Math-Bridge Deliverable D 6.1: Integration of External Tools

George Goguadze, Stefan Winterstein, Johan Jeuring

June 15, 2011

1 Introduction

This deliverable describes two major goals of WP 6: make existing interaction tools for interactive exercises, including assessment tools, interoperable by integrating them with the Math-Bridge system, and integrating external semantic evaluation tools such as Computer Algebra Systems (CAS) and domain reasoners to provide domain diagnosis in multiple mathematical domains. The latter services should enable automatic generation of interactive exercises with intelligent diagnosis and automatic feedback.

In addition, we also describe a generic approach to link Math-Bridge to other learning management systems via single sign-on, so that Math-Bridge can easily be accessed by users of the host LMS without additional registration and login.

The interoperability achieved by Math-Bridge is one of the measures to support exploitation of the project's results because (in addition to the personalized access) this will be a unique feature (in its coverage) among competing learning environments for mathematics.

On the Math-Bridge side, the approach is generic so as to integrate players of other parties (associates) in the future. That will increase the competitiveness of the final Math-Bridge service.

In this report we describe the integration of the external exercise system Stack, embedding of the interactive exercises of Mathe-Online, and the translation of the exercises of Kassel-Paderborn educational module, as well as generation of interactive exercises from static problem statements. Further on, we address the integration of semantic evaluation services, such as Computer Algebra Systems and domain reasoners for different math domains. Finally, we describe integration of Math-Bridge into external Learning Management Systems, namely Moodle, Clix and Ilias.

2 External Exercise Systems

External exercise systems can be integrated on the software level, or by porting the exercise sources to the Math-Bridge representation. Using the software approach, the external exercise tools are called from Math-Bridge and send their diagnosis back to Math-Bridge's event system. This makes the interactions visible to the core system, which in turn can use the information for the personalized access.

2.1 Stack Integration

Stack is an open-source system for computer-aided assessment in Mathematics [Sangwin 2008]. It is used by Tampere university, running independent of Math-Bridge.

2.1.1 Integration Concept

In order to integrate Stack exercises into Math-Bridge, the two applications are joined using a remote API provided by Math-Bridge. Via this API, Stack can retrieve user and session information and report exercise interactions and results back to Math-Bridge. Math-Bridge will then use this information as if the exercises had been run internally and update the learner model according to Stack's diagnosis.

In more detail, the integration is achieved as follows:

Placeholder exercises: Every Stack exercise installed on the remote server is represented by a *placeholder exercise* in Math-Bridge. The placeholder exercise contains only the metadata (such as title, difficulty, relations to the onthology) but not the script of the exercise. This information allows Math-Bridge's adaptive components to treat Stack exercises as local learning objects.

Start link: When rendering a Stack exercicse, for example as part of a course, Math-Bridge embeds an HTML link (URL) to start the exercise in a separate browser window. This link transfers control to the remote Stack server (e.g. running in Finland) and passes a disposable authentication token to Stack.

Authentication token and user data: Using the token from the start link, Stack issues a remote XML-RPC call to Math-Bridge's authentication service, passing the token. If the token is accepted, the lookup call returns the user and session data to be used for the exercise (for example to greet the user by name). The lookup call also returns the URL of the Math-Bridge remote event service, where later the exercise interaction events can be published.

Remote exercise interaction events: Stack performs the exercise locally. For every user interaction, Stack sends an *event* object to Math-Bridge, using a remote call to Math-Bridge's event service. Math-Bridge feeds the event data to its learner model as if the exercise had been run locally.

The integration approach outlined here is generic and can be used to integrate other external exercise systems in the future.

2.1.2 Integration Scenarios

Two distict scenarios have been developed for Stack: running an assessment test and running single exercises.

Assessment scenario: This scenario assumes that several Stack exercises are grouped into a single *assessment test*. The start link transfers control to that test, and Stack sends back diagnosed results after the user has finished the test.

While this scenario has been specified and implemented, its main drawback is that the user can abandon the test session at any time (by simply closing the browser window) without both applications noticing this reliably. This might prove problematic in real world use, so both partners TUT and USAAR agreed to pursue an alternative approach.

Single exercise scenario: Here, Math-Bridge starts a single Stack exercise from within a course. The language of the user is passed on with the start link, so for the user, the only difference to starting a local exercise is that it runs in a separate browser window or tab. While the exercise is running, Stack reports user interactions to Math-Bridge in real time, using remote ExerciseStep events (along with ExerciseStarted and ExerciseFinished). This event data is available to Math-Bridge components immediately and can be used for course adaptation and later reporting.

2.2 Embedding and enhancing Mathe-Online Exercises

Integration of a collection of Mathe-Online[1] exercises required modifications of both Math-Bridge system and the exercise collection. First of all, a specific exercise generator component has been implemented by Saarland University, which is responsible for proper embedding of the javascript-based exercises of Mathe-Online into Math-Bridge.

In addition to this, placeholder exercises pointing to the corresponding javascript files were encoded. These placeholders also include the exercise metadata to be used for selecting those exercises as well as for reporting to the student model.

Finally, several javascript libraries within Mathe-Online packages, which are responsible for evaluating the correctness of a learner's answer had to include the calls of an additional function, sending exercise events to the Math-Bridge event system. This function automatically extracts the metadata from the placeholder exercise and sends it, together with the evaluation results, to the Math-Bridge event system.

The approach we take is different from Stack integration in that it provides a mechanism for embedding exercises inside the Math-Bridge environment instead of connecting to an outside environment.

Our approach is general and will work for any kind of interactive exercises that run on the client within an HTML page and are able to report the interaction result using the defined javascript function.

In case an embedding is possible it is definitely preferable to communicating with an external system, since in this case no extra work for user autentication and related issues is needed.

2.3 Embedding Exercises from Kassel-Paderborn

The test modules provided by Kassel-Paderborn have been converted to Math-Bridge format and integrated into the corresponding content collections. The conversion has been performed semi-automatically, since the mathematical formulas were represented using presentation-oriented LaTex syntax.

The corresponding interactive exercises in the Math-Bridge format were annotated with appropriate metadata, used by the tutorial component when searching for an appropriate exercise for the learner.

[1]See `http://www.mathe-online.at/`

2.4 Converting static exercises into interactive ones

Additional work has been done on converting static representations of problem descriptions, provided by TUT, into simple single-step interactive exercises. In order to do so, the problem statements, represented in LaTeX, have been annotated with additional information, such as metadata and correct answers to the problem. The results of these annotations were saved in a file, which consists of records of tab-separated parameters, needed for the exercise generation. An automated conversion tool then generated from these parameters the exercise automatons in the Math-Bridge format. Since the LaTeX notation is not semantic, and hence not fully machine-readable, some manual pre-processing of formulas was necessary.

The resulting exercises use semantic evaluation of a student's answer and are annotated with all metadata and diagnosis information needed for updating the student model after a student interaction. These exercises have now been included in the corresponding content collections.

3 Semantic Diagnostic Services in Math-Bridge

The Math-Bridge system possesses a distributed broker architecture for connecting semantic web services that serve the needs of interactive exercises. This architecture is described, e.g. in [Goguadze, 2009 b]. Math-Bridge defines a set of educationally oriented queries to be sent to the external service. These queries are used by the exercise engine of Math-Bridge for obtaining a diagnosis of learner's actions and generating feedback in the exercise steps.

Based on the context information in the query, the broker component decides which external system is able to answer the query and forwards it to the corresponding system. Figure 1 shows the architecture of the query broker and the systems currently connected to Math-Bridge. Each newly connected service subscribes in the query broker by specifying the list of contexts it can answer. Furthermore, a dedicated component is implemented that establishes a connection to the given remote service using the necessary protocols. Finally, an additional transformation program called phrasebook needs to be implemented in order to translate the generic query language of the Math-Bridge system into the custom syntax of the corresponding external system and thereafter translate the result returned by the external system back into the syntax of the Math-Bridge system.

In order to connect the new domain reasoners of OUNL to Math-Bridge we followed this scheme and implemented a dedicated component,

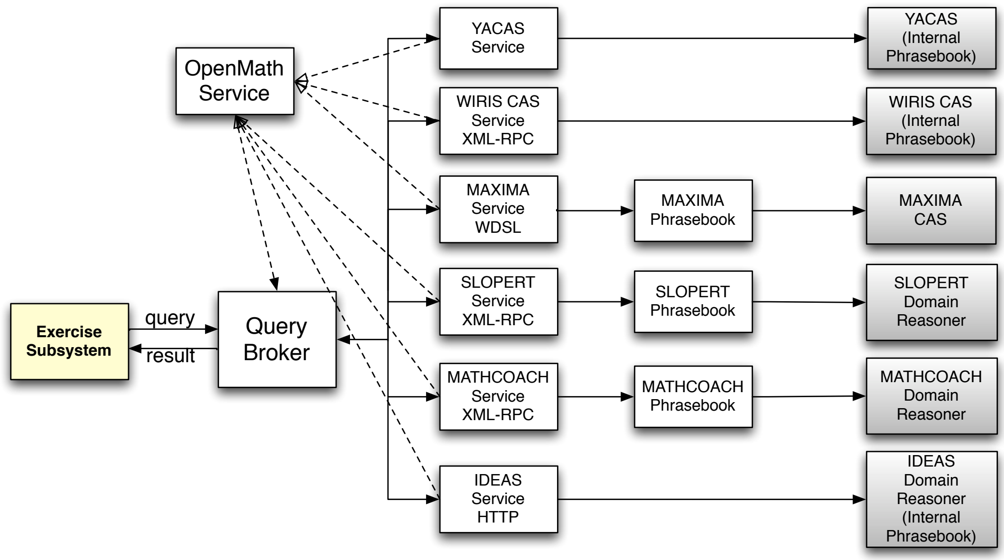

Figure 1: Semantic Service Broker

responsible for remote communication with the OUNL web-services, which included a partial translation of Math-Bridge educational query format into the query protocol of the OUNL reasoners. The translation was only partial since the OUNL reasoners already have native support for the OpenMath representation of mathematical expressions that are used in the body of the Math-Bridge queries.

3.1 YACAS **Computer Algebra Server**

The YACAS Computer Albebra System has already been used by the ActiveMath platform as an external back-engine for semantic evaluation. However, previously this system had to be installed on the computer serving ActiveMath and called as a local shell process. It did not possess a multi-threading mechanism, so for every semantic evaluation query, a separate process was started.

In Math-Bridge, an extension of YACAS has been implemented by Saarland University and DFKI. This extension represents a built in HTTP server, built on the top of existing YACAS server, which did not handle any protocol other then YACAS syntax, and was single-threaded, so one had to wait until one query finished to send the next one.

The new YACAS server processes remote semantic evaluation queries issued by Math-Bridge or any other system using the supported protocols. The server can run several instances and is fully multi-threaded. Moreover, it possesses a process management architecture, in which processes can be reused for queries having the same evaluation context. Under *context* we understand a mathematical description of a task, that can consist of set of production rules to be applied to the given mathematical expression to obtain the correct solution. For more information about the query context in Math-Bridge, see [Goguadze, 2009 b].

The choice of YACAS system in Math-Bridge is not accidental. It is one of the few Computer Algebra Systems using rule-based reasoning in a modular way. The mathematical reasoning modules are logically separated and stored in separate files containing scripts in YACAS syntax. Because YACAS incrementally generates solutions, it can be parametrized to generate full solution paths instead of only final results of calculations. This makes YACAS capable of deeper domain reasoning. As a proof of concept for this, two additional modules were developed at Saarland University that implement domain reasoners for fraction arithmetics and integrals. These reasoners are capable of stepwise diagnosis and hinting in the corresponding domains (see e.g. [Masood, 2009]).

The YACAS server can be configured to serve several mathematical contexts, each represented by a reasoning module. Each of the contexts can be accessed by addressing different server ports. The YACAS server supports two ways of communication. One way is a network socket connection used to pass raw XML strings in OpenMath format. The query is sent to the YACAS server as XML, and when the connection is closed the evaluation result is sent back in the same form: a serialized XML string.

Another way of communicating with the server is to use the HTTP protocol for reaching it via the web.

The HTTP server was implemented from scratch, consulting the HTTP specification[2]. It can handle any valid HTTP 1.1 connection. It ignores unsupported features, following the specification, and the result is always valid.

4 Exercise generation by external domain reasoners

Math-Bridge possesses a generic component for generating exercises powered by Domain Reasoners. This component has been reused and customized

[2]See `http://www.w3.org/Protocols/rfc2616/rfc2616.html`

within a subclass, which uses a simple strategy for stepwise problem solving. This strategy includes a stepwise evaluation of a learner's answer and provides next-step hints at every step as well as full solutions to problems. Since this simple strategy is valid for all supported domains, the same component is reused for generating exercises in several domains.

The decision which reasoner has to be used is based solely on the task context information, encoded as a parameter of the exercise element, and is performed automatically by the system. It is also possible to manually assign a concrete task to the exercise, which is specified as a parameter for the exercise generator. If not specified, the task is generated automatically by the domain reasoner, based upon the information about the task domain and the difficulty. More information on how to author exercises using generators in ActiveMath can be found in [Goguadze, 2009 a].

We have tested the generic exercise generation framework with IDEAS domain reasoners provided by OUNL and integrated this into Math-Bridge system by implementing one dedicated semantic service adapter within the Math-Bridge system which can be used to connect to all OUNL domain reasoners. The IDEAS domain reasoners are a family of formal systems implemented in the programming language Haskell, and capable of stepwise problem solving in several mathematical domains. For more information on the IDEAS reasoners and their problem solving technology, see [Heeren and Jeuring, 2010].

4.1 Internationalization of generated exercises

The generic exercise generation framework of the Math-Bridge system possesses a mechanism for internationalizing the generated exercises. The set of variables used in the feedback templates within the feedback generator component can be substituted on the fly by concrete feedback phrases in the current language of the user.

The feedback phrases for each of the languages are authored in the configuration files of the internationalization package of the Math-Bridge system.

4.2 Authoring parameters for generation

Saarland University has also implemented a small web-based authoring tool for domain reasoner-powered interactive exercises, shown in Figure 2. In this interface, authors can enter the problem statement of the exercise and some metadata, select the context of the task, let the system generate the

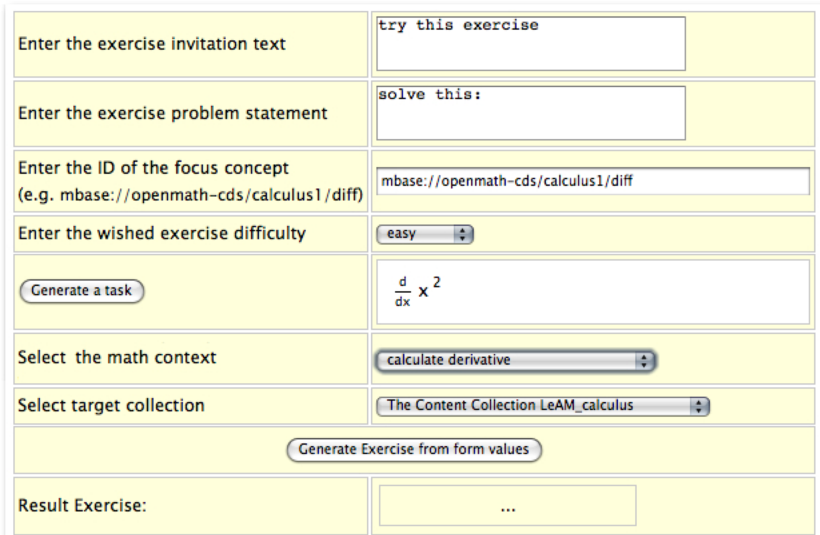

Figure 2: Authoring Exercises Using Domain Reasoners

exercise, and store it in the system. Note that the tasks can also be generated automatically by the domain reasoners, but if the author does not like the task, he can keep clicking on the "Generate a task" button, until he thinks a generated task is appropriate.

A sample exercise produced by this tool is shown in Figure 3. Such exercises offer hints for the next step as well as complete worked-out solutions starting at the current step.

OUNL has made its domain reasoners easily accessible within the Math-Bridge service by creating almost a thousand exercises that can be solved using the domain reasoners developed in the first one and a half year of the project. The exercises placeholders have been encoded and published in the Math-Bridge service.

5 Linking Math-Bridge to other Learning Management Systems

Wikipedia defines a learning management system (LMS) as "a software application for the administration, documentation, tracking, and reporting of training programs, classroom and online events, e-learning programs, and training content."

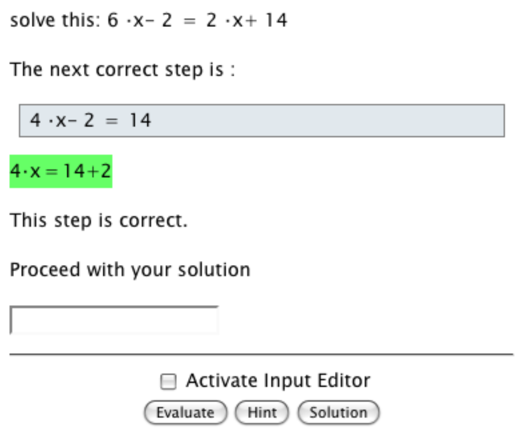

Figure 3: Resulting generated exercise

While being an LMS in itself, Math-Bridge can be linked to other LMSs. The advantages are mutual: Math-Bridge can provide specialized math services for the host LMS, whereas on the other hand it can benefit from a large installed user base.

5.1 Integration Concept

Math-Bridge has been integrated with three other LMSs: Moodle, Ilias and Clix. All LMS integrations of Math-Bridge rely on the same underlying concepts (see Figure 4), while differing in the technical details.

LMS roles: In all integration cases, Math-Bridge acts as the *client LMS*, being attached to a *host LMS*. The host LMS provides the main entry and registration point for the user. It owns and has control over the user database. The client LMS is configured to know the host LMS it is attached to, while the host LMS typically is unaware of the client system.

Single sign-on (SSO): The most basic form of integration (which also provides the most user-visible benefit) is *single sign-on*. In practice, this means that users of the host LMS will never have to subscribe or login to Math-Bridge directly, as all their user information is transfered automatically. Once users are logged in to the host LMS, a simple link will start a new Math-Bridge session in which they are automatically logged in.

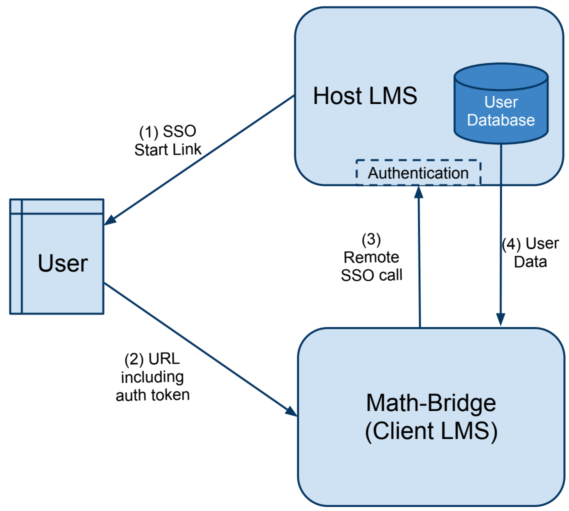

Figure 4: LMS integration by single sign-on mechanism

The necessary data is exchanged between the host and client LMS in the background, without the user noticing it.

Authentication and data exchange: When starting a new MATH-BRIDGE session, the host LMS typically transfers an authentication token via a URL. Using this token, MATH-BRIDGE will contact the host LMS via its respective interface and retrieve the user's data (such as account, full name, language etc.). If the user is not known, this information is used to create a new MATH-BRIDGE user on the fly (if the user is already known, the local user data is updated). MATH-BRIDGE will then proceed to login the user as if she had filled in the login form herself.

From the users' point of view, access to MATH-BRIDGE is presented as a learning resource in their familiar LMS. Opening the resource opens a new browser window or tab with a running MATH-BRIDGE session, where they can start using MATH-BRIDGE right away.

5.2 Integrated LMSs

Every LMS integration of MATH-BRIDGE relies on the concepts outlined above, but differs in the technical implementation.

5.2.1 ILIAS

Ilias is an open-source LMS, widely used in Germany and many other european countries[3].

SSO integration for Ilias has been implemented for Ilias version 4.1 and is based on the SOAP API provided by this LMS. The SOAP API gives convenient access to all required functionality, such as retrieving a user id from a session id and accessing a user's profile data. For authentication, a user's session id is used.

From the Ilias side, access to Math-Bridge is configured through a dynamic weblink that uses simple PHP code to create the dynamic URL.

5.2.2 CLIX

Clix is a commercial e-learning system developed by the German company im-c[4]. It is used by many universities in the Saar-Lor-Lux region, such as the University of Saarland.

SSO integration for Clix is based on a custom HTTP protocol developed in cooperation with im-c. The SSO request transfers a one-time token, which Math-Bridge uses to contact Clix via a secured HTTPS connection.

For Clix, access to Math-Bridge is configured through a special learning resource that creates the SSO link with the disposable authentication token.

5.2.3 Moodle

Integration of Moodle differs from integration with other LMSs, because the Moodle version used by Kassel-Paderborn (Moodle version 1.9) does not provide a remote API that could be used to retrieve user data (this API was only introduced with Moodle 2.0).

Therefore, a method know as *screen scraping* had to be used. To retrieve user data for an SSO request, Math-Bridge accesses Moodle's page with the user data and extracts the desired information from the HTML code directly.

In order for this to work, Math-Bridge needs access to the user's Moodle session. This is achieved by configuring Moodle's web server in such a way that Math-Bridge is run in the same domain as Moodle (using Apache's rewrite engine). Using this setup, the Moodle session cookie is

[3] `http://www.ilias.de`

[4] `http://www.im-c.de`

transferred with the SSO request and can be used by Math-Bridge to access the user's profile page.

For Moodle, access to Math-Bridge is configured via a static link, as no authentication token is required.

References

[Goguadze, 2009 a] G. Goguadze. Representation for interactive exercises. In J. Carette, L. Dixon, C. Sacerdoti Coen, and S. M. Watt, editors, *Proceedings of 8th International Conference on Mathematical Knowledge Management MKM 2009*, volume 5625 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 294–309. Springer, 2009.

[Goguadze, 2009 b] G. Goguadze. Semantic Evaluation Services for Web Based Exercises. In M. Spaniol, Q. Li, R. Klamma, R.W.H. Lau, (Eds.) Proceedings of the 8th International Conference on Web Based Learning - ICWL 2009, LNCS 5686, pages 172–181, Springer, 2009.

[Heeren and Jeuring, 2010] B. Heeren and J. Jeuring. Adapting Mathematical Domain Reasoners. In Proceedings MKM'10, the 9th International Conference on Mathematical Knowledge Management, LNCS, Springer, 2010.

[Masood, 2009] S. Masood. A Domain Reasoner for Fraction Arithmetics, Serving Intelligent Exercises, Master Thesis, Saarland University, 2009.

[Sangwin 2008] C. J. Sangwin. Assessing Elementary Algebra with STACK. International Journal of Mathematical Education in Science and Technology, 38(8):987-1002, December 2008.