Building Sound COTS Products Taxonomies

Juan P. Carvallo[1], Xavier Franch[1], Carme Quer[1], Marco Torchiano[2]

[1] Universitat Politècnica de Catalunya
UPC-Campus Nord (C6)
08034 Barcelona, Catalunya, Spain
{carvallo, franch, cquer}@lsi.upc.es
http://www.lsi.upc.es/~gessi

[2] Politecnico di Torino
C.so Duca degli Abruzzi, 24
10129 Torino, Italy
torchiano@polito.it
http://softeng.polito.it/torchiano

Introduction

The acquisition of process of COTS products requires the identification of the suitable products.
The evaluation of COTS products varies from type to type, e.g. a DBMS has to be evaluated differently from a web browser.
The techniques used to integrate the COTS products are strongly dependent on the type of products. The same applies to the architectural patterns and style used to build COTS-based systems.
So it is important to be able to classify COTS products into homogeneous classes. Taxonomies are a tool to perform such a classification.

Motivation

The first question we should ask is: why building taxonomies at all?
The purpose of taxonomies is to provide a classification mechanism; they allow distinguishing different categories of COTS products. It is important to provide a clear rationale and structure for the classification. This is required to make it easy both to understand and apply.
The hierarchical organization of taxonomies defines clear classification rules, which provide the rationale for the classification. The result is a consistent partitioning of COTS products into classes.

A basic and obvious goal of the classification of COTS products is to select the suitable products when we need to build a COTS-based system. The structure of the taxonomy drives the developers during such a selection activity. Following the branches of the taxonomy it is possible to navigate the taxonomy until the suitable class of products is found.

In addition, since within a consistent taxonomy each class contains similar products, it is possible to enhance the taxonomy associating a set of features to each class. Typical examples of features that can be linked to classes are:

- a collection of typical issues for that class [4],
- a toolbox of techniques and methods suitable for that class,
- quality models [5][6].

The above are just examples of features that can be attached to a class of COTS products; many more features can be added.

Taxonomies

Usually a taxonomy is represented as a tree-like hierarchical structure.
We propose to implement a taxonomy, together with its hierarchical structure and classification rules, as a decision tree [2] [3].
The categories and sub-categories are represented by the nodes of the nodes of the tree, being the leaves atomic classes, not further divided.
The classification rules are expressed by means of attributes. An attribute is associated to each node; for each possible value of the attribute (or for each subset of a given partition of the possible values) a new child category is introduced.
The evaluation of the attributes can be carried on formulating a question to which correspond several answers (i.e. the single values or subsets of values).
In addition we attach to each category a generic feature set specific for that category.

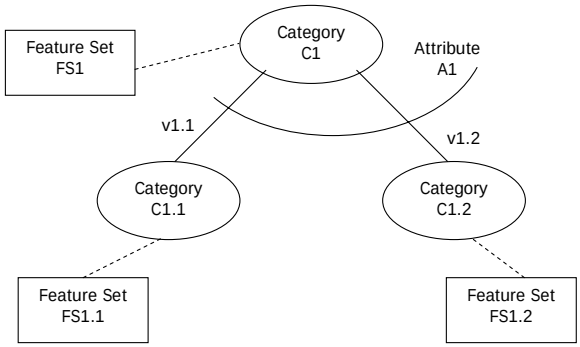

Fig. 2.1. The fundamental elements of a taxonomy.

The advantages of providing a decision tree based taxonomy of COTS products categories are:

- a formal and unambiguous definition of classes,
- a "guided" way to identify the categories,
- the capability to reuse features among categories.

The attributes define how to measure them and the possible values, the partition in sub-categories is formally defined on the basis of such attributes.
Since a question can be associated with each attribute evaluation, the process of applying the classification or looking for a classified element in the taxonomy is made simple.
The hierarchical structure of the taxonomy allows the definition of generic features in a category, which can be reused and refined in the sub-categories. Actually an inheritance mechanism is applied here: the feature set of a category inherits the feature set of its father that can be extended and refined.

The above properties hold only in presence of a valid taxonomy. Actually it is easy to make mistakes in the definition of taxonomies. In addition we must produce compact and reasonable taxonomies.
For these reasons we define the criteria for the validity of taxonomies:

- a category can be partitioned into sub-categories based on the values of an attribute,
- each attribute must be measurable answering to a question,
- the feature sets associated with each sub-category must differ from each other.

If the partition of a category into sub-categories is linked to the values of an attribute is easy to formulate a question. This gives the possibility of identifying the relevant category by a series of questions and answers.
Each attribute must be associated with a question, that is easy to answer, and must provide a set of possible values. For each of the possible values there is a sub-category.
If a partition of a subcategory does not enrich the feature set then it only adds "noise": it makes the taxonomy more complex and adds a question that has no purpose. We apply the Occam's razor and remove the categories that are not clearly useful.

Building taxonomies

To build a COTS taxonomy it is possible to proceed in several different ways:

- restructure existing application taxonomies (e.g. [1]),
- top-down definition from abstract principles,
- bottom-up construction from actual COTS products.

Several taxonomies can be found either in the literature, on the web, or from consulting firms. Thus it is reasonable to work on them to produce a taxonomy that conforms to the criteria defined in the previous section.

The key point in the definition of taxonomies is the definition of the attributes. A process to carry on this task has not been defined yet. We have to work using common sense and trying to minimize the differences from the original taxonomy.
In [1] an example of such a restructuring is presented. Here we summarize briefly the main differences among the original and the resulting taxonomies.

- Identification of new scopes. This is the most usual action we have taken, due to the nature of our activity.
- Division of existing scopes. Occasionally, some original scopes were too coarse-grained mixing different concepts belonging to different attributes.
- Merge of existing scopes. The other way round, some scopes are so similar that their differentiation does not really make sense.
- Removal of existing scopes. The main reasons being: the scope is not a software domain and the scope does not really add value to the taxonomy.

Conclusions

We believe that taxonomies based on decision trees constitute a simple and usable tool for classifying COTS products.
In addition they allow the definition of feature sets that can be build incrementally as we go down the hierarchy.
Examples of features are quality models that were defined in [1], but other can be defined.
This is a primary motivation for building classification of COTS products. This motivation is less subject to change when moving from a COTS-based application to another.
As a further work we see:

- the definition of wider and more reusable taxonomies,
- the definition of more complete feature sets, and

- the definition of a process to produce valid taxonomies from existing non-formal application taxonomies.

References

1. Juan P. Carvallo, Xavier Franch, Carme Quer, Marco Torchiano. "Characterization of a Taxonomy for Business Applications and the Relationships among them" In Proceedings of Third International Conference on COTS Based Software Systems (ICCBBS), Redondo Beach (CA), February 1-4, 2004
2. J.R.Quinlan. “Learning efficient classification procedures and their application to chess endgames” in Machine Learning: An Artificial Intelligence Approach, R.S.Michalski, J.G.Carbonell, and T.M. Mitchell (Eds.), Palo Alto, Tioga Publishing Compani, 1983.
3. J.R. Quinlan. “Learning decision tree classifier” in ACM Computing Surveys 28(1), March 1996.
4. M. Morisio, M. Torchiano. "Definition and classification of COTS: a proposal". 1st International Conference on COTS-Based Software Systems (ICCBSS), LNCS 2255, Orlando (USA), 2002
5. X. Franch, J.P. Carvallo. "Using Quality Models in Software Package Selection". IEEE Software, 20(1), January/February 2003, pp. 34-41.
6. J.P. Carvallo, X. Franch, C. Quer, "Defining a Quality Model for Mail Servers". 2nd International Conference on COTS-Based Software Systems (ICCBSS), LNCS 2580, Ottawa (Canada), 2003.