

Defining a Quality Model for Mail Servers

Juan P. Carvallo, Xavier Franch, Carme Quer
Universitat Politècnica de Catalunya (UPC)
Barcelona (Catalonia, Spain)

GESSE group: <http://www.lsi.upc.es/~gessi/>

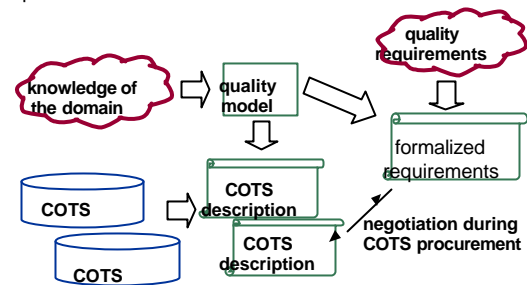
ICCBSS'03
Ottawa (Canada), February 2003

Introduction

Context: Software development with COTS components

- Activity: COTS procurement: focus on quality
- Success factor: Deep-enough knowledge about the COTS market

Objective: Providing means to obtain exhaustive and structured COTS descriptions of components belonging to concrete COTS domains as well as requirements



Software Quality Model: Definition

"The set of characteristics and the relationships between them which provide the basis for specifying quality requirements and evaluating quality"

ISO/IEC 14598-1, Software product evaluation – Part 1: General overview. 1999.

Elements of a generic quality model

- product properties that influence quality
- a set of high-level quality attributes
- a means of linking them

Dromey. "Concerning the Chimera". IEEE Software 13(1), 1996

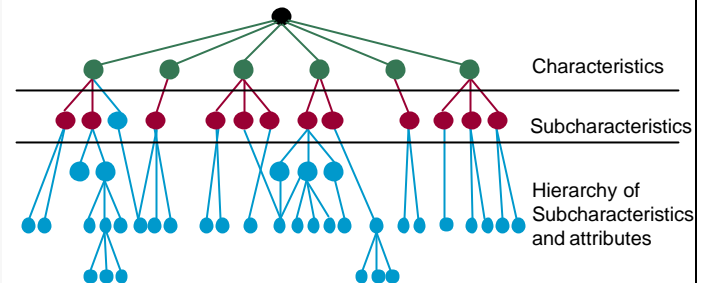
- metrics

3

ISO/IEC 9126-1 Software Quality Model

Based on the ISO/IEC 9126-1 quality standard:

- Main idea: definition of a quality model and its use as a framework for software evaluation
- General and customizable; hierarchical; allows overlapping; widespread



Quality Entities = Characteristics + Subcharacteristics + Attributes

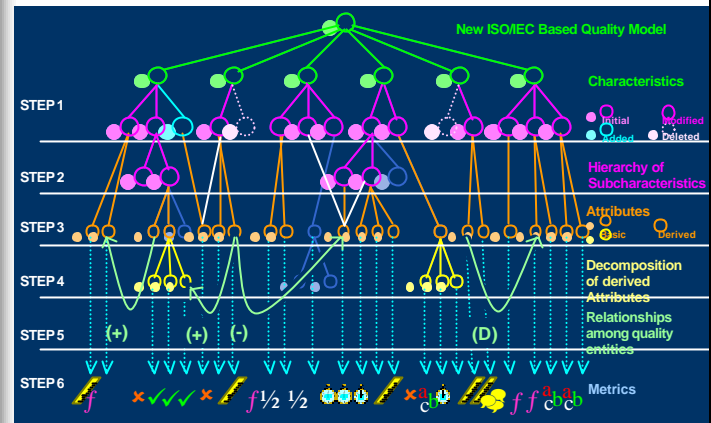
4

Predefined Quality Entities

| Characteristic | Subcharacteristics |
|-----------------|--|
| Functionality | suitability, accuracy, interoperability, security, functionality compliance |
| Reliability | maturity, fault tolerance, recoverability, reliability compliance |
| Usability | understandability, learnability, operability, attractiveness, usability compliance |
| Efficiency | time behavior, resource behavior, efficiency compliance |
| Maintainability | analyzability, changeability, stability, testability, maintainability compliance |
| Portability | adaptability, installability, co-existence, replaceability, portability compliance |

5

A Methodology for Building ISO/IEC 9126-1-based Quality Models



Details: (Franch & Carvallo 2003)

6

Why Mail Servers Domain?

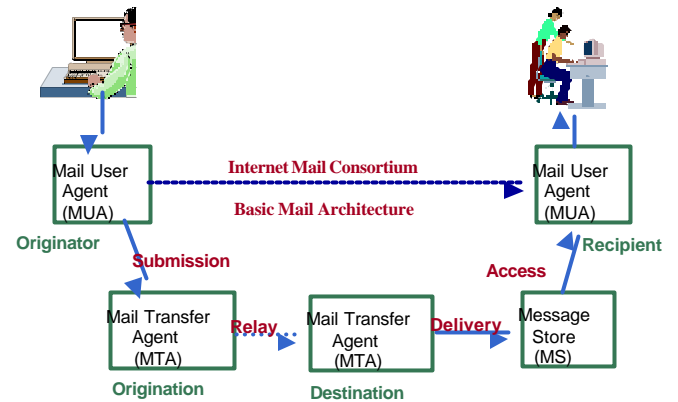
Some interesting points:

- needed by a huge number of companies & organizations
- great impact on companies & organizations
- time-consuming task; costly renovation
- lots of quality features to explore
- lots of products in the market

Preliminary analysis of the domain needed

7

The Mail Servers Domain



8

A Conceptual Model for the Mail Servers Domain

The description of the domain using UML diagrams provide us:

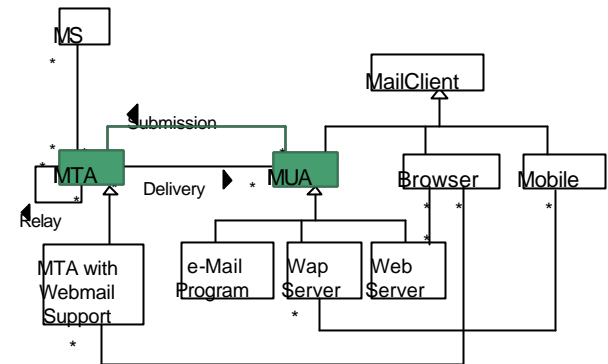
- the description itself
- a better understanding of the domain
- detection and resolution of unexpected misunderstandings

Two kind of diagrams:

- **Mail Server Context Diagram** that makes explicit which other types of COTS interact with Mail Servers
- **Mail Server Domain Diagram** that models the domain of Mail Servers

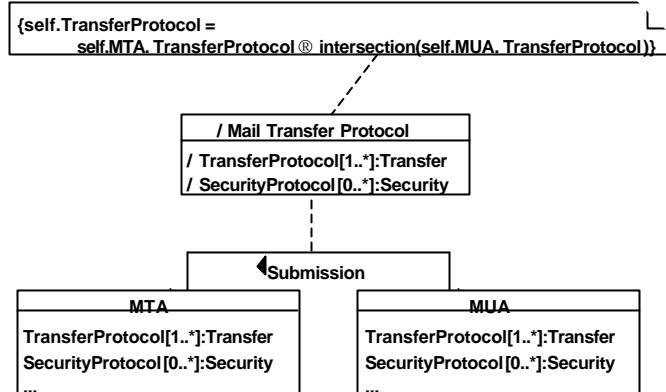
9

A Mail Server Context Diagram



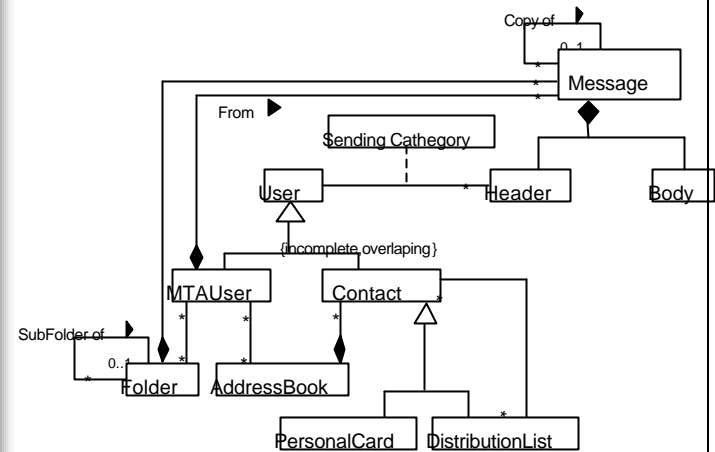
10

A Mail Server Context Diagram



11

A Mail Server Domain Diagram



12



The Mail Servers Quality Model: Step 1 Determining Quality Subcharacteristics

ISO/IEC 9126-1 characteristics and subcharacteristics as they are

13



The Mail Servers Quality Model: Step 2 Defining a hierarchy of subcharacteristics

Motivation: find groups of related quality features

Some new subcharacteristics have been identified and organized into a hierarchy

Examples:

- *Suitability* split into *Basic Suitability* and *Additional Suitability*
 - ◆ *Basic Suitability* split into *Folder Management*, *Mail Management*, ...
 - ◆ *Additional Suitability* split into *Video Conference Support*, ...
- *Operability* split into *Administration Operability* and *User Operability*

14



The Mail Servers Quality Model: Step 3 Decomposing Subcharacteristics into Attributes

Don't try to find every attribute in the world; focus on the most relevant ones

Information difficult to find and sometimes confuse or unreliable -> attributes provide a common reference model

Some issues:

- Lexical mismatches:
Distribution Lists = Group Document in Directory
- Attribute multidimensionality:
Message Tracking & Monitoring belongs to Accuracy and Analyzability

15



The Mail Servers Quality Model: Step 4 Decomposing Derived Attributes into Basic

Up to quantitative, product-feature attributes

Lower attributes must be atomic (if model is complete)

Examples:

- *Time Efficiency* split into *Message Throughput* and *Response Time*
- *Security* -> *Filters* split into *Host Filters* and *Message Filters*
- *Operability* -> *Administration Operability* -> *Message Delivery* split into *Maximum Delivery Retries* and *Time Between Delivery Retries*

16



The Mail Servers Quality Model: Step 5 Stating Relationships among Quality Entities

Information relevant for understanding the domain and for providing more guidance during procurement

Different kinds of relationships:

- *Dependency*: e.g. *Certification System* required for *Secure Protocol*
- *Collaboration*: e.g. *Kind of Store* may collaborate with *Average Response Time* (e.g. *Single Copy Store*)
- *Collisions*: e.g. *Kind of Backup* may collide with *Average Response Time* (e.g. *Incremental Online Backup*)

Relationships may be used to expand requirements and detect conflicts and synergies among them

17



The Mail Servers Quality Model: Step 6 Determining Metrics for Basic Attributes

A very complex process if maximum detail is required

Information difficult to find and sometimes confuse

Examples:

- Simple: *Subfolder Management* (bool), *Maximum Account Size* (integer), *Protocols Supported* (enumeration)
- More elaborated: *Default Folders* (set)
- Complex: *Message Throughput* (depends on platform and organization)

18

Use of quality models for describing requirements

From informal requirements to quality-model-oriented ones:

- use the quality model to translate to a known framework

Some common situations:

- direct mapping
"Spanish language support"
- expert guidance
"Support for the most commonly used certification standard" ← X.509
- ambiguity
"Support for accessing the server from other applications" ← which ones?
"Filter messages with viruses" ← and what to do with them?
- mixture of functionalities
e.g., about delivery functionality, operability and efficiency
- ill-formulation
"Transmission time less than 1 min. for messages without attachments"
← online attachments? long listings?
- incompleteness
"Average response time less than 1 min. and transmission rate less than 5 min. per Mbyte" ← required platform support? organizational constraints?

Conclusions (1)

A quality-model-based approach for describing and evaluation COTS components, with reliability, efficiency, understandability and return of investment in mind

The use of UML class diagrams has:

- Ameliorated the identification of quality attributes
- Allowed rearranging the hierarchy of attributes originally identified in a more harmonious way

Other uses of UML to explore (e.g., identification and testing of interaction points)



Conclusions (2)

The approach pays off in domains as the one of mail servers in which:

- COTS need to be present in a lot of companies and organizations
- The selection process is not trivial
- There is a great deal of available COTS

Its use in COTS procurement improves the process reliability, since:

- Allows having uniform descriptions of COTS in a domain
- Facilitates comparing these COTS
- Supports rewriting of quality requirements in terms of the quality model
- Suitable approach for tool support

21

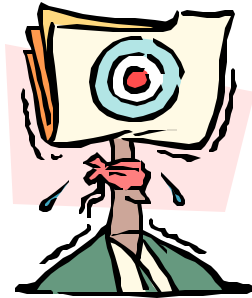


Current and Further Work

- Study which other UML diagrams can help in understanding COTS domains (i.e. use cases, sequence diagrams)
- Use of agent-oriented models (i^*) to find dependencies between components in different domains
- Reuse parts of quality models through different domains
- Incorporate managerial, marketing, etc., attributes into an ISO/IEC style (vendor viability, cost, ...)

22

Any question?



Contact: Xavier Franch, franch@lsi.upc.es
GESSI homepage: <http://www.lsi.upc.es/~gessi>

23