A new method for sustainable development of Open Educational Resources

Sylvia Stuurman
Open Universiteit, Heerlen, The Netherlands.
and
Marko van Eekelen
Open Universiteit, Heerlen, The Netherlands & Radboud University Nijmegen, The Netherlands.
and
Bastiaan Heeren
Open Universiteit, Heerlen, The Netherlands.

Open Educational Resources (OER) seem to be a natural fit with a distance learning university: open resources are in line with the university's mission to provide access to academic education, most material is available in digital form, and even the name of distance learning universities often contains the word "open". However, in practice, it is difficult to realize sustainable OER, especially if no existing material may be used. We propose a new method to create sustainable OER based on new educational material, and compare this method with existing models for sustainable OER. The main characteristic of the method is that OER are produced as side-effect of Continuous Professional Development (CPD). As an example of this CPD method, we describe the development of a short OER course about the programming language Scala.

Categories and Subject Descriptors: K.3.1 [**Computers and Education**]: Distance learning—*Open Educational Resources*; D.1.1 [**Programming techniques**]: Applicative (Functional) Programming—*Scala*

Additional Key Words and Phrases: Open Educational Resources, Sustainable OER, Continuous Professional Development, Distance Learning, Scala

ACM Reference Format:

Stuurman, S., Eekelen, M. van, Heeren, B. 2012. A new method for sustainable development of Open Educational Resources. CSERC'12 (September 2012), 8 pages.

Authors addresses: Sylvia.Stuurman@ou.nl, Marko.vanEekelen@ou.nl, Bastiaan.Heeren@ou.nl.

Conference CSERC '12, 10-11 September, Wrocław, Poland
© 2012 ACM ISBN 978 1 4503 1858 7 $10.00

1. INTRODUCTION

For some universities, it is too big a risk to offer all educational material for free in the form of Open Educational Resources (OER). This holds, for instance, for a distance learning university which prepares its own education material, complete with guidelines, exercises, and other elements that enable students to follow the course (in principle) without any further guidance[1]. The risk of losing considerable income when offering such material for free is very real.

On the other hand, distance learning universities and OER seem to be a natural pair. The mission of a distance learning university is to enable everyone to enjoy academic education, and to let students educate themselves at an academic level. Like other distance universities, ours has "open" in its name. In other words, the desire to contribute to OER is great. There has been an effort to prepare short courses, for free, in the form of OER [Schuwer and Mulder 2009]. This project was funded both by the Dutch government and the William and Flora Hewlett Foundation, and has now ended. A sustainable method for creating OER cannot rely fully on such funding.

The contribution of this paper is that we propose a new method for sustainable development of OER in the form of short courses, not based on existing material but on new material, without almost any extra funding. The core of the method is to produce OER as a side-effect of Continuous Professional Development (CPD) for teachers [Day 1999]. For that reason we coin our method *the CPD method*. We describe the development process of the CPD method and argue why this process is sustainable. We give requirements for choosing a suitable course subject, and we present the results of preparing a course with this method. We compare our approach with existing models of sustainable OER, and discuss advantages and disadvantages of our approach.

In this paper, we address the following questions:

—Is the CPD method suited for development of sustainable OER and how does it support sustainability?

—What are the characteristics of the CPD method to develop an OER course, and how does it differ from existing models of sustainable OER development?

—Is it possible to engage students and non-students during the development of a course, and how can they participate? What works, what does not work, in that respect?

[1] The Open Universiteit of the Netherlands is such a distance learning university.

—What are the criteria for choosing a subject to be used in this type of OER development?

—What are the advantages and disadvantages of the CDP method and the resulting type of OER?

The remainder of this paper is structured as follows. We start by introducing Open Educational Resources and existing models for sustainable OER in Section 2. We also present our CPD method to sustainable OER, including the funding, technical, and course content aspects. We also elaborate on the differences with the existing models (Section 3). In Section 4 we introduce an example OER product which is developed with the CPD method: a short course about Scala, which functions as a running example illustrating our approach. Next, we describe the development process (Section 5) of this first short course that was developed with the CPD method. Section 6 then discusses how the chosen course meets the criteria for applying the CPD method. We conclude the paper with an evaluation of the feedback from our readers, together with the numbers of visitors. Finally, we draw some conclusions and discuss alternatives.

2. OPEN EDUCATIONAL RESOURCES

There are many names for educational material that is offered for free. Open courseware, open academic resources, open educational resources are among those names. The emphasis differs slightly: courseware, academic, educational. In this paper we will use the name Open Educational Resources (OER), by which we mean material that can be used in education, and that is made available to the public for free.

The idea of OER started as Open CourseWare (OCW). The OCW movement stimulates to offer the material used by universities to educate their students for free to the rest of the world [Caswell et al. 2008]. The original idea was that the material is available anyhow and that reproduction cost in the internet age is almost zero. Therefore, universities could, in theory, offer their material without additional cost. Other forms of OER are parts from lectures in audio or video, such as the demonstration of an algorithm or physical principle, or the recording of a complete lecture (many examples of that can be found e.g. on YouTube EDU[2] and on iTunes U[3]). According to Hylén [Hylén 2006], OER may contain:

—open courseware and content,

—open software tools (e.g. learning management systems),

—open material for e-learning capacity building of faculty staff,

—repositories of learning objects, and

—free educational courses.

In practice, the costs of OER are far from zero [Downes 2007]. The William and Flora Hewlett Foundation[4] offers a grant for universities and other parties to open their content, and many universities have funded OER projects themselves. In the long term developing and maintaining OER should be possible without additional funding in order to make it sustainable. The fact that the costs of OER are not zero does not only stem from the fact that it costs time (and therefore money) to prepare existing educational resources for publication on the internet. There is also a risk involved for distance learning universities that create their own educational material: it is unknown how many students buy courses only to acquire the course materials, without the intention to complete the course. In general, the course material contains everything needed for studying the course. Therefore, it cannot be foreseen whether offering material for free will increase income (because more people get acquainted with the university), or decrease income (because less students buy courses since the course material is free).

Hylén [Hylén 2006] lists some arguments for institutional involvement in OER. First of all, in line with academic traditions one considers sharing knowledge as a good thing in itself. Second, educational institutions are funded (partly) from taxpayers' money, so it seems just to share and reuse resources developed by publicly funded institutions. Also, by reusing shared resources, the costs for content development can be cut, thereby making better use of available resources. Finally, one needs to look for new ways of making revenue, for instance by offering content for free both with the purpose of advertising the quality of the teaching institute and as a way of lowering the threshold for new students.

2.1 Sustainable OER

Important considerations for sustainability of OER are low costs both with respect to OER development and with respect to OER usage and maintenance. Examples of maintenance efforts are twofold: corrections of errors discovered by teacher or students and corrections that are necessary due to changes in the context such as the tools that are changed or new techniques that need to be covered.

Sustainability of OER can be defined as follows: "Having a mechanism in place for generating, or gaining access to, the economic resources necessary to keep the intellectual property or the service available on an ongoing basis" [Guthrie et al. 2008]. This definition is in line with the definition of sustainability of OER given by Downes: "Having long-term viability for all concerned – meets provider objectives for scale, quality, production cost, margins and return on investment" [Downes 2007].

Two aspects can be discerned in these definitions of sustainability. One aspect is the fact that an institution needs economic resources to make resources available. There are costs involved in the production of OER, even if the resources already exist. It is this aspect that is our main focus. Another aspect is that some resources need a mechanism to be available on an ongoing basis. An example of such an OER is the Stanford Encyclopedia of Philosophy[5]. This encyclopedia clearly needs to be updated on an ongoing basis, adding new philosophers or new insights. This requires some mechanism to be in place. In this paper we do not consider OER which require such a mechanism.

Downes [Downes 2007] describes funding models, technical models, content models, and staffing models for sustainable OER. We will provide a brief overview of these models.

2.1.1 *Funding models.* Since there are costs to make existing or new education material publicly available in the form of OER, sustainable OER needs to address how it is funded. Below, we sum up existing funding models.

Endowment Model. The project is sustained from interest earned on a fund. The Stanford Encyclopedia of Philosophy is an example of this model.

Membership Model. Contributions are made by interested organisations, e.g. the Sakai Foundation[6].

[2] `http://www.youtube.com/education`

[3] `http://www.apple.com/nl/education/itunes-u/`

[4] `http://www.hewlett.org/programs/education-program`

[5] `http://plato.stanford.edu/`

[6] `http://sakaiproject.org/sakai-foundation`

Donations Model. Donations are managed by a non-profit foundation. Wikipedia[7] uses this model of donations.

Conversion Model. Something is given away for free with the possibility to pay for extras. Several Linux distributors have adopted this model.

Contributor-Pay Model. The author pays (once) to the provider, who makes the contribution available for free. The Public Library of Science[8] is an example.

The Sponsorship Model. Sponsoring may have the form of advertisements or just mentioning the name. Examples are the MIT iCampus Outreach[9] (Microsoft) and the Stanford iTunes project[10] (Apple).

Institutional Model. An institution itself may pay for an OER initiative, such as MIT does for its OpenCourseWare project[11].

The Governmental Model. Here, the governmental model represents direct funding for OER projects. An example is the Dutch Wikiwijs[12] project.

Partnerships and Exchanges. Partnerships depend not so much on exchanges of funding as on exchanges of resources, where the output of the exchange is an OER.

The Ithaka report [Guthrie et al. 2008] on sustainability of online academic resources distinguishes between funding by direct beneficiaries (such as subscription payment, one-time payment, pay-per-use, and contributor payment) and funding by indirect beneficiaries (such as host institutional funds, corporate sponsorships, advertisers, philanthropic funding, and licensing).

These models for financial sustainability all concern funding for the costs of OER. Another pertinent aspect, which is not covered by these models, is a mechanism to reduce the cost of producing and maintaining OER.

2.1.2 *Technical models.* Technical models address how OER is made available, and how it is used. Roughly, Downes discerns two models:

—the OERs are used “as is” without modification, or

—resources are downloaded, adapted, and sent back to the system repository for vetting and potential use by others.

He does not elaborate on the implementation, i.e. the platform needed for these two models. Obviously, a platform which allows for downloads of content only supports the first model, whereas wikis or other community platforms allow for the second model as well.

2.1.3 *Content models.* Concerning content, Downes states that sustainability means that the content should be reusable: it should be possible to integrate it into another context. Our material is reusable by definition, in that sense: the material is designed to be studied without any extra guidance (although we do give extra guidance). Another content-related aspect is the licensing model used. He also mentions that sustainable OER might need a community around it. This last point is in line with the recommendations of the Ithaka report [Guthrie et al. 2008]:

—understanding user needs is paramount but often neglected,

—create a competitive advantage, and

—catalysing a dynamic environment for agility, creativity, risk taking, and innovation is imperative.

Often, it seems to be taken for granted that OER consists of existing material, but this is not always the case. Wikipedia is an example of an OER for which new content is created, and in the OpenER project [Schuwer and Mulder 2009] at the Open Universiteit, some of the OER courses were created out of existing material, but others were created from scratch.

2.1.4 *Staffing models.* Staffing models concern the people involved in producing OER, and making it available. Downes mentions:

Producer-Consumer model. OER is produced by professional staff. There is control over quality and content, but it requires great levels of funding.

Co-producer model. The consumers of the resources take an active hand in their production, or the production is done by volunteers. There is little control over quality and content, but this model requires much less funding.

2.2 Research on sustainability of OER

As is stated by Friesen [Friesen 2009], sustainability is structurally excluded from surveys and other forms of research on OER. In his survey, sustainability is taken into account. The Unesco report on OER [Antoni 2008] mentions the following three issues concerning sustainability: awareness rising and promotion, communities and networking of creators and users, capacity development, specifically as it relates to the development and pedagogical application of OER’s.

Friesen [Friesen 2009] found that these issues are not solved for most OER initiatives. One of the exceptions is MIT Open Courseware [MIT Open Courseware]. The material is used all over the world, as courses, and MIT benefits from the OER in several ways: it helps students and teachers at MIT itself, it increases the recognition of MIT as a leader in the subjects of the courses, and it benefits the recruitment of students. Friesen concludes that similar initiatives in OER will be expected from universities, in the future. In other words, in the future, the sustainability question might transform itself in the future, because a university without OER would receive less and less students. This conclusion means that it is vital for universities to invest in OER. In times of financial cuts by governments, methods to produce new material as OER at minimal cost are increasingly important.

2.2.1 *Massive Open Online Courses.* Massive Open Online Courses (MOOC) are a form of OER. MOOC are courses that are designed to be used by large numbers of students in an open way. There have been several initiatives in this direction. Well-known examples are Coursera and Udacity. Coursera[13] originated at Stanford University. At this moment, it offers 198 courses from 33 universities. This information was checked on October 10, 2012. While Coursera is offering courses from in principle all academic fields of interest, Udacity[14] puts the focus on Computer Science subjects only. Udacity had at the same date 14 computer science courses claiming to have 112,091 active students and instructors.

[7] http://en.wikipedia.org/wiki/Wikipedia:Contact_us/Donations

[8] http://www.plos.org/publish/pricing-policy/publication-fees/

[9] http://icampus.mit.edu/outreach/

[10] http://itunes.stanford.edu/overview.html

[11] http://ocw.mit.edu

[12] http://www.wikiwijs.nl/home/

[13] https://www.coursera.org

[14] https://www.udacity.com/

Inevitably, MOOCs are shaking the firm ground of the traditional business models of universities. It is unclear what their influence will have been if we look back in ten years. But it is extremely unlikely that they will have no influence at all. Which business model will prove to be most effective is something only the future can tell.

In any case, the fact that more and more, better and better courses are becoming available on the internet is likely to have an impact on the way universities work and probably also on the fundamental underlying business model. It is not unthinkable that OER will play an important role in that.

3. THE CPD METHOD FOR SUSTAINABLE OER

We now propose our Continuous Professional Development (CPD) method for creating sustainable OER, which is a combination of the models covered so far with some new parts. We believe that the proposed method is a viable option for other institutions as well.

3.1 Characteristics of the CPD method

Below, we discuss the characteristics of our method for the aspects funding, technical platform, content, and staffing.

3.1.1 *Funding.* We cannot rely on external funding, and want to offer OER for free, which rules out other types of funding. In some cases, there might be a small amount of funding from the institution itself, but the focus will be on minimizing the costs.

We minimize those costs in two ways. First, we don't use existing material for our OER courses, thus minimizing the risk of losing paying customers by offering the material for free. We use new material.

Second, we create OER based on new material as a side-product of Continuous Professional Development activities that are already performed by the teaching staff on a regular basis. University teachers, especially those within the dynamic domain of Computer Science, continuously have to update their knowledge. Ideally, a free course should be the valorisation of that effort: instead of keeping the acquired knowledge to themselves, university teachers could materialize their newly gained knowledge in the form of a free course.

Ideally speaking, the open courses should have the same quality as courses within the curriculum. One of the arguments for OER is that it functions as an advertisement for our courses and programs. That works best if the free course shows the same quality offered in regular courses. Another cost-related aspect is that it would be an advantage if parts of the open courses could be used in (future) regular courses.

Relating this approach to the existing funding models, we simply try to stay outside those models, and have found ways to reduce the costs. We achieve this by creating courses as side-products of efforts we already make. In this way the additional costs of production are virtually zero. Improving the course upon student feedback will require minimal costs that can still be considered part of the CPD. Such minimal costs lead to an improved image of the university and possibly further orders of related courses. Students may follow the OER as part of their own professional development activities. This may lead to ordering larger, related courses in order to acquire more thorough knowledge on related subjects.

3.1.2 *Technical.* Choosing a technical platform may seem to be of minor importance, but the availability of a platform that enables the type of OER you want to establish is vital for success. In the case of the Open Universiteit, such a platform is already available. OpenU[15], based on Liferay Portal[16], is used by the faculty of Computer Science to present courses, research activities, staff members, and news items to the public. OpenU comes with some typical social media features, such as profiles for registered users, and the possibility to blog or to use wikis [Schuwer et al. 2011]. This platform allows us to create OER in a flexible way, and simplifies interactions with users. The platform is accessible for everybody, and content can be viewed without registering. Furthermore, registration is free and also possible for non-students. Registration enables users to publish reactions on the content.

In our view there is a third alternative to the two existing models of offering OER "as is" or allowing users to download, adapt, and send back resources. This alternative is to allow users to give feedback and to interact with the content providers. A requirement for this approach is the availability of a social platform with a wiki. We consider this third alternative as an essential element of our CPD method. The feedback of students also serves as feedback on the professional development. Some subjects may not have been understood thoroughly by the teacher which may lead to reactions of students. When the developer improves the course on the wiki, the understanding of the developer improves.

3.1.3 *Content.* It is important that OER courses are different from the courses within the curriculum. The method we propose is intended for creating *new* courses, on *new* subjects.

Our approach is to engage students in an early stage, both to check whether the selected subject is of interest to the students and to receive early feedback. Such interaction is not only important for the development of OER, but it also serves a more general purpose. Interaction motivates students of a distance learning university to continue their study, which can be very hard for students with a full-time job having to study in the evening hours, or for students that are confronted with all kinds of circumstances in their personal situation.

The subject of an open course should be a new development within the field, not covered in the regular curriculum. The domain of Computer Science is a domain with fast-paced changes. Even though courses are aimed at the steady theoretical base of the domain, it is desirable to pay attention to recent changes, if only to keep students interested. To develop a new, regular course is labor-intensive (and hence costly). On top of that, we can only introduce a new course when we withdraw another one from the curriculum: short open courses offer an attractive alternative for covering hot topics without giving up the more fundamental courses in a curriculum. A short, free course offers students to get acquainted with new developments.

These requirements for the content of an OER are not covered by the existing content models. Engaging students in an early stage is in line with the recommendation of the OER Ithaka report [Guthrie et al. 2008] to understand users needs.

3.1.4 *Staffing.* The OER course is written by our own staff, but we engage students in an early stage, and ask for continuous feedback. This can be seen as a combination of the producer-consumer model and the co-producer model.

4. AN EXAMPLE OER PRODUCT

Before we introduce our CPD method for producing sustainable OER, we describe our first product created with this method, as a

[15] `http://portal.ou.nl/`

[16] `http://www.liferay.com/products/liferay-portal/`

running example. This first product is a free course on the programming language Scala. Because the material we will be discussing is targeted at the (potential) students of our university, it is at an academic level. However, there seems to be no restrictions for applying the method to other levels of education.

4.1 Scala

The design goal for the programming language Scala [Odersky et al. 2010] was to eliminate the need for different languages for different goals. For writing a script, for instance, one needs another programming language than for writing a compiled program. For programming in an object-oriented style, one needs another language than for programming in a functional style. For creating a special purpose (or domain-specific) language, one needs another programming language than a general purpose language. In many cases, a special language is needed for parallel programs as well. Scala has been designed to fulfil all these purposes.

On the one hand, these features make Scala easy to learn. The language can be learned by starting with very simple scripts, without the need to write a complete program (which always involves the use of classes and objects). Using the Read-Eval-Print-Loop (REPL), a student may start by writing single lines in Scala, and immediately see the results.

On the other hand, these features make Scala difficult to learn because of the diversity of language features. The object-oriented part of Scala for instance, has enhancements to the features of Java and C#. In Scala it is possible to create an object without an associated class, Scala has the notion of a companion object and a companion class, Scala has traits, case classes, and case objects, and all these features have to be learned. The same applies for the functional programming part of Scala. On top of that, even for students who are already familiar with both an object-oriented language and a functional language, it is difficult to learn when to use the object-oriented aspects, and when to use the functional aspects. And of course, there are features for writing parallel programs, and features enabling one to develop a domain-specific language with Scala. All those aspects require not only knowledge about syntax and the API, but also knowledge about best practices, and when and how to use each feature.

Scala is an attractive language for educational purposes because of its many features. In a course about concepts of programming languages, for example, Scala can be used to show examples of a variety of concepts. Scala could be used as a first programming language, by starting with writing scripts and explaining the concepts of variables and functions, later adding object-orientation, functional programming, and parallel and distributed programming. Scala would then become the language used in a variety of courses. The disadvantage of such an approach (besides the fact that all those courses would have to be rewritten completely) is that the main programming language of our curriculum would be a language that is not widely used at this moment, although the list of companies adopting Scala is growing, and contains successful companies like Twitter or LinkedIn.

What we did want, however, was to create the possibility for our students to get acquainted with at least several features of Scala. We therefore decided to assume knowledge of an object-oriented language such as Java as a prerequisite for a student to be able to study the course.

The course we created [Stuurman and Heeren 2012] can be read on a wiki (in 87 pages) or downloaded as a pdf document (of 124 pages). We estimate that someone with a good understanding of an object-oriented language such as Java needs about 30 hours to study the course.

The course is divided into four sections: “The Basis”, “Object-orientation”, “Functional Programming” and a project. Each section has an introduction, learning goals that direct the student, a body with text and exercises, feedback on the exercises, and a summary.

The first section (“The Basis”) introduces the language using the REPL (Read-Eval-Print-Loop), and teaches the fundamentals of the language: the syntax, its types, operators, and control structures, and how to define a function. After having studied this section, the student is able to write scripts in Scala.

The second section (“Object-orientation”) teaches the way classes and objects are used in Scala. In the first place, Scala makes it possible to write shorter, clearer code for classes than other object-oriented languages. In Scala, one can write an object directly (which means that the Singleton design pattern is part of the language). Each class may have a companion object to hold variables and methods which are the same for each object of that class. A companion object may also be used to create a factory for objects of subclasses of the class (which means that the factory method design pattern is part of the language). Scala has the notion of case classes and case objects, which make it possible to use pattern matching on objects. In this section we also introduce traits, and explain why they overcome the problems of multiple inheritance. After having studied this section, the student is able to use Scala as a “‘smarter” object-oriented language: a student will be able to write elegant, object-oriented code in Scala.

The third section (“Functional Programming”) starts by showing the difference between a functional and an object-oriented approach to a problem. We use a lamp as an example, which can be switched on and off. In an object-oriented solution, each lamp contains a state, which can be on or off, and a method to use the switch. In Scala, the code would be as follows:

```
class Lamp {
        var on = false
        def switch() = {on = !on}
}
```

In the functional approach a lamp would also be an object, but with a state that cannot be altered. The method that implements the switch would produce a new lamp, with a different (non-changeable) state. This absence of a changeable state in the functional approach eliminates state-related problems. The code would be as follows:

```
case class Lamp(on: Boolean = false) {
        def switch = new Lamp(!on)
}
```

In Scala, it is possible to define operators, so it is even possible to use the unary operator ! to act as the switch:

```
case class Lamp(on: Boolean = false) {
        def unary_! = new Lamp(!on)
}
```

The course does not ask any experience with functional programming, so the section on functional programming is introductory. We explain functions as a parameter and as a return value (together with the concepts of partially applied functions and closures), and explain how to use collections in Scala. After having studied this section, the student is able to think about how to use functional features together with object-oriented features, is able to use the functional aspects of the collections library, and can create functions that make use of the functional aspects of Scala.

The last section of the course consists of the exercises we used at the initial meeting with the students. The exercises suggest modifications to a basic implementation of the snake game. For the course, we wrote down a complete set of answers to the exercises, so whoever gets stuck can see how they can be solved.

5. THE PROCESS IN THE CPD METHOD

In this section we discuss three essential ingredients for creating OER course material using the CPD method: organizing a reading group of teachers around a topic for which professional development of these teachers is considered to be required, an early meeting with interested students, and continuous feedback by students on the course material.

5.1 Reading group

Teachers at a university have to keep their knowledge up-to-date: this updating is part of the continuing professional development, which is a responsibility of every university teacher. This is especially true for a faculty of a fast-changing domain such as Computer Science. This task can be implemented by forming a reading group around a topic of interest. Organizing such a reading group involves activities such as choosing a textbook or set of articles, preparing presentations for each other, and discussing the content in a number of sessions. To engage in such a reading group from time to time should be part of the job. In the case of the Scala reading group, we also organized a joint programming day with the participating teachers. The idea of such an event is to work out a bigger program than the typical short examples found in textbooks, and to share some practical programming skills. This is an implementation of Kennedy's community of practice model for continuing professional development [Kennedy 2005].

Instead of keeping this newly acquired knowledge for ourselves, the method we propose here for developing OER prescribes that an additional effort is made to transform the lecture notes and the prepared exercises into a short course. In our first OER course on Scala we made this effort afterwards; for future projects we suggest to make such a transformation each time after a session. In our setting, the course consists of a collection of pages in a wiki, which makes it easy to extend the content, to make adjustments afterwards, or to give feedback on each other's work.

Whether this transformation of lecture notes into course pages in a wiki is extra work or not is open for discussion: by transforming sheets and private notes into public pages on a wiki, the teacher is forced to get a deeper understanding of the subject than when preparing a presentation for fellow teachers who have read the same textbook or article. By adopting the practice of valorisation into an open course, the members of the reading group acquire deeper knowledge of the subject they study than would have been the case otherwise. Over a period of time, this means a change from broad knowledge into deep knowledge.

With the CPD method, a course can be prepared almost without additional cost. Hence, the process provides a way to create sustainable OER. For the Scala OER course, we did get support from the faculty (200 hours) to create a course of the same quality as our regular courses, with learning goals, exercises with feedback, summaries, and so on.

5.2 Meeting with students

One of the requirements for the development of an open course is the interaction with future users in the process. This interaction should start in an early stage. Below, we explain how this was implemented in the Scala OER.

Because students of a distance learning university rarely meet each other or the teaching staff, the Computer Science faculty of the Open Universiteit organizes three meetings a year for her students. Each meeting explores a popular topic. Generally, members of our staff contribute to these days, and experts are invited to give a guest presentation. On one such an occasion, we provided the students with an introduction to the Scala programming language (a topic studied by a reading group). After an introduction to the basic concepts of the language, an experienced Scala programmer explained the fundamentals of the Scala Lift web-framework. In the afternoon we organized a Scala workshop around a program that was written during the reading group's programming event.

For the workshop, we decided to provide an implementation of the classic snake game as a starting point, with only basic functionality. Students were asked to program a number of extensions, such as detecting that the snake bites itself, food disappearing after a configurable number of seconds, and so on. We had some idea of how to implement those extensions, but we did the workshop without having everything worked out ourselves. This turned out to be sufficient. Afterwards, we worked on a model solution and published this online, together with the OER course.

The meeting made it very clear that the selected subject attracted a considerable amount of attention. The first indication was the number of students registering for the meeting. Where these meetings generally attract around 30 students, the meeting with Scala as a subject attracted more than 50 students. The second indication was the fact that most students stayed until the end of the workshop, struggling with exercises in Scala. The attending students were characteristic for the students of a distance learning university: adults with a full-time job, studying in the evening hours, and many of them with a family. They not only attended this meeting on a free Saturday, but they also spent their energy on listening to lectures on a far from trivial level, and on trying to complete the exercises in Scala we challenged them to make. They were enthusiastic, and wanted to learn more about the language. This meeting made it very clear that a course on Scala would be welcomed by a part of our student population.

5.3 Continuing feedback

There are several possible ways to engage students in the creation of educational material. One possibility is to ask students what they would like to learn, for instance by organizing brainstorm sessions. Also, wikis can be used in several ways [Homola and Kubincova 2009] as a collaborative medium, for instance to enable students to prepare lecture notes in a collaborative way, to construct a library of algorithmic problems and solutions, or to collaboratively edit a textbook.

In our case, we had the choice between trying to have students construct a course on Scala by themselves, for instance using a wiki, or to start writing course material ourselves, and to use the wiki to facilitate feedback by students on the content. The first option is possible when enough students are willing to spend a lot of time in trying to learn the language using different sources, and to collaborate on a course. In our particular case, we expected that this option would not be viable. The course on Scala is not part of the curriculum, hence studying the language does not help completing the curriculum. Secondly, the language is relatively new, implying that the number of people with some understanding in the topic is limited, especially when restricted to the Dutch-speaking population. Lastly, the Scala language is very rich and supports many different features, which makes it difficult to select coherent parts for self-study.

Research on the use of wikis in education [Bower et al. 2006] shows that wikis are useful for "negotiated meaning", where a wiki page deals with one clear subject and students can comment on the content. Wikis are known to be less suitable for tasks regarding unstructured information, such as writing a new course. Such findings also point out that wikis are useful for getting feedback on course material, but less useful for creating a new course from scratch. The same conclusion was drawn from an experiment by Cole [Cole 2009], in which students were very reluctant to publish something on a wiki, especially when there was no clear simple task to perform, and in which the time invested was long in comparison to how students would benefit from the contributions to the wiki.

These findings support our decision to develop the course ourselves, to use a wiki for this, and to ask readers to give us feedback by leaving messages on wiki pages, for instance because an explanation was not clear to them, or because a mistake was found. The technical choice for a wiki makes such a collaboration possible; the platform we use makes it even more attractive because all messages are from registered users. The user profiles of students that interact give us some insights in the students studying the OER.

6. SATISFYING CPD SUBJECT CRITERIA

Our requirements concerning the subject are, as we have explained in Section 3.1.3, that it should persuade students to engage, that it should not (yet) fit in our regular curriculum (because we would create a regular course in that case, not an OER course), that it concerns a recent development, and that there are several teachers who want to gain knowledge on this subject. One example of such a subject is the programming language Scala.

6.1 How Scala meets our requirements

Scala has been around for some time now: its development started in 2001 and its first release was in 2003. However, adoption of Scala by enterprises is of more recent origin, and has grown significantly since the creators of the language received funding, which made it possible to launch Typesafe[17], a company providing commercial support, training, and services for Scala. Functional programming in general gains attention (slowly) from outside the academic domain, and Scala is one of the factors in that growing attention. So Scala can be seen as a new development, which is attractive for current and future Computer Science students.

Scala is also an attractive subject for the reading group: we teach our students to program in Java, and as teachers, we want to explore alternatives to be prepared for making a switch to another programming language. It seems too early to use Scala for that purpose, but it is certainly valuable to have knowledge of Scala if you are a university teacher with programming languages as one of the themes.

Having an open course on Scala would also make it possible to use parts of it in regular courses, for instance in a course on principles of programming languages. It would be an advantage if we could point interested students to a free course on the subject.

7. EVALUATION AND CONCLUSION

We first evaluate the Scala course itself and then we evaluate the use of the CPD method and draw conclusions.

7.1 Scala course evaluation

A number of staff members started to add feedback on the first couple of pages, to lower the threshold for students to do the same. This approach worked. Students started to give feedback on subsequent pages. Sometimes the feedback had the form of a question ("I don't understand x, is it such that y?"); sometimes it had the form of a suggestion ("I miss attention for z"). In some cases, students started to help each other by answering questions. The total number of comments, however, is somewhat modest (42), with only 9 students writing the comments (staff members not included). The last comment, at the moment of writing, is from one week ago, more than half a year after the release of the course. There were also three students who sent comments by email.

These numbers do not imply that the open course has few readers. Nielsen [Nielsen 2006] concludes that in online communities, 90 percent of the users does not actively contribute anything, 9 percent contributes something once or a few times, and only 1 percent of the users actively contributes to the community. The same applies to Usenet [Whittaker et al. 1998]: only a few users actively contribute. Given the fact that this course is not part of a community, but is a given piece of information on which people may comment, we think that the amount of feedback we received can even be considered high.

Since the course came online, we had about 15,000 page views on the course pages, in 8 months. The course contains 87 pages, which means there was a mean of about 72 page views per page (but obviously, the first pages of the course were viewed more often than pages deeper in the course). For instance, the opening page of the second part about object-orientation was visited 385 times, and the opening page of the third part about functional programming was viewed 357 times. There were 414 downloads of the pdf document containing the whole course. These numbers clearly show that the course interests people.

Also interesting to see is where our users come from. Many follow links on our own web pages, about half finds the course using search engines, and many users come from various places where the course has been advertised, including LinkedIn, Twitter, Facebook, other universities, and various other sites. Also, we noticed that people have emailed each other about the course, and followed a link in the email. Obviously, this is good news for the name and credits of our university.

7.2 CPD method evaluation

We evaluate the Continuous Professional Development (CPD) method for sustainable development of Open Educational Resources by answering the five questions we posed in the introduction.

The first question was about the suitability of the CPD method for sustainable OER. The development part of the cost is low mainly since the bootstrapping is done as part of the professional development of the teachers. Furthermore, students are involved in a very early stage such that student experience can be incorporated, even when the educational material is not fully developed. This saves time since dealing with the feedback in the raw ma-

[17] `http://typesafe.com/company`

terial stages avoids the laborious development of detailed educational material which is not used in the end. The costs of sustaining the course is kept low by asking the students to submit corrections and making one teacher responsible for corrections. When larger changes are needed due to new tools or new technological developments then a new CPD effort will be used to adapt the course.

The second question was to identify the characteristics of the CPD method we propose to develop an OER course, and how it differs from existing models of sustainable OER development. One characteristic is that we try to minimize the costs by creating a course as a side-product of activities that are carried out anyway as part of the regular continuous professional development. This approach differs from existing funding models. We think it is important to make the knowledge that individual teachers acquire when keeping up in their field of work, available to the public. Another characteristic is that we require a social platform with a wiki for our OER. This also differs from existing technical models. A third characteristic is that we develop new material for OER, that the subject should meet several criteria (not covered by the curriculum, a recent development, interesting both for students and the staff studying the subject), and that we engage students early on. This approach also differs from existing content models. A fourth characteristic is that the OER course is produced by staff members, but students are engaged in an early stage, and that we ask for continuous feedback. This is a combination of the producer-consumer model and the co-producer model.

The third question queries whether it is possible to engage students and non-students during the development of a course. What works and what does not work, in that respect? The function of a meeting in an early stage is to check the attractiveness of the selected subject. It also prepares students on what is to come, and will make it more likely that they will provide active feedback. A wiki with the possibility to add comments works very well. In this case, a forum to discuss the course did not work out. The threshold to expose yourself appears to be too high (which is supported by the fact that we also received feedback by email, with the explicit statement that giving feedback in public did not feel right). As for the cooperation of students in constructing a course in a setting without mandatory exercises, one should not set the expectations too high. The fact that students do give feedback reflects their interest, and the fact that one is able to give feedback stimulates an active attitude while studying.

The fourth question concerns the criteria for selecting a subject that is suitable for this type of OER development. We have described our requirements earlier. Another example of a subject meeting the criteria is the development of mobile applications for the Android platform. At the moment, we are following the same process with this subject: we are organizing meetings around this topic, and are reading material, giving talks, and programming.

The last question searches for the advantages and disadvantages of the CPD method, and the resulting OER. In our opinion, it is worthwhile to spend some of the time that each teacher devotes to continuous professional development to materialize the acquired knowledge into OER material. The question is, however, to what standard. In the case of the open Scala course, we were able to put in 200 extra hours to create a course that resembles our regular course material in structure and in quality: material which can be studied at home without any extra guidance. As a comparison, to create a regular 100-hours course from scratch we generally need more than 2400 hours: 200 hours of work for a 30-hours course is a low investment. These extra hours are not always available. When one would like to use this method to create an OER course without the funding for extra hours, the trade-in will be quality: a course created by different authors in a wiki, to reflect what they learned about a new subject, will not be as consistent and as logical in structure as our Scala course. We think that even the buzz it creates in social media may make it worthwhile.

7.3 Conclusion

We have presented a new method for creating sustainable OER with as main characteristic that OER are created as a side-effect of Continuous Professional Development (CPD).

We compared the CPD method to existing methods and applied it to creating an OER on the Scala language. This course was followed by enthusiastic students who improved the course via feedback on a teacher created wiki. The course received considerable attention.

The CPD method was evaluated and shown to be suited for sustainable OER development.

All in all, the experience with the Scala course confirmed the usefulness of the use of the CPD method for creating OER.

REFERENCES

Antoni, S. D. 2008. Open educational resources: The way forward. `http://www.icde.org/?module=Files;action=File.getFile;ID=280`.

Bower, M., Woo, K., Roberts, M., and Watters, P. 2006. Wiki pedagogy – a tale of two wikis. In *Proceedings of the 7th International Conference on Information Technology Based Higher Education and Training, ITHET '06*. Ultimo, NSW, Australia, 191 – 202.

Caswell, T., Henson, S., Jensen, M., and Wiley, D. 2008. Open Educational Resources: Enabling universal education. *International Review of Research in Open and Distance Learning 9,* 1 (February), 1–11.

Cole, M. 2009. Using wiki technology to support student engagement: Lessons from the trenches. *Computers and Education 52,* 1 (January), 141–146.

Day, C. 1999. *Developing Teachers: The Challenges of Lifelong Learning*. London: Falmer Press.

Downes, S. 2007. Models for sustainable open educational resources. *The Interdisciplinary Journal of Knowledge and Learning Objects 3*, 29–44.

Friesen, N. 2009. Open educational resources: New possibilities for change and sustainability. *International Review of Research in Open and Distance Learning 10,* 5 (November).

Guthrie, K., Griffiths, R., and Maron, N. 2008. Sustainability and revenue models for online academic resources: An Ithaka report.

Homola, M. and Kubincova, Z. 2009. Taking advantage of Web 2.0 in organized education (a survey). In *Proceedings of the International Conference on Interactive Computer Aided Learning (ICL2009)*, M. E. Auer, Ed. Villach, Austria.

Hylén, J. 2006. Open Educational Resources: Opportunities and Challenges. In *Proceedings of Open Education 2006: Community, Culture and Content*. 49–63.

Kennedy, A. 2005. Models of continuing professional development: a framework for analysis. *Journal of In-service Education 31,* 2, 235–250.

MIT Open Courseware. MIT Open Courseware. `http://ocw.mit.edu`.

Nielsen, J. 2006. Participation inequality: Encouraging more users to contribute. *Alertbox Columns*. ISSN 1548-5552.

Odersky, M., Spoon, L., and Venners, B. 2010. *Programming in Scala, Second Edition*. Artima.

Schuwer, R., Lane, A., Counotte-Potman, A., and Wilson, M. 2011. A comparison of production processes for OER. In *Open Courseware Consortium Global Meeting*.

SCHUWER, R. AND MULDER, F. 2009. OpenER, a Dutch initiative in Open Educational Resources. *Open Learning: The Journal of Open and Distance Learning 24,* 1, 67–76.

STUURMAN, S. AND HEEREN, B. 2012. Scala Opener Cursus (in Dutch). `http://portal.ou.nl/web/topic-scala`.

WHITTAKER, S., TERVEEN, L., HILL, W., AND CHERNY, L. 1998. The dynamics of mass interaction. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work*. CSCW '98. ACM, New York, NY, USA, 257–264.