Loosely Coupling Web-Applications

Paul Libbbrecht [a1] Enrique Machuca [b] Mark Spanbroek [c]
[a] *DFKI GmbH, Saarbrücken, Germany*
[b] *Universidad Malaga, Malaga, Spain*
[c] *Eindhoven University of Technology, The Netherlands*

Abstract.
When developers of two web-based environments wish to share activities, they need to negotiate the ways to do so. It may boil down to an a simple authorization, or URL exchange.
Generally, however, the exchange is more convoluted and measures have to be taken to guarantee the highest quality in the browsing experience. We address this problem in an attempt to generalize the action of following a hyperlink into a *browser delegation*. Report on the experience gained in implementing the delegation between ACTIVEMATH and SIETTE is provided.

Keywords. world-wide-web, browser, delegation, decentralized, integration,

Introduction

The World Wide Web has brought thousands of knowledge libraries and hundreds of educational activities within two clicks of a mouse. Inviting a reader from a web-page to another site is thus natural. This is generally materialized by a hyperlink, an ingredient which has made the world-wide-web so powerful.

An author who writes a link assumes the target of the link will stay what he expects. When the client browser follows the link, it will browse the link's target site, but will have no way to *get back* except by following browser history.

When the two sites are more elaborate web-applications, such links are typically generated instead of authored and are often only valid during a specific session of user activity. For example the fact that the user is logged in, or the relation to its user-model. Some web-applications provide relatively transparent links which could be used as in the hand-authored HTML case. But the problem of getting back is still not solved.

As a result a coupling is needed between the two web-applications. This coupling should allow the managed delegation of the web-browser between one server and the other and should allow an experience close to the one that would happen on a single server. To achieve such experience, currently, developers of the web-applications need to be involved, they need to exchange and agree on protocols and information to be exchanged during the delegation. The results are rarely (ex)portable and will only be maintained as far as the collaboration between the two sites' owners remains.

An integration of several systems is the objective of several general purpose architectures:

- Knowledge-Tree [2] uses the proxy-approach where a central server tunnels the content and requests which allows them to be adapted and tracked and which allows a central learner-model to be built from it.
- MEDEA [11] is a project whose aim is to integrate several existing learning-systems in a single one based on a shared ontology of the domain knowledge and of the user modelling information.
- Through the authoring of web-service composition scenarios, APeLs [4] has a potential to integrate heterogeneous services to provide a complete learning experience. It needs, however, a centralized service composition framework.

We describe an approach to generalize the task of *loosely coupling* two web-applications so that it should be possible for two authors to decide to loosely couple the web-applications which host their content.

A typical example is that of two collaborating researchers in the same domain. They will, typically, have complementary course-material and want their students to use each others course material.
Typically, these two researchers will, at most, be authors of content. Most probably they will not even have administrative rights on the e-learning system. Should it prevent them to share their course material?

This paper starts with an analysis of the requirements. We present the three systems that are planned to be loosely coupled within the LEACTIVEMATH EU project. The browser delegation scenario is, then, described in some detail. Finally, we describe the prototypes and current development, and present future work.

1. Loose coupling requirements

The servers that take part in the process of delegating a user-and-his-browser from one server to the other are as follow:

- the *guide-server* is the server where our browser is before this scenario comes into play. This guide server has some reasons to wish to send the user's browser to the following server. These reasons are probably the result of some previous exchange of knowledge such as the browsing and writing of authors or such as exchanges between agents using a shared ontology.
- the *activity-server* which is the server that will serve the activity to the browser that it is delegated from the guide-server. The term activity is used quite loosely to describe any sequence of web-interactions. In the setting of two communicating learning environments such as ours, it will be refined.

From the end-user perspective it should be perceived that the two servers are more or less the same servers. We can note the following requirements:

- single-sign-on: the user with his client-browser should not need to re-authenticate when passing from one server to the other. Based on the trust between the servers and on authentication at the guide-server, authentication should be deduced on the activity server.
  Note that it is unreasonable to expect that the first contact of a user with the web-application on the activity server will always directly start the activity. For

example, several web-applications will ask registration questions in order to seed their user-models.

- interface homogeneity: one could expect a similarity of textual and graphical language between the two applications without requiring full homogeneity.
- keep-going: a dead-end in the navigation course will be taking the user away from his current goal. Therefore the guide server should continue guiding after an activity is finished or indicate the achievement of a milestone.
- privacy: since we are talking about the process of exchanging information related to the user experience, personal-history, and/or credentials to enter domains, the user should be sure that his data is kept safely. This safety is generally guaranteed at each servers site but needs to be cared about in the exchange between the two servers. We expect encrypted and authenticated communication to be sufficient and manageable for this purpose.

It is important to note that a request for coupling is very far away from an integration at the development level. We hope to make possible the coupling of two web-applications on the impulse of two content-authors and under the auspices of system administrators.

2. Intended Usage

In this section we take the time to describe the web-applications that we intend to loosely couple and the scenarios where delegation is to happen.

2.1. The SIETTE *adaptive assessment tool*

SIETTE [3] is a web implementation of *Computerized Adaptive Tests*. The system is based upon a well-founded theory, *Item Response Theory* which explains how to diagnose learner's knowledge from answers to questions (items). The main advantage of using this approach is that we do not depend on number of items used to obtain the knowledge level of a student. With few items this theory ensures the correct knowledge level estimation.

SIETTE provides a web interface to pose questions to students, but also can be integrated within external systems. SIETTE stores the items in a *knowledge base*. These items can be authored by a teacher using the *test editor* but he can also analyze the behavior of students in tests using the *analyzer*, and correct difficulty on items thanks to the *item calibration tool*

Recent features [7] include *external items*, so called because these are exercises that are presented in a different system than SIETTE. The evaluation of the user's input in these items is then received as input from some external site. A first attempt of the delegation process actually happened as SIETTE delegating the browser to ACTIVEMATH for an exercise, using this approach.

Further work is being made to achieve delegation in both directions. Currently SIETTE is being adapted to be accessed using the delegation process, in order to let ACTIVEMATH delegate the evaluation of a topic to SIETTE. In the same way, ACTIVEMATH will let SIETTE discover the resources in the learning environment, so as content can be stored into SIETTE. Then, an assessment session from a topic on ACTIVEMATH can be delegated, but also an assessment on SIETTE can delegate the exercise's presentation to

ACTIVEMATH. In this way, SIETTE is used just as a sequencer of items to provide the adaptive behavior and the posterior statistical analysis of results.

2.2. *The* ACTIVEMATH *learning environment*

ACTIVEMATH is a web-based learning environment. It uses a semantic representation of the content, assorted with metadata annotations, to present learning material to learners and to offer interactive mathematical exercises. ACTIVEMATH maintains a learner-model describing the estimated *knowledge*, *comprehension*, and *application-capacity* of the learner for each concept. Using it, ACTIVEMATH selects books of content to be learned to achieve learning objectives and suggests further reading. For more details about the learning environment see [13], [10] and the references therein. Every user-interaction in ACTIVEMATH is on the web and ACTIVEMATH uses several web-services (see [9]).

The ACTIVEMATH learning environment is at the heart of the FP6 EU project named LEACTIVEMATH. Among others, this projects intends to provide an integrated platform including ACTIVEMATH, the SIETTE assessement tool, and an exercise repository. Moreover, it should combine exercises with tutorial dialogues, prototype a course-generator based on a reactive planner (see [12]) and a learner-model based on the competency model of the Pisa study [1].

2.3. *The* LEACTIVEMATH *exercise repository*

As described in the previous section, the need for interactive exercises is fundamental for an experience in the ACTIVEMATH learning environment. It was proposed to make in the LEACTIVEMATH EU project a collection of re-usable exercises that could be browsed, searched, experienced, and re-used by the public in the domain of calculus. The LEACTIVEMATH repository is being realized at the Eindhoven University of Technology.

2.4. *Delegations in* LEACTIVEMATH

Maintaining a learner-model with values that approximate the estimated mastery of the learner is a delicate task which relies, in ACTIVEMATH, on tracking the learner's reading actions and receiving exercise diagnoses. Both of these methods, however, can only be achieved after some time using the learning environment whereas one wishes to offer guidance as early as possible.

A first attempt in this direction was a form of *self-assessment* which invited the learner to estimate her mastery of each of the topics in the current course. Little positive experience was gained with this self-assessment approach and more efficient bootstrapping mechanisms of the learner model should be proposed.

One of them can be to invite the learner for an assessment session which should discover her knowledge of the domain as in SIETTE. This delegation will be integrated.

The decision to do so shall be taken by the tutorial component, which is the component responsible for the content selection and further-reading advise, or the open-learner-model, which is the component responsible to present to the learner the estimated mastery. During the interactions with the tutorial component or the open-learner-model, hyperlinks inviting the learner for an assessment session will be presented. The latter, when clicked, will take the browser for an assessment session in SIETTE which upon completion returns the learner's browser to the component it was before. After such a session, the learner-model will be updated as a result of the tests. The tutorial component will be able to provide better advice on further content to read or exercises to achieve. The open learner-model will present these links, among others, within dialogues with the student about the learner-model when doubts are emitted about the learner-model estimates. After the assessment, it will be able to present estimates with a greater evidence which should enhance the learners' trust in the capabilities of the system.

A requirement for the link inviting the session to be displayed is the discovery of the assessment sessions available. For such a session to be useful to the LEACTIVEMATH learner-model, the assessment measures should handle the same topic-names and do so using the same domain knowledge. A simple web-service call was agreed upon that should take, as parameter, the name of a domain knowledge node and return a list of resource-identifiers for each activities that are *for* this domain knowledge node. This allows the discovery of exercises in the repository and of assessment sessions in the assessment tool.

For the integration of SIETTE and ACTIVEMATH, more tuning is needed as more of the domain knowledge is needed for adaptive testing. From the set of concepts in ACTIVEMATH, and based on a table-of-contents typical of the domain, an export of the domain knowledge can be done. This exported domain knowledge can be enriched with the exercises attached to each exercise in ACTIVEMATH.

In the first version of this export, the only exercises supported where multiple-choice-questions and could be exported to SIETTE. Since then, however, richer interactivity exercise-types are supported by ACTIVEMATH. Therefore we use the delegation process again: SIETTE can delegate the browser back to ACTIVEMATH for exercises done there. The export is then limited to export *external exercises*.

3. The Browser Delegation Scenario

In this section we describe in detail the steps of the delegation scenario as a sequence of remote procedure calls. This sequence can be followed in picture 1.

The browser-delegation scenario starts with our user using the browser currently interacting with the *guide-server*. The latter, through authored content or discovery, offers a link that should lead the browser to the *activity-server* for the time of the activity and *bring it back* when finished.

The scenario described here is a sequence of web-service calls to the guide or activity servers interleaved with browser actions. It is not clear wether any web-service sequencing or choreography language can be used for this description.

check-availability a call should be made to check that the delegation is possible. It should include:

- a *resource-identifier*, a string describing the activity
- optionally, an *interaction type* that represents a *verb* describing what is expected the browser will have as interaction with the activity server. Examples include "run an exercise", or "see a piece of content"
- a user-identifier: a string to identify the user on both sides. This string may be the result of a translation to allow mapping between learners.

Faults may happen as the result of this request. Most of the HTTP error-codes [6] can apply here. The guide-server, receiving such a fault should update its knowledge about the availability of the activity for the given user and should let it proceed to a *further step*. No other result except a possible fault is expected from this call.

wish-to-start the guide-server notifies the activity-server, through a web-service call, that it intends to send a browser to interact with a given resource. It provides at least the following arguments:

- the *resource-identifier*, *interaction-type*, and *user-identifier*
- an amount of other optional information to allow the interaction to be best suited to the user. Provide a handle to the learner model may be a solution if the activity-server supports it. More realistically, we expect this to be a space for a small set of values computed from the learner model and encoded within a shared ontology or such information as the expectation about the duration of the activity.
- a URL-to-return-the-result-to which is a resource-locator to the web-service where to invoke the *activity-finished* web-service called later
- optionally, a URL-to-send-the-events-to can be provided so that events can flow between the servers while the interaction happens (see [9] about the usage of events).

In response to the notification received from the guide-server the activity-server should provide a URL-to-lead-the-browser-to, the guide-server now directs the browser to this URL. This URL should contain enough information so that the interaction can start right away. Among others, this means that this URL contains extra *tickets* that would automatically log-in the browser on the activity-server when it first requests this URL.

The request response may contain extra information such as URLs where events can be sent to or where the learner actions can be tracked. If the *check-availability* call has been done shortly before, no fault should be raised.

activity-cancelled In some cases, the call to *wish-to-start* will not be followed by the actual activity. The guide-server should then call this method with the parameters of the

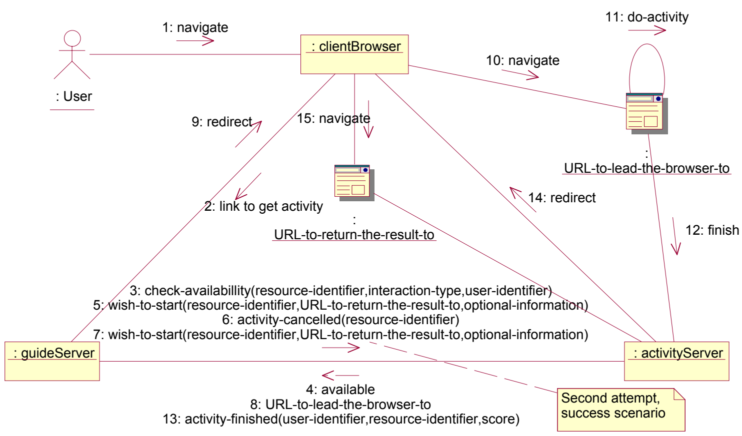

Figure 1. Loose-coupling steps.

resource and user-identifiers which should de-allocate any resource allocated for activities for this user- and resource-identifiers.

Now the learner interacts with the resources on the activity server.

activity-finished when the interaction is finished, the activity-server should direct the browser to future interactions that the guide-server should indicate.

It contacts the URL-to-return-the-result-to and invoke an activity-finished method with parameters including the user- and resource-identifiers as well as a numerical score, a floating number between 0 and 1. This should be promptly answered by the guide-server which will provide a URL-to-come-back which will be sent to the browser as the next place to go to. Faults should only occur in exceptional cases here as there would be no other option but to forward the error to the user, who can then only go back using his browser history.

4. Implemented Prototypes

The integration of SIETTE and ACTIVEMATH within the LEACTIVEMATH learning-environment is currently under work. The delegation process seems to be sufficient aside of the discovery activities. Among the steps for this, we have implemented the delegation of ACTIVEMATH exercises within an assessment session in SIETTE using XML-RPC web-service calls. It would be too verbose to present the method-names and parameters here, but it suffices to say that the method-names were the ones of the delegation scenario and parameters were them as well with space for extra information exchanged during the delegation (such as the URL to a learner-model web-service).

Based on this experience, the generalization of the delegation process has been under-work and will be used in the integrations planned in LEACTIVEMATH.

The prototype that we realized delivered activity-links which point to the guide-server and, only when requested, trigger the *wish-to-start* web-service call. This, in turn, resulted in a redirect of the browser to the activity-server. The same was true when invoking *activity-finished*. This setting made several HTTP requests (between various hosts thousand of kilometers apart) for a single page download and turned out to appear quite unresponsive. Calling the *wish-to-start* method earlier as link presentation seems to be a better approach.

Conclusion

We have presented a practical approach to couple two web-applications together with preliminary experiments. Reviewing the litterature seems to show little efforts in the direction of *loosely* coupling web-applications where, in principle, only content-authors and system-administrators are involved by writing content and providing authorizations. This seems, however, to be a fundamental ingredient to allow a quality user-experience while not requiring centralized systems such as Microsoft Passport[1] or the Central Authentication Service initiative.[2]

[1] MicroSoft Passport is a centralized single-sign-on solution, `http://www.passport.net/`.

[2] CAS is an open-source centralized authentication system based on cookies, see `http://www.yale.edu/tp/auth/`.

The implementation realized thus far is partial. Among the problems left to be solved for a completely portable approach to browser-delegation are the seals of mutual trust between the servers, the mapping between user-names, and the possible translation of resource identifiers. We have found in [8] several advice that might help in this direction.

This integration effort has tackled little the problem of discovering the resource of an activity. We do expect the semantic web technologies to be applicable in the future to solve this discovery problem. Similarly, exchange of learner information has not been considered here even though work has been done in this direction such as [5]. Our research, however, attempts the qualification of the delegation of activities with limited human contribution and in a decentralized way. It is probable that, once discovery and user-model exchange is achieved, a protocol similar to the browser-delegation will be used between two peers of such discovery, thereby avoiding the manual contribution of a link of the content author or the system administrator.

References

[1] *Measuring Student Knowledge and Skills – A New Framework for Assessment*. Organisation for Economic Co-operation and Development (OECD), 1999.

[2] P. Brusilovsky. Knowledgetree: A distributed architecture for adaptive e-learning. In *Proceedings of The Thirteenth International World Wide Web Conference, WWW 2004*, pages 104–113, New York, NY, May 02–05 2004. ACM Press. See `http://www2.sis.pitt.edu/~peterb/papers/p641-brusilovsky.pdf`.

[3] R. Conejo, E. Guzman, E. Millan, M. Trella, J. L. Perez de-la Cruz, and A. Rios. SIETTE: A Web-Based Tool for Adaptive Teaching. *International Journal of Artificial Intelligence in Education (IJAIED 2004)*, 14:29–61, 2004.

[4] Owen Conlan, David Lewis, Steffen Higel, Declan O'Sullivan, and Vincent Wade. Applying adaptive hypermedia techniques to semantic web service composition. In Paul de Bra, editor, *Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2003*, 2003. See also about `http://wwwis.win.tue.nl/ah2003/`.

[5] Peter Dolog. Identifying relevant fragments of learner profile on the semantic web. In *Proceeding of Workshop on Applications of Semantic Web Technologies for E-learning SW-EL'04*, Hiroshima, Japan, 2004.

[6] Roy Fielding, J. Gettys, J. C. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. RFC 2616: HTTP/1.1, HyperText Transfer Protocol, 1999. See `http://www.w3.org/Protocols/`.

[7] E. Guzman and R. Conejo. A Brief Introduction to the New Architecture of SIETTE. In P. de Bra and W. Nejdl, editors, *Proceedings of the third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 405–408. Springer, 2004.

[8] Norman Walsh Ian acbos. Architecture of the world wide web, volume one, Dec 2004. `http://www.w3.org/TR/webarch/`.

[9] P. Libbrecht and S. Winterstein. The service architecture in the activemath learning environment. In N. Capuano, P. Ritrovato, and F. Murtagh, editors, *First International Kaleidoscope Learning Grid SIG Workshop on Distributed e-Learning Environments*. British Computer Society, 2005. See `http://ewic.bcs.org`.

[10] E. Melis and E. Andres. Global feedback in ACTIVEMATH. *Journal of Conputers in Mathematics and Science Teaching*, 24:2, 2005. Accepted.

[11] M.Trella, R. Conejo, D.Bueno, and E. Guzman. An autonomous component architecture to develop WWW-ITS. In *Proceedings of workshop: Adaptive Systems for web based education*, pages 69–80, Malaga, Spain, 2002. See `http://www.lcc.uma.es/~eva/WASWBE/trella.pdf`.

[12] C. Ullrich. Tutorial planning: Adapting course generation to today's needs. In *Proceeding of 12th International Conference on Artificial Intelligence in Education*, 2005.

[13] C. Ullrich, P. Libbrecht, S. Winterstein, and M. Mühlenbrock. A flexible and efficient presentation-architecture for adaptive hypermedia: Description and technical evaluation. In Kinshuk, C. Looi, E. Sutinen, D. Sampson, I. Aedo, L. Uden, and E. Kähkönen, editors, *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT 2004)*, pages 21–25, 2004.