

# Over-computing CSCL Macro scripts?

## Gaining flexibility by using WikiPlus instead of specialized tools for authoring macro scripts

Beat Doebeli & Michele P. Notari

Beat Döbeli Honegger, PHZ Schwyz – University of Teacher Education Central Switzerland Zaystrasse 42, CH-6410 Goldau, [beat.doebeli@phz.ch](mailto:beat.doebeli@phz.ch)

Michele P. Notari, PHBern – School of Teacher Education, University of Applied Sciences, Gertrud-Woker-Strasse 5 CH-3012 Bern, [michele.notari@phbern.ch](mailto:michele.notari@phbern.ch)

**Abstract:** In this paper we show how teachers can improve collaborative learning by designing and implementing macro scripts and by regulating script sessions using enhanced wikis, called WikiPlus. To achieve optimal collaborative learning it is important to have a well designed script (preparation phase). It is also very important to being able to regulate learners' activities, because some activities can not be predicted in complex pedagogical scenarios as collaborative learning (regulation phase). WikiPlus helps enhancing the collaboration process by enabling the teacher to adapt the script whenever non predicted learner's activities happen. A prototype has been implemented, the impact of different regulation mechanisms on collaboration is discussed and wishes for macro script authoring tools of the future are formulated.

## 1. Scripts, didactic structures of collaborative learning

### 1.1 Scripts

Scripts have been widely discussed in the field of Computer Supported Collaborative Learning (CSCL). We present some common definitions we base our paper on:

A collaborative script is a pedagogical scenario that learners have to follow when they are engaged in a collaborative learning setting. A script structures the collaboration process by guiding students' activities as building groups, guiding argumentation processes, visualizing ongoing work, and so on. Some of these activities are computer-based, some are not (Dillenbourg, 2002)

The main idea of collaboration scripts is to prompt cognitive and social processes by participants that might otherwise not occur, thus enforcing a fruitfully structured interaction, and consequently improving the joint problem-solving and knowledge acquisition (Rummel, 2006).

In a wide area of CSCL research scripts are implemented in combination with a semi structured computer interface. The computer interface helps organizing learner's communication and collaboration, helps keeping control of the script and is the main source for research data.

### 1.2 Micro- Macro- Meta and other -scripts

The scientific CSCL community differentiates two kinds of scripts:

**Micro scripts** are dialogue models, mostly argumentation models, which are embedded in the environment and which students are expected to adopt and progressively internalize. For instance, a micro-script may prompt a student to respond to the argument of a fellow student with a counter-argument (Weinberger et al., 2002).

**Macro scripts** are pedagogical models, i.e. they model a sequence of activities to be performed by groups. (Dillenbourg & Hong, 2008a)

**Meta scripting** means showing the scripting strategy to the learners (visibility). Meta scripting may include integrating students in the scripting design process (Notari & Döbeli Honegger, 2007b).

Dillenbourg and Hong differentiate three different operationalization layers of macro scripts (Dillenbourg & Hong, 2008a). A **script instance** is a **script class** with a specific subject and content. If this script instance is carried out by specific students it is called a **script session**.

## 2. Manyscripts, a macro scripts implementation

### 2.1 What is Manyscripts?

Dillenbourg and Hong define their platform Manyscripts as follows:

Manyscripts is a web-based environment where teachers may prepare the script they want to use with their students. Later on, the student will login Manyscripts to do the different activities that compose the script. It is similar to a learning management system such as Moodle, but focused on a few pedagogical methods called scripts (Dillenbourg & Hong, 2008b).

Up to now the following macro script classes have been implemented in Manyscripts: 'ArgueGraph', 'ConceptGrid' and 'WiSim' (Dillenbourg and Hong, 2008b). The focus of 'ArgueGraph' is on group-building strategies and the guiding of the communication among learners, 'ConceptGrid' is a Jigsaw-like scenario (Jigsaw: Aronson et. al, 1978) trying to guide the process of knowledge distribution and among the learning group. 'WiSim' increases collaborative effort by distributing simulation inputs across different phones and hence requiring students to negotiate values and coordinate their experimental design.

### 2.2 Properties of Manyscripts

Manyscripts:

- sustains implementation of three different macro scripts (ArgueGraph, ConceptGrid and WiSim) within the same environment
- can easily be implemented (using shibboleth as standard) in many university computer-networks
- has a certain didactical flexibility (group -size management like dropout of a learner within a group)
- is still quite complex to be managed by a teacher (see also: implementation of scripting sessions)
- can not be used for other types of macro scripts
- does not allow a flexible handling of activities when the learning process is ongoing.

## 3. Macro script implementation with WikiPlus

### 3.1 Potentials of wikis in education

Wikis are the simplest form of content management systems, invented by Ward Cunningham in 1995 (Leuf & Cunningham, 2001). It didn't take long until their potential for education was discovered (see for example: Guzdial, 2001). A definition of a wiki could be:

A wiki is a web server with revision control on the internet, where everybody can create, change and link WebPages without additional tools and without HTML-knowledge (Döbeli Honegger, 2007).

Taking this definition we can describe some of the potentials of wikis in education:

- **create:** creating content activates and motivates students, two important prerequisites for learning
- **change:** wikis (as every computer based editor) ease the modification of content. This allows enhancing the number of revisions required by the students.
- **link:** wikis (as every hypertext system) allows links between different parts of a text. This requires that students read and understand the parts they want to link and find fitting relations. This enhances deep understanding of the topic.
- **everybody:** wikis ease collaborative content creation and therefore ease working in groups.
- **revision control:** The revision control of wikis not only lowers the danger and damage of vandalism, the revision control can also be used to track the creation process by the teacher and the students.

Three properties of wikis ease their use in education:

- **on the internet:** As wikis can be hosted on a server in the internet, schools don't have to install hardware in their own buildings and the wiki is reachable from everywhere
- **without HTML-knowledge:** using a modern wiki is as easy as using a text processor.
- **without additional tools:** As wikis only need a web browser as a tool, there is no need for software installation on the computers of the learners. This lowers the barriers for using wikis as a learning tool.

### 3.2 WikiPlus

By design, wikis are initially unstructured. Users have to build their own structures on page level as well as on site level. Theoretically all content in a wiki is generated manually by users, there are no automatisms or

predefined structures. Nevertheless, most wikis have built in functions for dynamic content generation. We differentiate five levels of dynamic content generation in wikis and call a wiki with levels III to V a WikiPlus.

- **Special pages (Level I):** Every wiki has some special pages where existing content of the wiki is aggregated in some way or another. Recent changes, Page Index and Usage Statistics are wide spread examples of such automatic aggregation pages.
- **Built-in functions (Level II):** The next two levels of dynamic content generation offered by some wiki engines are functions which can be placed anywhere on a wikipedia. The most prominent example of such a function is a dynamically generated table of contents of a wiki page.
- **Built-in-Functions with parameters (Level III):** Parameters rise the level of freedom of built-in functions. They allow filtering, sorting, and formatting the results of functions. As an important example, the wiki engine TWiki (<http://twiki.org/>) allows the dynamic inclusion of search results into wiki pages. These features lead to content aggregation and the dynamic generation of semi-structured content.
- **Plugins (Level IV):** Some wiki engines can be extended by plugins which allow the dynamic generation of content. Example: A plugin for generation of graphs with data saved on wiki pages.
- **Integrated programming language (Level V):** The highest level of dynamic content generation is reached by providing an integrated programming language which can access and manipulate the content of the wiki. This idea has been discussed in the wiki community for a long time (see: <http://c2.com/cgi/wiki?ProgrammingInWiki> and <http://c2.com/cgi/wiki?WikiWithProgrammableContent>). Well known examples of such wiki engines are Swiki (<http://wiki.squeak.org/swiki/>) which allows the embedding of SmallTalk code and FlexWiki (Bleske 2006) which uses a proprietary programming language WikiTalk.

These levels can also be read as a temporal tendency from static content to dynamic content. First wikis allowed the easy generation of static content for non-technicians. Future wiki-like systems allow the easy definition of processes for non-technicians. This evolution makes WikiPlus a suitable tool for supporting macro scripts.

### 3.3 WikiPlus as a flexible macro-scripting engine for teachers

To use WikiPlus as a flexible macro-scripting engine the steps showed in table 1 are required.

Table 1: steps to set up a macro script using Wiki plus

Level	Who	Actions necessary
script class	IT professional	defining the structure and workflow by building template pages and writing default instructions for the teacher and the students
script instance	teacher	taking a script class template and instantiating it with specific content and modifying parameters and instructions where needed
script session	teacher & students	executing the script

With WikiPlus an IT professional is only needed for the implementation of a new script class. The instantiation or execution of a script can be done by teachers and students. WikiPlus allows rapid prototyping new script classes. Most features needed for implementing a script class are available in a WikiPlus. With WikiPlus it is possible to:

- support group building process
- structure the workflow and the content given by the teacher and produced by students in a script session
- monitor the activities:
  - by normal wiki functions like version control, recent changes or user authentication
  - by special aggregation functions provided by WikiPlus level three.

While WikiPlus has almost the same potential as specialized script engines like Manuscripts in the design phase, the big advantage of WikiPlus is the flexibility in the regulation phase. WikiPlus allows a teacher to change or enrich a running script session without programming skills or the necessity to ask an IT professional.

### 4. Prototype of ConceptGrid with WikiPlus

In the following chapter we present a possible script session of a ConceptGrid macro scripts implemented with WikiPlus. Main goal of the script session is to learn more about the concepts of semantic information structures. Students read scientific publications of researchers in the field of semantic web and information architecture. After the lecture they have to define the different key concepts and compare the different concepts. the Script design of the ConceptGrid is shown with a Didactic Process Map (see: figure 1.).

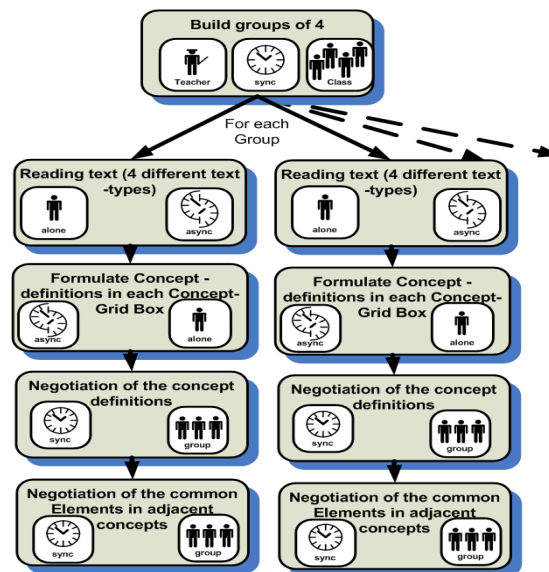


Figure 1. Visualization of the phases of a 'ConceptGrid macro script' Didactic-Process-Map-Language (Notari & Döbeli Honegger, 2007a)

#### 4.2 A ConceptGrid class prototype implementation with TWiki

We implemented a prototype template of the ConceptGrid class with the WikiPlus engine TWiki. The TWiki can host several separate wiki spaces (called *web*) sharing the same user registrations and overall definitions. The ConceptGrid class is implemented as a *template web*. An instance of the ConceptGrid class is made by creating a new web using the ConceptGrid web as a template (see figure 2).

Adding a New Web		
Create a new web by filling out this form.		
Name of new web:	<input type="text" value="MyGridInstance"/>	The name must start with an upper case letter, followed by upper or lower case letters or numbers. Specify a short name to avoid long URLs.
Based on web:	<input type="text" value="Conceptgrid"/>	Select a <a href="#">TemplateWeb</a>
Web color:	<input type="text" value="#D0D0D0"/>	Enter a <a href="#">StandardColors</a> code for the web
Description:	<input type="text" value="ConceptGrid instance for the Information Architecture class 2008"/> Enter a short description of the web. Write <code>Web.TopicName</code> instead of just <code>TopicName</code> if you include links. This will list the web in the <a href="#">SiteMap</a> (leave field empty if you prefer not to update the directory.)	
<input type="button" value="Create new web"/>		

Figure 2. Creation of a new macro script instance ConceptGrid by copying the existing template web.

The teacher now has to configure the script instance by adding the specific content (papers to read), modifying or enhancing explanations and setting certain parameters. He does this on a special wiki page (see figure 3) where the necessary steps are explained and highlighted with a special color for teacher explanations. In our example the 16 concepts, the number of students per group and all explanations for students are defined on this single page. These parameters will automatically be used in all script sessions derived from the script instance defined.

We are convinced that it is possible to reach most of the administrative facilitations of a special macro script authoring tool (like Manuscripts) with WikiPlus while retaining almost the flexibility of a wiki (level one).

### 5. Conclusions

The holy quest of Computer Supported Collaborative Learning (CSCL) is to establish environments that directly or indirectly favor the emergence of rich and intensive interactions. Rich interactions happen especially whenever learners are engaged in argumentation, when elaborated explanations emerge, the negotiation of meanings happen and the mutual regulation of cognitive processes takes place (Dillenbourg & Hong 2008). Effective Scripting leads to maximizing these rich interactions. Existing macro script processing tools like manuscripts and others are conceived for the **designing process** of one or few specific types of macro script instances and script sessions. The potential of macro scripting tools can be enhanced by a better adaptability and flexibility. WikiPlus offers flexibility and adaptability of macro scripts for the design and the implementation of

scripts. It even offers the opportunity to regulate and change student activities during the ongoing learning session. All these features can be implemented and used by teachers or even learners without the need of coding skills. Future developments of macro scripting tools should enhance design and regulation ergonomics. We propose to develop some monitoring tools to show rich and intensive interactions of the learners in order to be able to react quickly and adopt the script session. Using WikiPlus reduces the danger of over-computing by giving more possibilities to interact with design and activity of macro scripts compared to existing macro script authoring tools.

**Note to teacher**

On this page you can configure your concept grid wiki to your needs. This page is divided into four sections:

1. **Must Change:** You have to change all the settings in this section before your students can start working
2. **Can Change:** You can change these settings if you are not comfortable with the default settings
3. **Do Not Change ConceptGrid-Preferences :** Please do not change these settings unless you understand in detail what you are doing
4. **Do Not Change TWiki-Preferences :** Please do not change these settings unless you understand in detail what you are doing

**1. Must change**

**Note to Teacher:** Please fill in 16 concepts by replacing the '-' with labels for concepts (e.g. behaviorism, constructivism etc.) by [editing this page](#)

- Set CONCEPT01 = tagging
- Set CONCEPT02 = social tagging
- Set CONCEPT03 = controlled vocabulary

Figure 3. Creation of a new macro script instance ConceptGrid by copying the existing template web.

## References

- Aronson, E., Blaney, N., Sikes, J., Stephan, G., & Snapp, M. (1978). *The jigsaw classroom*. Beverly Hills, CA: Sage.
- Bleske, C. (2006): *FlexWiki*. In C. Lange: *Wikis und Blogs - Planen, Einrichten, Verwalten*, Computer- und Literaturverlag, 2006.
- Dillenbourg, P. (2002) Over-scripting CSCL, The risk of blending collaborative learning with instructional design. In Paul A. Kirschner (Ed.), *Three worlds of CSCL* (pp. 61-91). Heerlen, Open Universiteit Nederland.
- Dillenbourg P., Hong F. (2008a). The mechanics of CSCL macro scripts in Computer-Supported Collaborative Learning 3:5-23 (DoI 10.1007/s11412-007-9033-1)
- Dillenbourg P., Hong F. (2008b): Website Manyscripts, <http://manyscripts.epfl.ch>
- Döbeli Honegger, B. (2007). Wiki und die starken Potenziale. Unterrichten mit Wikis als virtuellen Wandtafel. In: *Computer und Unterricht* Nr. 66. Friedrich Verlag, <http://beat.doebe.li/publications/2007-doebeli-honegger-wiki-und-die-starken-potenziale.pdf>
- Guzdial, M., Rick, J., Kehoe, C. (2001). Beyond adoption to invention: Teacher-created collaborative activities in higher education. *The Journal of the Learning Sciences*, 10(3):265–279.
- Rummel N., Spada H., Hauser S. (2006) *Learning to collaborate in a computer-mediated setting, observing a model beats learning from being scripted* in Proceedings of the 7th international conference on Learning sciences.
- Leuf, B. and Cunningham, W. (2001). *The Wiki Way: Quick collaboration on the Web*. Boston, MA: Addison Wesley.
- Notari M., Döbeli Honegger B. (2007a). Didactic Process Map Language; Visualisierung von Unterrichtsszenarien als Planungs- Reflexions- und Evaluationshilfe. In: Marianne Merkt, Kerstin Mayrberger, Rolf Schulmeister (Eds.): *Studieren neu erfinden - Hochschule neu Denken* (page 416); Merkt M., Mayrberger K.; Schulmeister R, Waxmann.
- Notari M., Döbeli Honegger B. (2007b): *Biblionetz* : <http://beat.doebe.li/bibliothek/w02000.html>
- Weinberger, A., Fischer, F., & Mandl, H. (2002). Fostering computer supported collaborative learning with cooperation scripts and scaffolds. In G. Stahl (Ed.), *Computer support for collaborative learning: Foundations for a CSCL community*. Proceedings of the conference on computer support for collaborative learning (pp. 573–574). Boulder, CO.