RE for configuration of OTS software

Evguenia (Elmi) Eflov

School of Computer Science
University of Waterloo

CS 846 Advanced Topics in Software Engineering
Requirements Engineering
2015
Outline

1 Motivation
   - Preliminaries
   - RE for OTS Details

2 Examples of Work on OTS Configurations
   - OTS Configuration and Adaptation
   - RE for COTS Software Stack
Outline

1. Motivation
   - Preliminaries
   - RE for OTS Details

2. Examples of Work on OTS Configurations
   - OTS Configuration and Adaptation
   - RE for COTS Software Stack
What is OTS (Off-The-Shelf software)?

- OS, DBMS, word processor, encryption library, etc.
- Usually specialized to perform a particular task in a reliable, secure, repeatable way
- **COTS** - Commercial Off-The-Shelf software
  - Designed by a dedicated team of experts
  - Well tested in a variety of environments
  - Has support (for a price) and documentation provided
- **OSS** - Open Source Software
  - Produced by a team of enthusiasts
  - Source is available and bug fixes can be made in the source
  - Support may be available
- **GOTS** - Government Off-The-Shelf software
- **SaaS** - Software as a Service
  - Hosted and maintained by the provider
  - Data is typically hosted in the same environment as software
**Why Use It?**

- **Time savings** - software is already implemented, tested, and ready to use
- **Cost reduction** - no need for additional development and support staff in house
- **Feature availability** - dedicated software includes wider variety of features than one-off implementations; experts involved in design are more likely to implement best practices in the industry
- **Higher stability** - dedicated testing for the OTS functionality in a wide variety of environments
- **Support** - support provided by the third party experts
- **New functionality** - new features likely to be provided
- **Infrastructure** - in case of SaaS, hardware infrastructure and maintenance is provided
Outline

1. Motivation
   - Preliminaries
   - RE for OTS Details

2. Examples of Work on OTS Configurations
   - OTS Configuration and Adaptation
   - RE for COTS Software Stack
Why is RE for OTS Different from Regular RE?

- Details of OTS behaviour may be unknown until implementation has started
- May need to adapt requirements to existing software
- Need to expect unexpected (resource requirements, concurrency issues, bugs that will not get fixed in time and have to be worked around)
- Additional security considerations - where is the data stored? who has access to it?
- Additional considerations for availability of third-party packages that can extend the functionality of the OTS
- Future considerations - features that are not currently relevant but may be desired in the future are part of the requirements
Why is RE for OTS Different? (cont’d)

New considerations not related to particular technical requirements

- **Support availability**
  - How long will the company stay in business?
  - Will support for the OTS continue?
  - What level of support is available?
  - Will support be extended to other environments/will SaaS become available?

- **Mainenance**
  - When/How are updates provided?
  - Are updates pushed/pulled?
  - How long are older versions of software maintained?

- **Licensing**
  - Updates/support included?
  - Complexity of obtaining additional licenses/features?
Generic Steps for OTS Selection

Step 1: Define evaluation criteria based on system requirements
Step 2: Search for OTS products
Step 3: Filter search results based on a set of ’must-have’ requirements in order to define a short list of OTS candidates to be evaluated in more details
Step 4: Evaluate OTS candidates (in the short list)
Step 5: Analyze the evaluation data (i.e. the output of Step 4) and select the OTS product that has the best fitness with the requirements
Outline

1. Motivation
   - Preliminaries
   - RE for OTS Details

2. Examples of Work on OTS Configurations
   - OTS Configuration and Adaptation
   - RE for COTS Software Stack
MiHOS - Mismatch Handling for COTS Selection

During COTS selection phase, evaluates fitness of COTS for the requirements after mismatch resolution - *anticipated fitness*

After COTS selection, used to plan the optimal set of adaptations and configurations - *mismatch-resolution plan*

Inputs to MiHOS may be imprecise due to estimation difficulties, documentation and manual errors, and language imprecision
Content Management System (CMS) selection for e-business solution

A set of 275 goals identified at different levels of abstraction

122 strategic goals - not directly measurable in a COTS

153 technical goals - correspond to COTS features

30 must-have technical goals selected and used to trim candidate list to 5 CMSs
Budget and Effort constraints were applied to determine *anticipated fitness* once mismatch resolution is applied to each candidate.

Iterative process was applied to identify COTS with highest *anticipated fitness* and full resolution of all mismatches related to ‘security’ goal.

*Note that the match between each feature and technical goal was defined subjectively, and may have arisen from incomplete data.*
MiHOS-SA

- Sensitivity Analysis (SA) module for MiHOS
- Evaluates effect of uncertainty of inputs on the outputs of the model, changing one or more inputs at a time randomly within analyst-defined range and evaluating the effect
- Evaluates fitness of candidates in presence of uncertainty in input values
MiHOS-SA Case Study

- Robustness of rankings for COTS used in the case study of MiHOS was evaluated with respect to changes in number of work hours and the budget allocated to mismatch resolution.
- In this particular experiment, application of MiHOS-SA confirmed the original ranking.
- SA component allows identification of mismatches that must be resolved for a COTS to satisfy mandatory requirements, and identification of minimum required resources to increase fitness of a COTS to acceptable level.
Outline

1. Motivation
   - Preliminaries
   - RE for OTS Details

2. Examples of Work on OTS Configurations
   - OTS Configuration and Adaptation
   - RE for COTS Software Stack
COTS for High Availability

- Automated selection of software stack configurations for SaaS environments
- Experiment-based method for selection of COTS components and configurations
- Rejects components that would prevent Service Level Agreements (SLA) from being met at prototype stage
- Evaluates time to recover the complete application stack from a failure
  - Time to transfer control to the node without failure
  - Time to recover data to the point of failure
  - Time to restart the failed process and recover the data
Summary

- While a lot of work was done in selection of OTS for a task, none generic enough to apply to all scenarios
- OTS selection is frequently considered the last step in RE, however, the choice of OTS affects requirements for the system and should be part of the iterative RE process
D. Galorath

*Software Reuse and Commercial Off-the-Shelf Software.*
Galorath Incorporation, El Segundo, CA, USA. 2005
Retrieved on 29 June 2015 from

A. Mohamed, G. Ruhe, A. Eberlein

MiHOS: an approach to support handling the mismatches between system requirements and COTS products
A. Mohamed, G. Ruhe, A. Eberlein
Sensitivity analysis in the process of COTS mismatch-handling

P. Pourali, F. Khendek, M. Toeroe
Enhanced Configuration Generation Approach for Highly Available COTS Based Systems
*2014 9th International Conference on Availability, Reliability and Security*
A. Khaled, M. A. J. Idrissi
A Semi-Structured Tailoring-Driven Approach for ERP Selection