A systematic approach to produce small courseware modules for combined learning and knowledge management environments

Authors:
Ines Grützner,
Niniek Angkasaputra,
Dietmar Pfahl

Accepted for publication in Proceedings of SEKE 2002, Ischia, Italy, July 15-19, 2002

IESE-Report No. 015.02/E
Version 1.0
Mai 2002

A publication by Fraunhofer IESE
Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft. The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competitive market position.

Fraunhofer IESE is directed by Prof. Dr. Dieter Rombach
Sauerwiesen 6
D-67661 Kaiserslautern
Abstract

Combining technology enabled learning and knowledge management is a very promising strategy to systematic competence building in software organizations. In order to set-up a combined learning and knowledge management environment and to support organization members in packaging small chunks of knowledge into courseware modules, Fraunhofer IESE developed the IntView-KM methodology. IntView-KM defines how to provide the organizational, methodological, and technological prerequisites that enable software engineers to produce small courseware modules (contents) as a byproduct of their daily work.

This paper presents the two phases of the IntView-KM methodology. Furthermore, tool support for courseware module production is recommended and the existing tool support summarized. Finally, experiences in applying IntView-KM and results of a first evaluation of the tool support for IntView-KM are described.

**Keywords:** Knowledge Packaging, CSCL, Courseware Module Production
Table of Contents

1 Introduction 1

2 IntView-KM 3
   2.1 LKM E Set-up Phase 3
   2.2 Small Courseware Module Production Phase 7

3 Recommended Tool Support For Courseware Authors 10

4 Existing Tool Support For Courseware Authors 12

5 Conclusions 16

References 17
1 Introduction

For software organizations, continuing professional education is a must to be competitive, to introduce new innovative technologies, and to be prepared for new challenges. The fast delivery of adequate courseware is essential to facilitate continuing education within the organization.

From the users’ point of view, courseware may be seen as an educational material (content), which is distributed via the web for training purposes. From the developers’ point of view, courseware can be perceived as a collection of multimedia documents interrelated by means of navigational structures.

These days, courseware development is characterized by three phenomena:

- The courseware imparts a large chunk of knowledge.
- It is developed by teams consisting of subject matter experts, instructional designers, graphical designers and artists, programmers and many more [1].
- The development often starts from scratch.

We call this setting “Development of large courseware from scratch” (LCS setting). Due to the phenomena listed above, it is commonly accepted that courseware development in the LCS setting is a very expensive and labor-intensive process [2]. Nevertheless, most of the current courseware development approaches (for example, [3], [4], [5]) focus on this setting. For several reasons (e.g., lack of support for all disciplines involved in courseware development, lack of life-cycle encompassing quality assurance), the application of the current courseware development approaches make many courseware development projects suffer from the same problems as traditional software projects, i.e. schedule and budget overruns [3]. To alleviate the problems of courseware development in the LCS setting, the Fraunhofer Institute for Experimental Software Engineering (Fh IESE) developed the IntView courseware development methodology. First results from case studies suggest that IntView reduces the effort spent to develop large courseware significantly [6].

Recently, software organizations have begun to pursue another courseware development setting, which we call “Fast development and delivery of small courseware modules” (FSC setting) [7]. The emergence of the FCS setting is strongly related to the introduction of knowledge management in an organization. In the FCS setting, the organization delivers not only large courseware developed from scratch but also small courseware modules that package small chunks of knowledge of its staff members. There are two major advantages of such small courseware modules:
They make existing knowledge and skills of staff members explicit and accessible for peers.
They make new knowledge and skills fast accessible in order to deploy them throughout the organization without delay.

Anything that is already part of the stored corporate knowledge, such as presentations, experience reports of staff members, guidebooks, handbooks, and technical reports can be linked to courseware modules or can be used to produce them. The main difference of courseware development in the FSC setting to courseware development in the LCS setting is that small courseware modules are no longer developed by large multi-disciplinary teams but by individual software engineers that act as courseware authors. This kind of courseware authors are subject matter experts that normally lack specific instructional, technical, or courseware user interface design knowledge and skills. That is, they need instructional, technical, and design support by an environment that we call a “combined learning and knowledge management environment (LKME)”. An LKME, for example, manages and delivers large monolithic courseware as well as courseware modules, supports courseware authors in producing small courseware modules, and provides additional functionality for knowledge management. It mainly consists of a framework providing instructional and design support as well as associated tool functionality. When using such an LKME, courseware authors can focus their work on the content specification, design, and production.

In the following, we present an adaptation of the IntView courseware development process to the FSC setting. We call it IntView-KM to stress the tight relationship of courseware development in the FSC setting with knowledge management. IntView-KM is described in detail in chapter 2. The recommended tool support for IntView-KM courseware module production is specified in chapter 3. Chapter 4 surveys the currently existing learning environments and knowledge management tools and summarizes the tool support already available.
2 IntView-KM

IntView-KM comprises two phases, the LKME Set-up Phase and the Small Courseware Module Production Phase. The Small Courseware Module Production Phase covers the development of small courseware modules by subject matter experts that occasionally act as courseware authors. These courseware authors need support in courseware module production from the LKME. The specification and build-up of the LKME is done in the LKME Set-up Phase.

2.1 LKME Set-up Phase

The LKME Set-up Phase assembles the LKME framework and the LKME tool environment.

The purpose of the LKME framework is to compensate the lack of instructional, technical, and courseware user interface design knowledge and skills of the courseware authors. It provides courseware authors with:

- Sets of predefined target learner groups, educational objectives, instructional strategies, instructional structures, and navigational structures;
- Guidelines for the application of the predefined educational objectives, instructional strategies, instructional structures, and navigational structures;
- Instructional structure as well as navigational structure templates.

The LKME tool environment, besides managing and delivering courseware, supports courseware authors during the Small Courseware Module Production Phase in applying the LKME framework.

The LKME Set-up Phase includes activities of the requirements specification and of the courseware design phase (both stemming from the LCS setting) and adapts them to the FSC setting. These activities cover mainly the instructional strategy, the content presentation style, and the functionality dimension of courseware modules, making sure that these dimensions are consistent in all courseware modules of the LKME. Therefore, the set-up of LKME has to be done once before courseware production by individual authors starts.

The sequence of activities contained in the LKME Set-up Phase is shown in Figure 1 using a UML activity diagram.
Audience analysis (1): Results are an overview of all LKME target user groups, in particular of the target learner groups, and a detailed characterization of these groups. This characterization also includes the social and technical setting in which the LKME users work. From the characterization, default target learner groups and a description schema of these groups with default entries for each schema category is derived.

Needs assessment (2): The result of the needs assessment is a list of educational needs that have to be met by the LKME. That is, all knowledge, skills, and attitudes to be imparted in small courseware modules are listed. The needs assessment can be conducted not only in parallel to the audience analysis but also jointly.

Specification of educational objectives of the LKME (3): During this activity, the set of default educational objectives is specified. It comprises all educational objectives required to meet the educational needs of the default target learner groups of the LKME. In addition, guidelines to select the appropriate educational goal for a courseware module according to the selected default target learner group and to their educational needs are defined.

To ease the work of courseware authors in the Small Courseware Module Production Phase and to make the small courseware modules comparable, the educational objectives should be classified according to an educational objective taxonomy. A well-known educational objective taxonomy was established, for instance, by Bloom et al. [8], [9].

Analysis of LKME content (4): The analysis of LKME content defines all topics to be delivered in the LKME that are required to meet the educational needs of the default target learner groups and to achieve all default educational objectives.

Specification of instructional strategies in the LKME (5): This activity specifies the set of default instructional strategies that are supported in the LKME and, therefore, in its courseware modules. Furthermore, guidelines are established that define which instructional strategy should be used to achieve a particular default educational objective.

Specification of learning / teaching situation (6): The result of this activity is a specification of the learning / teaching situation that the LKME has to support. The learning / teaching situation describes how learners and other user groups will work with the planned LKME, for example, the average time a learner can learn with the LKME without external interference.
Specification of requirements of the LKME (7): During this activity, the functional and non-functional requirements of the LKME are specified. These requirements assure that the LKME manages and delivers large courseware as well as courseware modules, supports courseware authors in producing small courseware modules according to the results of the activities (1) - (6) of the LKME Set-up Phase, and provides additional functionality for knowledge management in the best possible way.

Specification of the architecture of the LKME (8): The specification of the LKME architecture defines the functional components of the LKME and their interrelationships as well as the internal structure and behaviour of these components. It also defines the metadata required to describe the courseware modules. The metadata can be used, for example, to specify access rights for courseware modules, set up curricula consisting of several modules, retrieve modules based on learner queries, maintain modules, or reuse modules or parts of them.

The required metadata depend on the requirements of the LKME. Therefore, current metadata sets like LOM [10] or SCORM [11] are not sufficient as-is and should be extended to the needs of the LKME.
Specification of instructional courseware module structures (9): This activity specifies all default instructional structures of the LKME. These structures provide rules to arrange the pages of a module in order to realize the previously defined instructional strategies. In addition, guidelines are defined in order to select the best instructional structure for the target learner group, the selected educational objective, and the selected instructional strategy of a courseware module.

Specification of navigational courseware module structures (10): This activity specifies default navigational structures that supplement the default instructional structures. Furthermore, guidelines to select the navigational structure that supports a selected default instructional structure and a selected default target learner group are defined.

Component selection (11): The result of this activity is a set of compatible components that is appropriate to fulfill the LKME requirements and to implement the default instructional structures and the default navigational structures of the LKME framework. A method supporting component selection is the CAP method for the acquisition of commercial-off-the-shelf components [12].

Design of the user interface of the LKME (12): This activity results in the full-designed user interface of the LKME. It also includes the design of the user interface of courseware modules, for instance, as a set of several page templates.

Specification of the courseware module test plan (13): The result of this activity is the courseware module test plan. This test plan is applied in the content production phase to validate the quality of courseware modules regarding content, structural, design, and technical aspects.

Installation of the LKME (14): This activity is dedicated to the installation of the LKME tool environment consisting of the components that are selected in activity (11). If required, the interfaces of the components are initialized or, if they have not been existed yet, implemented.

Implementation of decision support rules (15): The result of this activity is a set of decision support rules implementing the guidelines of the LKME framework.
Implementation of templates (16): During this activity, one structural template is implemented for each specified combination of a default instructional structure and its supporting default navigational structure.

An important issue throughout the LKME Set-up Phase is quality assurance. Each product developed during the LKME Set-up Phase has to be verified in order to remove failures or eliminate problems as soon as possible. In addition, the user interface design should be evaluated with potential users of the environment. Finally, a system test and an acceptance test have to be performed at the end of the implementation activities.

2.2 Small Courseware Module Production Phase

The results of the Small Courseware Module Production Phase are small courseware modules that make existing knowledge and skills of staff members explicit and accessible for peers. The courseware authors are supposed to be subject matter experts of the software organization.

While the LKME Set-up Phase covers all important aspects of the instructional, presentation style, and functionality dimension of courseware, the Small Courseware Module Production Phase comprises all activities related to the content dimension of courseware. This exclusive focus on the content dimension is facilitated by utilizing the LKME framework.

The production of small courseware modules can start when the activity “Component selection” of the LKME Set-up phase has started. At that stage, the prerequisites for the first activities of the content production phase are available, i.e., the predefined defaults and guidelines of the LKME framework needed for the activities “Specification of courseware module” and “Structuring of courseware module content”.

Figure 2: UML activity diagram of the Small Courseware Module Production Phase
The sequence of activities contained in the Small Courseware Module Production Phase is shown in Figure 2, again using a UML activity diagram.

**Specification of courseware module (17):** Courseware production starts with the specification of the specific module. The first specification steps are a detailed analysis of the planned audience and its educational needs. The courseware author has to answer questions like “Who will work with my module?”, “How are the learners characterized?”, “What do the learners have to know or to do?”, or “What should the learners be aware of?”. The categorization of target learner groups and the predefined description schema of these groups with default entries for each category defined in the LKME Set-up Phase support the courseware author in this task. On the basis of the results of the audience analysis (1) and the needs assessment (2), the educational objective and the instructional strategy to achieve this objective are selected. The sets of predefined educational objectives and of predefined instructional strategies are provided by the LKME framework. The LKME framework also defines the guidelines used to select the appropriate educational objective and, in the following step, the appropriate instructional strategy. Furthermore, the courseware author specifies the content of the module in detail according to the educational objective and non-textual media (for example, pictures, animations, audio samples, videos) required to visualize the textual parts. Last but not least, the LKME suggests instructional and supporting navigational structures to the author. Ideally, the environment provides only one instructional and one supporting navigational structure according to the selected educational objective and instructional strategy. If there is more than one instructional and/or more than one navigational structure suggested, the author has to select the most appropriate structure based on additional parameters provided by the LKME framework or defined by the author him-/herself. When the structures are selected, the environment also provides the structural template to be used in producing the courseware module.

**Structuring of courseware module content (18):** The courseware author cuts the specified detailed content into self-contained pieces that are ideally disjoint. Such piece should be a single page in the courseware module otherwise it has to be split up further. The resulting pieces have to be arranged according to the selected instructional structure and its corresponding guidelines.

**Media production (if required) (19):** Non-textual media are developed with specialized tools. Only few courseware authors, especially when the authors are subject matter experts only, will be able to efficiently produce non-textual media for content visualization with such a tool. Therefore, a multimedia expert from a media production team should develop the required media or at least provide assistance.
Page production (20): In this activity, the courseware author develops each page defined during the activity “Structuring of courseware module content”. First, the courseware author selects an appropriate page template and opens it. In this template, he/she inserts the final text of this page. Finally, he/she includes all media produced for this page.

Implementation of courseware module structure (21): The produced pages have to be arranged according to the selected instructional structure and its supporting navigational structure. To do this, the courseware author opens the selected structural template, inserts the pages, and arranges them according to the structure defined in the activity “Structuring of courseware module content”.

Courseware module test (22): The courseware author tests the module applying the courseware module test plan. Alternatively, an external person (dedicated to quality assurance in the LKME) can conduct the test. If the courseware module is error-free, the courseware module is released and the author can proceed. Otherwise he/she has to go back to the earliest activity that introduced errors.

Specification of metadata (23): If the courseware module is released, the courseware author is prompted for the metadata of the module.

Publication of courseware module (24): Finally, the courseware module is published. After publication, everybody can access the module and use it.
3 Recommended Tool Support For Courseware Authors

IntView-KM provides processes for the *LKME Set-up Phase* and the *Small Courseware Module Production Phase*. These IntView processes only provide prerequisites but do not guarantee that courseware production is as fast and easy as required by authors that are mostly subject matter experts and not full-time professional courseware authors. Therefore, the LKME has to provide support mechanisms that simplify the work of subject matter experts that act as courseware authors. The following support mechanisms are recommended:

- Questionnaire to support characterization of the target learner group of a courseware module and their educational needs
- Decision support to select appropriate defaults
- Easy access to templates of the LKME
- Partial test automation
- Questionnaire to specify metadata of a courseware module
- Hiding LKME internal structures and functionalities from courseware authors

The mechanisms can be paper-based in the simplest environment but computer-based support should be aspired, i.e., by providing structural and page templates as well as supporting tests.

Authoring support starts with activity 17 (Specification of courseware module) of the *Small Courseware Module Production Phase*. For this activity, the LKME has to provide the user with a questionnaire to characterize the target learner group of his/her module as well as their educational needs. This questionnaire already incorporates the results from the audience analysis (1) and the needs assessment (2) of the *LKME Set-up Phase*. That is, the description schema of the target learner groups and the overview of the range of educational needs are presented as selection objects. The courseware author can then easily characterize the target learner group and their educational needs by selecting list entries. The addressed default target learner group is then derived automatically by the LKME.

Once the audience of the courseware module and its needs are specified, the courseware author has to be supported while selecting the appropriate defaults for his/her module. First, the default educational objective (for example, “Given a defined problem, the learners will demonstrate knowledge of troubleshooting procedures by describing the troubleshooting procedure with complete accuracy” [3]) is selected on the basis of the specified educational needs. In the next step, the appropriate instructional strategy for achieving the selected educational objective is derived according to the specified default tar-
get learner group (for example, technical consultants). Finally, the appropriate combination of an instructional and a navigational structure for implementing the instructional strategy is selected and the corresponding template is specified. During this multistage selection process, the decision support rules implementing the guidelines to select the defaults defined in the LKME framework are applied. The specification of the target learner group and their educational needs as well as the selections that have been made serve as input for the next selection step. In this way, the selectable standards of a specific step can be narrowed from step to step. In their simplest form the decision support rules are presented as paper-based decision tables that the courseware author has to use while making his/her selections. In a more comfortable form, the decision support rules are represented by a rule-based system that prompts the author for input and presents the selected default or template automatically.

The courseware author requires easy access to the structural template specified. It should be available within one or two clicks. The same holds for page templates defined by the LKME. All accessed templates have to be provided by the LKME in a ready-to-use and easy-to-use form. The courseware author should only have to arrange the produced pages in the right order or to insert page elements, respectively.

Some tasks of the courseware module test should be done by the LKME automatically. For example, the LKME can validate if the selected structural template was really applied. It can also validate all links to other courseware modules [13].

After the release of a courseware module, the LKME has to prompt automatically for the metadata of this module. That is, the courseware author has to fill in a form provided by the LKME. This form is designed according to the metadata set defined in the LKME Set-up Phase and provides, if available, selection lists of predefined entries.

Last but not least, the LKME has to hide internal structures and functionalities from the courseware author. For example, the author is not interested in the storage structure of the modules and of the navigational relationships of the module pages. He/she only wants to specify the module pages in the right order in a given instructional / navigational structure template. The mapping to the storage structure has to be done by the LKME automatically.
4 Existing Tool Support For Courseware Authors

Existing courseware authoring tools, learning environments, or LKMEs are not able to provide all recommended tool support.

Current authoring tools are designed to deal with the complexity of large courseware. Therefore, they are too sophisticated for subject matter experts who occasionally act as courseware authors. In addition, they are not interrelated with a special learning environment or LKME.

Existing learning environments without knowledge management functionality and existing LKMEs provide no or restricted support for courseware authors. Existing authoring support tools include:

- Templates to produce pages or tests,
- Wizards to create courses that provide easy access to templates,
- Facility to implement instructional structure templates,
- Navigational templates and guidelines for their application,
- Automatic maintenance of link integrity,
- Predefined metadata, and
- Hiding internal structures and functionalities from courseware authors.

The mapping of the existing support tools to the recommended tool support for courseware authors can be seen in Table 1. The table also shows the tool support provided by the WBT-Master infrastructure, an advanced prototypical LKME of the EU-funded project CORONET (IST-1999-11634). It implements different content structuring paradigms allowing courseware authors to compose small basic elements to a courseware module within a short time. Beside the creation of new content, the infrastructure improves the effectiveness of the authoring process by means of extensive reuse of existing contents/courseware modules.

The feature that enables fast authoring of small courseware modules is the set of content structuring paradigm of WBT-Master, which are modelled hierarchically as three levels of content abstraction:

- Basic Elements: documents, portals, questionnaires;
- Logical Composites: Learning Units, Learning Goals, Discussion Threads;
- Semantic Composites: Knowledge Cards, Knowledge Domains, Content Taxonomies.
The logical composites on the second level are corresponding to the navigational templates of IntView-KM. They combine a collection of basic elements and other logical composites into a navigable structure. They deal with inter-document relationships and do not affect document content. An individual document can be defined as a member of a number of different logical composites. Authoring guidelines on how to use these logical composites are defined in the authoring handbook of WBT-Master.

The authoring of logical composites is based on the HM-Data model that consists of addressable logical composites called Structured Collections (S-Collections). An S-Collection encapsulates members together with some internal structure of navigational topology to express the associations between members. The integrity of the links is automatically maintained by the system [14].

WBT-Master provides not only data structures supporting courseware authors but also functionality for authoring, maintenance, and delivery of the composites on the logical composite level. The authoring of the basic elements may use external authoring tools to develop different types of multimedia documents.

Basically, an author uploads documents to the remote server repository, combines the uploaded documents and other resources (created portals, questionnaires) into navigable Learning Units to develop a courseware module (activity “Implementation of courseware module structure” of the IntView-KM Small Courseware Module Production Phase). These Learning Units are further combined into a composite Learning Unit or left as-is and are published as a courseware. While doing these authoring activities, the internal structures and functionalities are hidden from the courseware authors.
Table 1: Implementation of recommended tool support in existing LKMs

<table>
<thead>
<tr>
<th>Recommended tool support</th>
<th>Existing tool support</th>
<th>Support by WBT-Master</th>
</tr>
</thead>
<tbody>
<tr>
<td>Questionnaire to support target learner group characterization</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Decision support to select appropriate defaults</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Easy access to templates</td>
<td>Templates to produce pages or tests</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Facility to implement instructional structure templates</td>
<td>Facility to implement instructional structure templates</td>
</tr>
<tr>
<td></td>
<td>Navigational templates and guidelines for their application</td>
<td>Navigational templates and guidelines for their application</td>
</tr>
<tr>
<td></td>
<td>Wizards to create courses</td>
<td>Authoring functionalities for assembling courseware modules</td>
</tr>
<tr>
<td>Partial test automation</td>
<td>Automatic maintenance of link integrity</td>
<td>Automatic maintenance of link integrity</td>
</tr>
<tr>
<td>Questionnaire to specify metadata</td>
<td>Predefined metadata</td>
<td>Predefined metadata</td>
</tr>
<tr>
<td>Hiding internal structures and functionalities from courseware authors</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

On the same composite level, WBT-Master supports the development of a training curriculum by defining a Learning Goal. The trainer/author defines a sequence of learning actions and selects resources needed to perform each action. Therefore, a Learning Goal can be seen as an instructional template, if there are predefined sequences of learning actions.
Last but not least, the content structuring paradigms also deal with another simpler authoring metaphor, i.e. gathering personal knowledge of a number of subject matter experts on a particular topic, and presenting this knowledge in a form of a training resource. An author can simply access the server to initiate a structured discussion forum on the topic and define a structure of the resultant document or courseware fragment. The subject matter experts fill this structure with relevant materials. Finally, the author converts the structured discussion (or selected components of the discussion) into a homogeneous HTML document or S-Collections (i.e. Learning Units).
5 Conclusions

Nowadays, software organizations develop large courseware mainly from scratch. They also have begun to develop small courseware modules that package small chunks of knowledge of its members. The authors of these small courseware modules are subject matter experts who act as courseware authors occasionally. Therefore, they need instructional, technical, and design support by an LKME. IntView-KM provides a methodology to set-up such an LKME and to produce small courseware modules in this LKME. Unfortunately, existing authoring tools, learning environments, and LKME provide only limited tool support for IntView-KM. Therefore, an LKME providing all recommended support tools will be developed in near future. Nevertheless, a first evaluation of the support functionality provided by WBT-Master has been done in the course of the CORONET project. The two CORONET partners that took part in the evaluation as courseware authors appraised the WBT-Master as a valuable LKME that supports occasional courseware authors in a convenient way. A second evaluation of the improved WBT-Master is ongoing. First results of this second evaluation will be available at the end of this year.


Schanda, F. “Computer-Lernprogramme: wie damit gelernt wird; wie sie entwickelt werden; was sie im Unternehmen leisten”, Beltz, Weinheim, Basel, 1995. (In German)


Advanced Distributed Learning Initiative “The SCORM Content Aggregation


Document Information

Title: A systematic approach to produce small courseware modules for combined learning and knowledge management environments

Date: Mai 2002
Report: ISEE-015.02/E
Status: Final
Distribution: Public

Copyright 2002, Fraunhofer IESE. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means including, without limitation, photocopying, recording, or otherwise, without the prior written permission of the publisher. Written permission is not needed if this publication is distributed for non-commercial purposes.