

Introduction of BEMES, a Webtool to Simplify Business Process and Equipment Cost Modelling

Jonathan Spruytte, Marlies Van der Wee, Sofie Verbrugge and Didier Colle
*Department of Information Technology, Ghent University – iMinds, iGent-Toren,
Technologiepark-Zwijnaarde 15, B-9052 Gent, Belgium
{jonathan.spruytte, marlies.vanderwee, sofie.verbrugge, didier.colle}@intec.ugent.be*

Keywords: Techno-economic, Equipment Cost Modelling, ECMN, Process Cost Modelling, Business Process, BPMN.

Abstract: Modelling both business processes and equipment costs is an elementary step in the techno-economic evaluation of a business case (infrastructure project, network deployment, etc.). Most of a upfront business case evaluation, on which strategic decisions are based, relies on inflexible and error-prone spreadsheet models. In order to provide a stable and reliable alternative, this paper presents the equipment coupling modelling notation (ECMN). ECMN allows calculating the costs for equipment installation using dynamic, easy-to-understand, graphical equipment trees. For business processes, we developed an adapted version of the business process modelling notation (BPMN). In order to simplify the usage of both ECMN and BPMN, we also present the BEMES-tool (Business Modelling and Simulation). BEMES is a graphical web-based tool that can be used to draw both models using drag-and-drop and calculate the models resulting in visual output. The BEMES-tool consists of a set of interlinked modules; we describe the functionality of each of these modules and discuss the benefits this strong modularized approach yields for both the current and future features.

1 INTRODUCTION

The techno-economic evaluation of a network deployment planning problem can be split in 4 major components: scope, model, evaluate and refine as discussed extensively in (Verbrugge, 2009). In the modelling step, the input collected during the scope phase is used to model both costs and revenues. These models serve as input for the evaluate phase, which combines the outcome of both business process and equipment cost models with revenue estimates to evaluate the business case for one or multiple stakeholders. The fourth and final phase, refine, applies game theory, real options and sensitivity analysis to capture the impact of uncertainty. The techno-economic research group at Ghent University (www.technoeconomics.ugent.be) has developed several Java-based tools to automate (parts of) these modelling phases (Tahon, 2014), (Kasier, 2009).

Within the context of this paper, two modelling languages which simplify cost modelling will be discussed; the distinction is made between models for equipment and models for processes.

For process modelling, the well-known Business Process Modelling Notation (BPMN) can be used

(White, 2004). BPMN uses a flowchart-like notation to model the sequence of actions and decisions in a business process; an example is provided in Fig. 1.

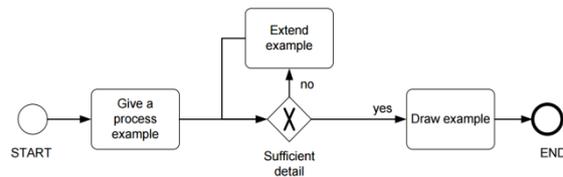


Figure 1: A basic process with 1 decision and 3 actions modelled in BPMN.

A lot of planning projects require reliable estimations of the Bill of Material (BOM) for all equipment that needs to be installed (e.g. installation of a central office, dimensioning of a data center, etc.). The BOM lists all equipment and offers a view of the total equipment cost of a planning project. There however is no formal, standardized approach to equipment modelling (to draft a BOM) so far. A typical equipment cost model approach starts from a hierarchical equipment tree, interlinking equipment (e.g. each server needs a slot in a server rack). This approach is often based upon a large amount of sheets

in your favourite spreadsheet software package and may or may not include additional tools (Oase, 2011). This approach is not only time-consuming, it also offers little flexibility and makes calculating the same model for a variety of input a tedious job. For tackling these issues, we have developed the Equipment Coupling Modelling Notation (ECMN) (Casier, 2014).

Using ECMN, it is possible to graphically draw the equipment tree, add granularities to each branch in the tree and link cost-drivers to each piece of equipment (Van der Wee, 2012). A cost-driver is a time-dependent function, which serves as an input for the model and can be linked to any element of a model.

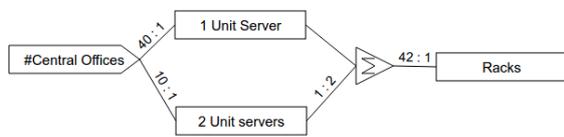


Figure 2: An example ECMN-model consisting of 1 cost driver and 3 pieces of equipment.

Figure 2 shows a basic ECMN-model which consists of a total of 5 elements, from left to right:

- *#Central Offices*: is a driver and is the only input considered.
- *1 Unit Server* and *2 Unit Servers* are as the name indicates servers that respectively require 1 and 2 units in a rack. As the links (granularities) indicate, one central office requires 40 1-Unit Servers and 10 2-Unit Servers.
- A *summation*, which sums the total of required rack-space of both types of servers. The granularity of the links is respectively 1:1 and 1:2 as the 2-Unit Servers requires two spaces in a rack.
- *Racks* is the last piece of equipment; one full-sized rack typically has 42 units, so the granularity of the link has been set accordingly. (Note: the resulting amount of a link is always round to the next integer, if the start of the link is 5 with a granularity of 3, the end of the link will be 2. This fits within the equipment installation reasoning: as soon as you have an extra server that no longer fits in the first rack, you have to install a second rack).

When giving a value to the driver (*#Central Offices*), the quantity of both types of servers and number of racks is automatically calculated with the corresponding cost. A number of additional parameters (e.g. reinstallation period) can be provided which will influence the total cost.

In order to easily create both ECMN and BPMN models, we have created the BEMES-tool which is further discussed in this publication. The remainder of this publication is structured as following: in section 2 we discuss the modular approach of the tool and the benefits it yields. In section 3 we introduce a public test version of the tool, which allows everyone to try out the BEMES-tool. Finally, section 4 briefly summarizes this paper and presents further steps for the BEMES-tool.

2 THE BEMES-TOOL

In order to incorporate both ECMN and BPMN in the modelling phase we developed a graphical web-tool: BEMES (Business Modelling and Simulation). The BEMES-tool allows us to create both ECMN and BPMN models using simple drag-and-drop-actions and consists of three interlinked modules (Figure 3): (1) the graphical web-based frontend (the editor, which offers separate views for ECMN and BPMN), (2) the repository that stores the models and (3) the calculator hub linked to a set of calculators that perform the actual calculations of the models and return the result.

Between these modules, the models are exchanged in XML-format using the REST-protocol.

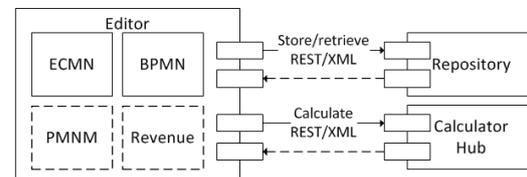


Figure 3: The BEMES-tool has a modular approach. Interaction between the modules uses the REST-protocol and the XML-format.

As an addition to these modules, a Java-interface has been created. This interface can retrieve, extend/modify and calculate models via the Java programming language. In the next paragraphs, the functionality of the modules is summarized and the modular approach of the BEMES-tool is discussed.

2.1 Editor

The first module is the editor which provides the graphical web user interface and interlinks behind the scenes with the Repository (2.2) and the Calculator Hub (2.3).

After authentication, a user has the ability to switch between a number of views, depending on the

wanted type of model (ECMN or BPMN). Using the editor, a user can draw an entire model using drag and drop, change parameters of objects, calculate a model and get visual output without having to write one single formula or write any code.

2.2 Repository

The repository is responsible for the storage of - and the access to - the different models. The current version of the repository stores the models directly, in XML-format, on disk. Since at the moment only one repository is used and models are directly linked to a single user, using a file-based storage entails no problems. In the future, once models may be shared between users and concurrent edits may be possible, any kind of database should be used to simplify the synchronisation process.

2.3 Calculator Hub & Calculators

The Calculator hub is the single point of entry to calculate models. The calculator hub receives a request to calculate a model (represented in XML). Based upon the type of model, it selects a matching calculator and forwards the request, as shown in Figure 4.

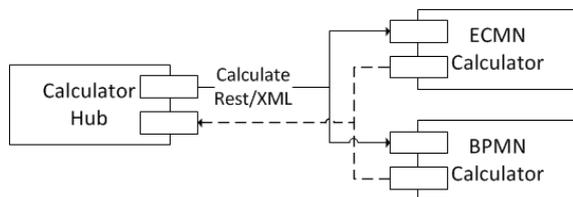


Figure 4: The Calculator Hub links to a set of calculators.

This single point of entry approach has a number of benefits.

- **Single Point of Control:** since any model passes through the Calculator hub, it is the perfect location to validate the incoming XML-stream.
- **Flexible and Extendable:** based upon the incoming XML-stream, the Calculator hub selects the correct calculator. At the moment, only one calculator per type is deployed, so this forwarding is straightforward. In the future, multiple calculators per type may be deployed and the Calculator hub may be used for load balancing.
- **Simplified Backward Compatibility:** instead of having calculators that support every version of the XML-format, they should only support one (major) version. This makes the implementation of each calculator less boated and thus much

simpler. When changing to a new major version of the XML-format, the old calculators which are based upon the older format can simply stay online. When a request is made to the Calculator hub, it will select a matching calculator based upon the type and (major) version of the model.

- **Access Control:** the Calculator hub is the perfect location to implement any rate limits (e.g. number of model calculations per hour) or may in the future be used to limit access to the calculation of some model types (e.g. licensed access to modules).

2.4 Java-to-BEMES-Interface

The last module, the Java-to-BEMES-interface allows access to models designed in the editor via the Java programming language. Using the Java-to-BEMES-interface, a user can load a model, override model values (e.g. the cost of an equipment item) and calculate the modified model. To do so the, the interface links directly to both the repository and the calculator hub, in a similar fashion as the Editor, as shown in Figure 5.

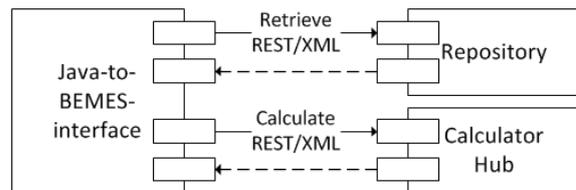


Figure 5: The Java-to-BEMES-interface allows the retrieval, modification and calculation of a model drawn in the Editor.

While the graphical user interface is great for building the model and receiving basic output, the Java-interface is great for running a model with a variety of input values e.g. sensitivity analysis or scenario analysis. Note that, currently, this Java-interface is limited to ECMN-models.

2.5 Benefits of the Modular Approach

As said, the BEMES-tool consists of multiple modules; this clear division yields a number of merits:

- Modules can easily be distributed across multiple servers; this way, modules can easily be duplicated and accessed using any load balancing scheme (e.g. round-robin).
- Modules can easily be re-implemented, as long as the existing REST-interface is respected (e.g. a

new repository which stores models in any kind of database).

- Any module can be considered as a standalone-module for development. As long as the XML-format is respected, any module can be changed/extended without any impact for the other modules.
- Any additional modules can be linked via the calculator-hub with minimal effort.

3 TEST CASE: LIVE DEMO

To show the possibilities of the BEMES-tool, a live demo has been made available at <http://bemes.atlantis.ugent.be/EditorDemo>. A guest-account has been created, to log in use guest for both the username and password.

When logging in, the demo-model as discussed in paragraph 1, should automatically be loaded, if not the model can be loaded using the navigation bar on top. Using this demo the functionality of the BEMES-tool can be tested; it allows the creation of both ECMN and BPMN models. In this demo-version, saving a model to the repository has been disabled.

4 SUMMARY AND FUTURE WORK

Modelling equipment and process costs is an important part of a project business case evaluation (e.g. network deployment, data center dimensioning, etc.). BPMN is the standard language for process modelling, but equipment models are often made using spreadsheets which are error-prone and rather inflexible to calculate for a wide variety of values. In order to simplify this process, we have developed the Equipment Coupling Modelling Notation (ECMN) which is a graphical notation which allows the modelling of an equipment tree. In order to incorporate ECMN in the modelling process we have created BEMES. BEMES (Business Modelling and Simulation) is a graphic webtool specifically developed to draw both ECMN and BPMN models using very simple drag and drop-actions.

Internally BEMES has a strong modular approach resulting in added flexibility and extendibility. Future work includes further development of the Java-to-BEMES interface for BPMN, and inclusion of automatic sensitivity analysis for both type of models.

Apart from BPMN and ECMN, the techno-economic group is also developing models for

network infrastructure deployment cost (PNMN) as well as revenue modelling based on different types of pricing strategies. The goal is to incorporate these models in the BEMES-tool, so the tool can be used to model each part of a (network deployment) business case evaluation.

REFERENCES

- Verbrugge, S. et al, 2009. White paper: Practical steps in techno-economic evaluation of network deployment planning.
- Tahon, M. et al., 2014. Real options in telecom infrastructure projects - A tutorial. *IEEE Communications Surveys and Tutorials*, 16(2), pp.1157–1173.
- Casier, K, 2009. Techno-Economic Evaluation of a Next Generation Access Network Deployment in a Competitive Setting. Ph.D thesis, Ghent University.
- White, S. a, 2004. Introduction to BPMN. *BPTrends*, pp.1–11.
- Van der Wee, M. et al., 2012. A modular and hierarchically structured techno-economic model for FTTH deployments. 16th international conference on Optical Networking Design and Modeling, *Proceedings*, (1), pp.1–6.
- OASE, 2011. Deliverable 5.1 – Overview of tools and methods. <http://www.ict-oase.eu/index.php?page=120&>