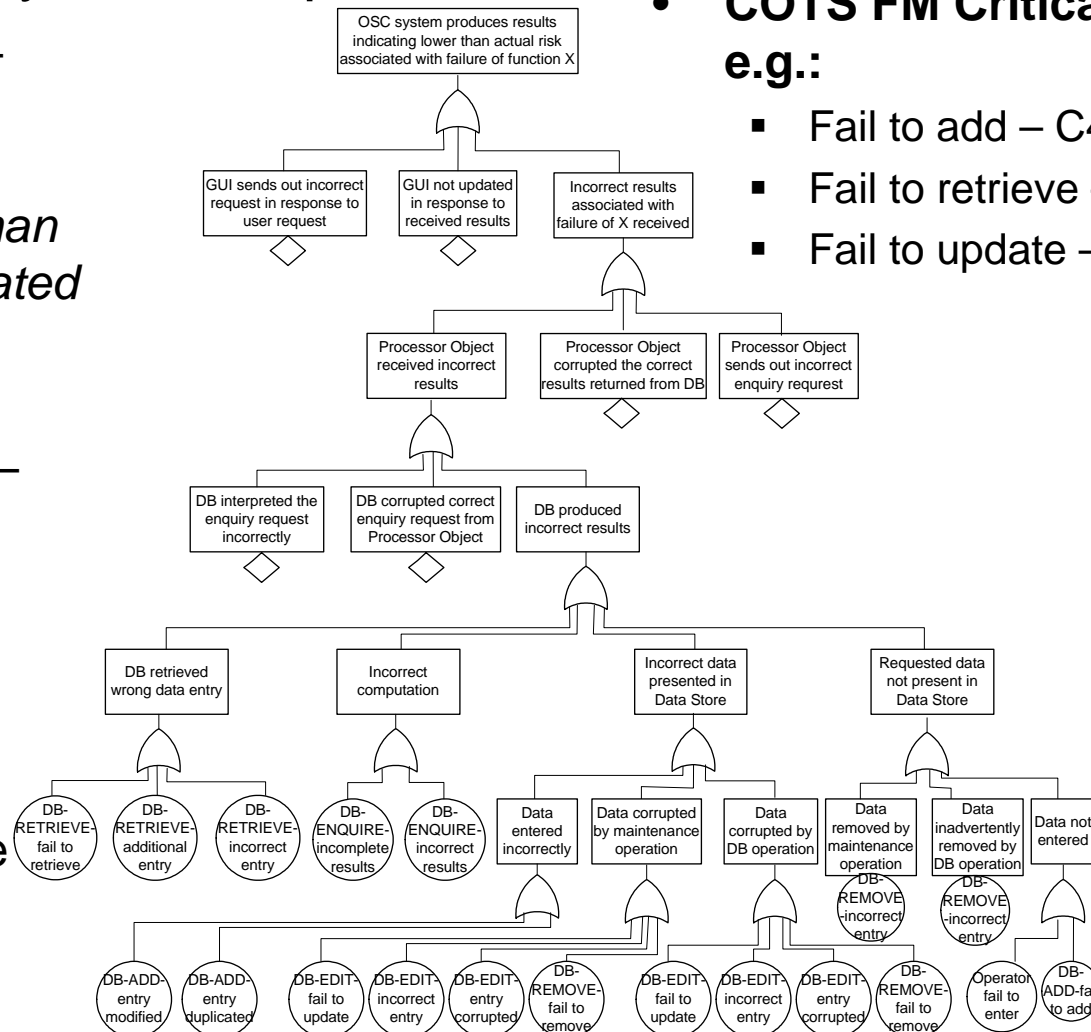
- **System Hazard** –
OSC produces incorrect results indicating lower than actual risk associated with the failure of Function X

- **Hazard Severity** –
Catastrophic

- **COTS FM** – e.g. fail to add would result in data source incomplete thus the system hazard.

- **COTS FM Criticality**
e.g.:

- Fail to add – C4
- Fail to retrieve – C4
- Fail to update – C4



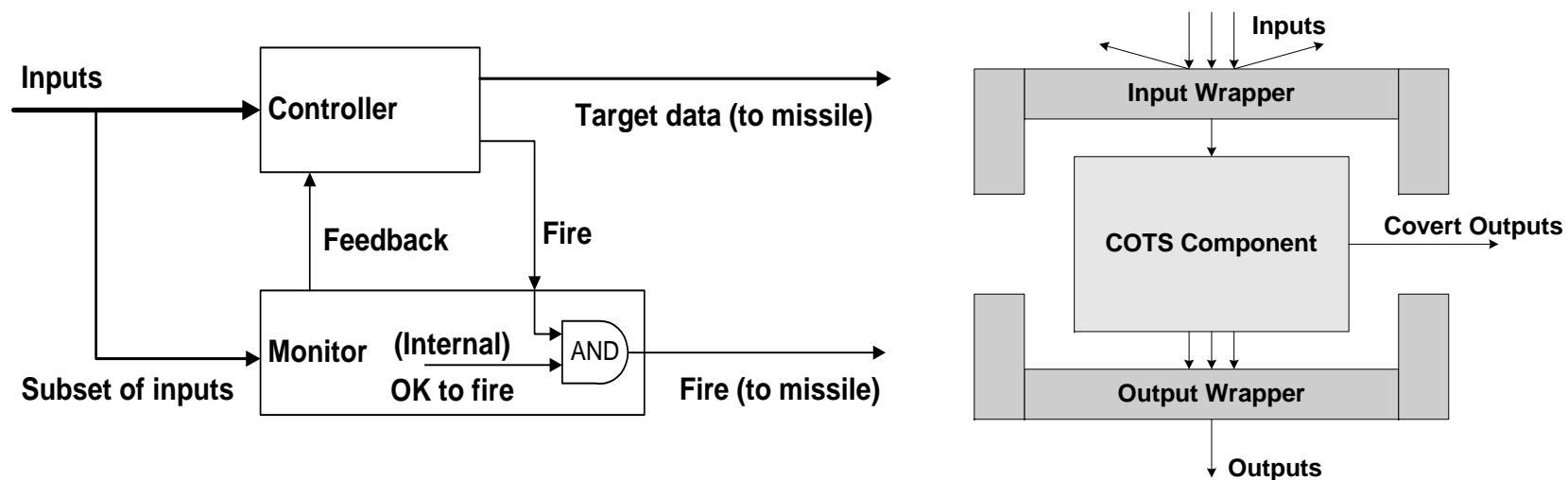
Phased Activities - #3

- Contract establishment
 - Criticality rating
 - SAL assignment
 - Evidence
- Example contract:

ID	Contract Term	SAL	Mitigation
1	DataBase Enquiry Service		
1.1	DB must not return incomplete enquiry results	3	Wrapper to check the results
1.3	The integrity of data item(s) must not be violated	4	Digital signature

Phased Activities - #4

- COTS product selection
 - Contract matching
 - Mismatch handling
 - Fault tolerant architecture
 - ⇒ Design diversity / N-version programming (e.g. Simplex, Control + Monitor)
 - ⇒ Component containment/COTS wrapping



Beyond Selection

- Interaction analysis
- Contract-based safety justification and safety case construction
- Contract-based safety maintenance

Benefits of CBCPS

- Take safety consideration early to avoid project failure due to failure to establish safety case
- Help later stage of safety justification and maintenance

Conclusion

- The need for COTS use in safety context
- The need for systematic approach that takes application specific safety requirements into account during COTS product selection
- The proposed CBCPS approach
- Potential further use of the established safety contract

Key References

- [McDermid 98] Talbert N., "**Interview with John McDermid: The Cost of COTS**," *IEEE Computer*, vol. 31, pp. 46-52, June 1998.
- [Mayer&Oberndorf 01] Meyer B. and Oberndorf P., ***Managing Software Acquisition: Open Systems and COTS Products***: Addison-Wesley, 2001.
- [Hauge 01] Hauge H. J., "**A Survey of Software Safety**," Department of computer and information science, Norwegian University of Science and Technology, November 2001.
- [Jones et al. 01] Jones C., Bloomfield R. E., Froome P. K. D., and Bishop P. G., "**Methods for Assessing the Safety Integrity of Safety-Related Software of Uncertain Pedigree (SOUP)**," Health and Safety Executive (HSE), Contract Research Report CRR337, 2001.
- [Weaver 03] Weaver R. A., Fenn J., and Kelly T. P., "**A Pragmatic Approach to Reasoning about the Assurance of Safety Arguments**," in *Proceedings of 8th Australian Workshop on Safety Critical Systems and Software (SCS'03)*, Canberra, Australia, 2003. P. Lindsay and T. Cant, Eds., Australian Computer Society.
- [UK MoD 00-58] MoD, "**00-58 HAZOP Studies on Systems Containing Programmable Electronics**," Ministry of Defence, Defence Standard, May 2000.