Softw Syst Model (2014) 13:457–458
DOI 10.1007/s10270-014-0409-1

EDITORIAL

Modeling big smart data

Robert France · Bernhard Rumpe

Published online: 3 April 2014

Large volumes of data are generated continuously by billions of human data producers, sensors, surveillance systems, communication devices and networks (e.g., the Internet). Proper analysis of this data can lead to new scientific insights, new products and services, more creative outputs (e.g., new recipes, music scores, fashion styles), improved performance of business and civic organizations, and to better informed government and non-government organizations. In other words, deriving information from these large volumes of data can lead to, among other things, smarter individuals capable of making scientific breakthroughs, producing innovative products, and making effective decisions.

Data can be well structured or not. We have observed that the term "semistructured" data is also used in cases where the structure of the data is not yet known or is overly complex (e.g., the structure of natural language). One of the challenges facing the big data community relates to inferring the structure behind the data when it is not known beforehand. In terms of modeling, this challenge relates to inferring a data model from a set of data. The challenge arises because there may be more than one way to structure data, only some of which may be based on inherent data properties. Deriving structures that facilitate human understanding or appropriate computer manipulation may require considerations beyond inherent data properties. For example in Grounded Theory, analysis involves mapping data to "ideas" and "ideas to ideas".

In the context of object-oriented modeling we might ask: What is the class diagram that adequately models a given set of data expressed as object diagrams? This would involve identifying classification hierarchies for data elements as well as relationships among the classifiers and their cardinalities. In the software language domain, the related problem is deriving a metamodel given a set of models.

Big sets of data do not necessarily mean that the data model is also complex. Complexity of the data model and the size of the associated data set seem to be relatively unrelated. For example, some security agencies collect petabytes of data related to tens of relationships between individuals. Sensor data are often voluminous, but the data have pretty simple data structures. We observe that pictures and sound data structures are complex because of compression information. Business models, on the other hand, sometimes have thousands of related entities, where many of the entities have few instances.

While some relationships between data elements can be captured by class associations, others might require more complex descriptions. In the context of the UML, the Object Constraint Language (OCL) can be used to describe some of these more complex kinds of dependencies. Therefore from a modeling point of view, a second significant issue related to smart use of data is the identification of OCL constraints representing complex relationships from data sets. Deriving logical dependencies between data entities is an area of active research, and we expect to see some good results from the data mining domain, despite the theoretical undecidable/exponential issues.

It is interesting to note that in software engineering data models are developed and data conforming to the models are produced, while in data mining, the data comes first and the challenge is to infer suitable data models. That is, in the smart data domain, the data sets are given and the existing (generic) system tries to construct a data model from that data. However, this is not the full story: smartness, to

R. France
Colorado State University, Fort Collins, Colorado, USA
e-mail: france@cs.colostate.edu

B. Rumpe (✉)
RWTH Aachen University, Aachen, Germany
e-mail: Bernhard.Rumpe@sosym.org

a large extent, comes from the understanding and later use of probabilities for dependencies. Probabilistic models are not very much used in software development, except when we talk about reliability and robustness of dependable systems. May be it would be interesting to take probabilistic concerns in consideration when producing data models before developing the system. For example, this could be a way we could collect and resolve conflicting requirements.

In smart data, the word "metadata" is used quite often. In current literature, it has a number of interpretations, but in general it refers to information about the data. In the communication domain (e.g., telecommunication, internet) metadata includes transmission time, receiver, sender, type, and size of the transmitted data. For data records (e.g., medical data), it includes information on who produced the record, who is allowed to read it, what modifications are allowed, and how long the data is valid. Digital rights management for media is a sophisticated form of metadata. In sensor domains (e.g., building heating, plant engineering), we are interested in, for example, quality of the sensor data or precise time of measurement. Metadata, therefore, plays an equally important role in the data domain as meta-modeling plays in the modeling domain. Interestingly, these technologies are intertwined, as there seems to be common agreement that the data structure description (model of the data) is a form of metadata as well. Furthermore, metadata itself has a data structure that needs to be modeled. It may be the case that this model may at least be partially derived from available metadata. When we look at XML (the ASCII of the new century), we can observe that sometimes data itself carries parts of its metamodel. Relationships between data, metadata, models of data, and metamodels of metadata still need to be explored in more detail.

An often implicit assumption is that data can be modeled by class diagrams, entity/relationship diagrams, or ontologies. While this may be the case in principle, it is may be the case that domain specific languages (DSLs) are more appropriate for some kinds of data. Picture or video encodings are DSLs used for efficiency. Building control systems have proprietary DSLs for sensor data.

Many forms of data, especially sensor data or data about business performance comes in time series. Learning the execution models behind this kind of data is challenging. The software engineering/modeling community provides quite sophisticated modeling techniques, like state machines, activity diagrams, Petri nets, sequence diagrams, or business process models that may be useful in tackling this challenge.

The modeling and data domain are closely related, and we would very much like to see synergies further develop between the modeling and the data communities. We do have a common history with the database, E/R and conceptual modeling communities. We hope that this editorial will inspire some of you to build bridges to the smart data community.

Announcement: a change in the SoSyM organization

Before we describe the contents of this issue, we have an important announcement to make. We have reorganized the Editorial Board to help us streamline the review process with the intent of further improving the average review time. The Editors-in-Chief will now be assisted by the three Associate Editors who will be responsible for vetting new submissions and assigning vetted submission to editors. The new Associate Editors are as follows:

- **Marsha Chechik**, University of Toronto, Canada
- **Martin Gogolla**, University of Bremen, Germany
- **Jean-Marc Jezequel**, University of Rennes, France

These individuals have been outstanding editors for a number of years, and we are happy that they have agreed to take on the responsibilities of Associate Editors. Please join us in welcoming the Associate Editors to the SoSyM team.

Content of this issue

As you probably have seen already, we have doubled up the number of pages this year as our journal feels strong enough to flush at least a larger part of its online backlog. We are pleased that SoSyM is in such a healthy state. This issue includes:

- A theme issue on "Model-Driven Service Engineering" organized by Juan Manuel Vara, Mike Papazoglou, and Il-Yeol Song with six papers.
- A theme issue "Models and Evolution" organized by Dalila Tamzalit, Bernhard Schätz, Alfonso Pierantonio, and Dirk Deridder with eight papers.
- Three regular papers.

Both theme issues cover pretty important and interesting topics: Services are the building blocks of the internet as well as the backbone of any modern intranet. Evolution of systems and their models is strongly needed for businesses staying innovative, allowing variants, and enhancements. The three regular papers talk on data collection resp. constraint solving in the context of software architectures and on synthesis techniques for model transformations.

We hope this issue inspires new and ongoing research and solutions.