A GENERAL-PURPOSE TOOL FOR DOCUMENTING DISTRIBUTED LABORATORY ASSAYS

Patrick Arnold, Oliver Kusche
Institute for Applied Computer Science, Forschungszentrum Karlsruhe, 76344 Eggenstein Leopoldshafen, Germany

Andreas Schmidt
Department of Computer Science and Business Information Systems, Moltkestrasse 30, 76133 Karlsruhe, Germany

Keywords: Ad-hoc workflows, LIMS system, activity coordination.

Abstract: In the context of a publicly funded cooperative project aimed at the assessment of potential risks related to nanoparticles, a general-purpose system for documenting the laboratory assays is presented. The system is designed to enable users distributed over various organizations to jointly document their activities, yielding a data base that allows for the reproducibility of the entire lifecyle of any given specimen. As requirements and activities are constantly evolving, the concept is focused on a flexible approach where new activity types can be added and configured at runtime.

1 INTRODUCTION

The *Nano Care* project funded by *BMBF* (Federal Ministry of Education and Research) is aimed at generating new scientific findings with respect to potential impacts of nanoparticles on man and the environment. The participating research centres, universities, and industry partners study the toxicity of various synthetic nanoparticles and develop highly innovative applications and measurement methods for a preventative and sustainable use of nanotechnology.

The *Institute for Applied Computer Science* supplies the information technology platform for this project. This platform shall ensure communication and knowledge transfer both within the project and to the public. The major component of this platform is a web-based application that offers comfortable data acquisition similar to a laboratory information management system (LIMS) in order to document the complete lifecycle of the nanoscaled materials studied under the project.

2 REQUIREMENTS

The system should have a high flexibility in meeting new and changing requirements made by the individual project partners, in particular as far as newly developed measurement methods are concerned. Other requirements result from the variety of laboratories that are spatially distributed over a number of enterprises and research institutions. The application shall be executed independently of the computer platform used by the user and access a central database. As the institutions compete with each other outside of this joint project, authorisation and authentication are big concerns and thus a mature and flexible role concept is required to restrict access to confidential information.

Research of already existing systems revealed that most of the necessary functions are covered by commercial and freely available software. However, no system was found that meets all requirements, as existing laboratory information management systems mostly are centred in enterprises. It was therefore decided to develop an own system. To the extent possible, existing components shall be applied to minimize the time needed for implementation, as these components have already been tested and, hence, are free of defects.

Moreover, an appropriate framework shall be used as the basic structure for development, which makes available solutions for general routine tasks and, hence, allows for the maximum attention being paid to the implementation of the business logic.

Arnold P., Kusche O. and Schmidt A. (2008).
A GENERAL-PURPOSE TOOL FOR DOCUMENTING DISTRIBUTED LABORATORY ASSAYS.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems*, pages 301-304
DOI: 10.5220/0001734303010304

3 CONCEPT

A generic approach was chosen, where no predefined workflow templates are given, but activities can be documented flexibly in the form of ad-hoc workflows with a few restrictions only, which can be formulated at runtime (e.g. by an administrator). In line with concepts known from LIMS, it is planned to design a measurement line to acquire and store the measurement values and a control line for assay management and data evaluation.

Modelling shall focus on the activity that represents a concrete process step in the measurement line, such as fabrication, separation, or characterization of a specimen. By linking these activities, various types of workflows can be generated. As already mentioned, there are no predefined workflows. They are created by user inputs during the runtime of the system.

3.1 Definition of Activity Types and their Properties

To generally describe an activity, an activity type is defined, which specifies the properties and behaviour of its later activity instances. A property is modelled by the entity *AttributeType* (Figure 1) and, for instance, contains information about the data type of this property (text, number, Boolean value, or others) and about whether it may assume multiple values or is mandatory. With *Unit*, it is possible to choose a unit for later values of this attribute type. In this way, an attribute type named *Temperature* with the data type *Number* be assigned the unit "$^{\circ}C$ ". If user input from a pre-defined list is required as opposed to input of arbitrary data for a certain attribute type, these values may be defined using the entity *DefaultValues*.

For a better handling, the attribute types may be grouped into categories that are represented by the entity *AttributeCategory*.

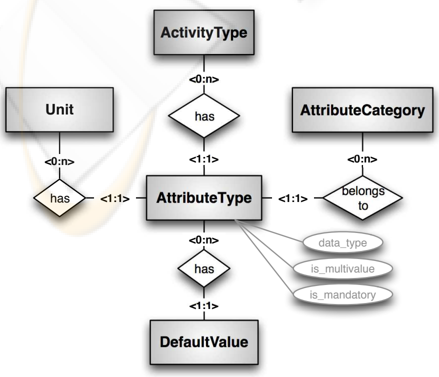

Figure 1: ER model for activity type definition.

3.2 Definition of Specimen Types

An activity in the measurement line always is carried out on an object, in this concrete project, a sample of synthetically produced nanoparticles, for instance. These are produced in larger amounts, but may be divided into any number of, also divisible, sub-sets during their lifecycle (Figure 2). The system allows for the definition of any specimen types, e.g. a hierarchical structure of batches and subordinate lots and assays.

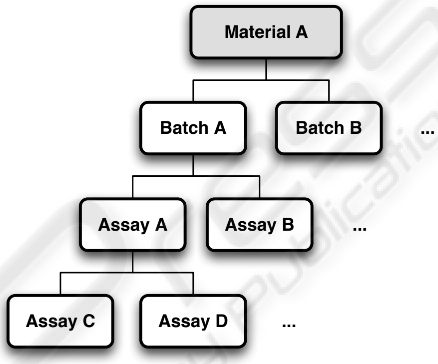

Figure 2: Division of a batch into assays.

3.3 Rules for Linking Activities

Using activity and specimen types, it is now possible to formulate rules for the linking of activities. Figure 3 shows the respective model. Depending on these rules, a set of possible activities is generated, which can be carried out on a certain specimen. From the point of view of application, such a rule may be as follows:

- Following the activity *Shipping*, an activity of *Acceptance of the good* is required, another successive activity is not considered.
- Shipping may only be accomplished for the specimen type of *Assay* or *Lot*, as a complete batch shall never be shipped.

3.4 Runtime

After having shown how the behaviour and the properties may be configured to create a workflow, it shall now be dealt with the modelling of the components required during runtime (Figure 4). These are real instances of specimens and activities as well as a relationship between the activity and the attribute type. The relationship represents the value (e.g. recorded measurement value) of a certain attribute type (e.g.

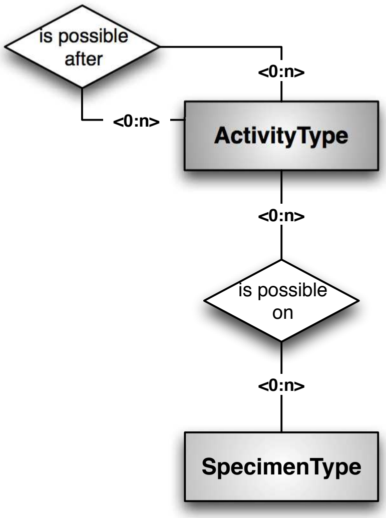

Figure 3: Rules for linking activities.

temperature) that is specified by the activity (e.g. characterization) using the specimen.

The input of the activity always is a specimen. The output may include one or several (in case of a separation) specimens. An exception is the initial activity which represents the starting point of the measurement line.

An activity may be provided with any number of attributes that have to correspond to a previously defined value range (string, decimal, selection list). In addition, one or several documents and/or images may be added to an attribute.

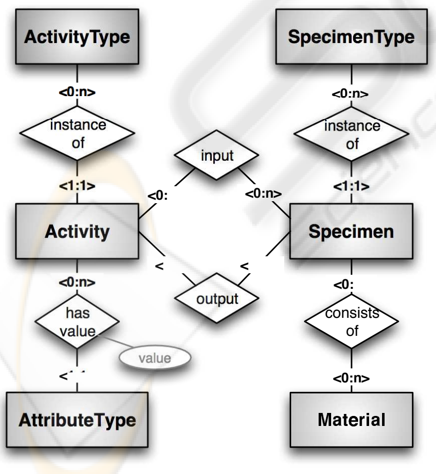

Figure 4: Model-metamodel-relation.

Since activity, specimen and attribute types are explicitly modelled as entities, new ones can be created and/or modified at runtime and thus new use cases can be addressed just by configuration without the need of modifying the application code. In this way, requirements occurring in the course of the project only in terms of new or existing activity types or their properties may be addressed easily e.g. by an administrator via a specialised back-end.

4 IMPLEMENTATION

As the users of the application are distributed over various sites and organisations, a web-based solution was chosen, which can be run nearly independently of the user platform. The software is executed on a web server within an Apache Tomcat servlet container (Brittain and Darwin, 2007) and can be accessed via the internet using a conventional web browser.

The basic framework of the multi-layer software architecture was generated with the help of the web framework *AppFuse* (Raible, 2007), which in turn heavily relies on the *Spring* (Walls and Breidenbach, 2007) and *Hibernate* (Bauer and King, 2006) frameworks. Its prototyping functionality allows to most rapidly implement requirements and to early and permanently verify features of the software by test-driven methods. The generated, but already executable artefacts are used as a basis of the implementation of the business logic. For managing row-level security, the *Spring Security framework* (Interface21, 2007) has been employed.

5 USE CASES

Figure 5 shows a typical case of application in a simplified manner. Partner *A* produces a batch of a material that shall be studied within the project. This is documented by the creation of the initial activity *Manufacture* in the system. Doing this, a unique batch number is given (BH-1). Of this batch, three samples shall be separated first, one of which shall be shipped for being investigated. The execution of the activity *Shipping* generates an e-mail to the recipient of the sample, partner *B*, who documents the receipt of the good in the system. For the characterization of the material, another sample is separated. The data obtained by the characterization are recorded as attributes of the activity *Characterization*. Then, the original sample is shipped further to partner *C*, where a biological investigation is made that is also documented as an activity with the respective attributes.

By recording all relevant activities, the corresponding attributes as well as the relationships of the specimens to each other, the lifecycle of each individ-

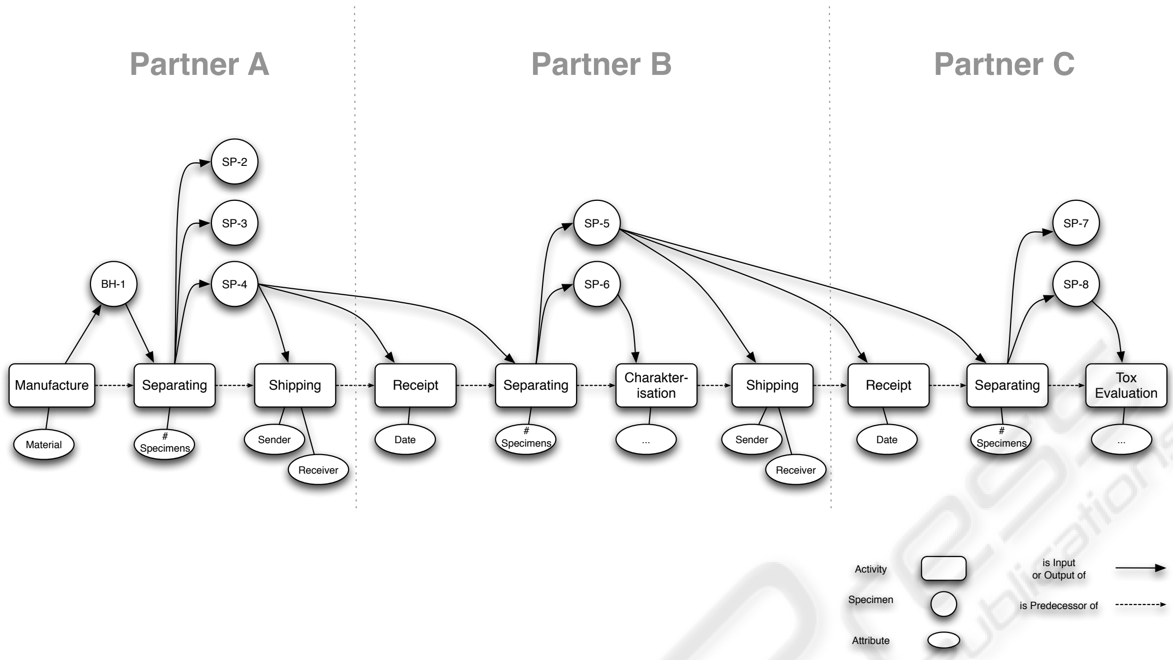

Figure 5: Example workflow.

ual specimen can be reproduced at any time, including its origin from a certain batch or assay.

6 SUMMARY AND OUTLOOK

The system described allows for a generic documentation of the complete lifecycle of a material sample. By the flexible modelling of the system, it is possible to rapidly respond to changing requirements and to quickly supply extended functions. For example, new activity types can be added during runtime and configured with the corresponding attributes and rules, which are then available immediately. Within the *NanoCare* project, this system will be used to document the activities with respect to the investigation of materials and, hence, to support the reproducibility of the scientific results obtained from the physical and biological studies.

For further enhanced flexibility, it is foreseen to add scripting capabilities (Kearns, 2002) in order to enable customising of new activity types with specific behaviour.

To allow for a largely intuitive operation of the system, further efforts are currently underway to optimise the user interface in terms of usability (Betche, 2008).

REFERENCES

Bauer, C. and King, G. (2006). *Java Persistence with Hibernate*. Manning, Greenwich, CT.

Betche, K. (2008). Gegenüberstellung etablierter Methoden zur Evaluation der Gebrauchstauglichkeit von Software-Anwendungen sowie Konzeption einer Usability-Studie anhand einer gegebenen Software. Master's thesis, University of Cooperative Education, Karlsruhe.

Brittain, J. and Darwin, I. (2007). *Tomcat: The Definitive Guide*. O'Reilly, Sebastopol, CA.

Interface21 (2007). Acegi security [computer software]. http://www.acegisecurity.org/.

Kearns, D. (2002). Java scripting languages: Which is right for you? http://www.javaworld.com/javaworld/jw-04-2002/jw-0405-scripts.html.

Raible, M. (2007). Appfuse [computer software]. http://www.appfuse.org/.

Walls, C. and Breidenbach, R. (2007). *Java Persistence with Hibernate*. Manning, Greenwich, CT.