Develop This - Education in the Age of the Internet of Things

Geert de Haan

School of Communication, Media and Information Technology
Section Media Technology / Human Centered ICT
Rotterdam University of Applied Sciences
P.O. box 25035, 3001 HA, Rotterdam, The Netherlands
geert.de.haan@upcmail.nl

Introduction
Developments in the area of Media Technology proceed very fast, and perhaps, revolutionary; where the day before yesterday the internet ruled, yesterday the web arrived, this morning it was the web 2.0 and after lunch, we enter the Internet of Things.

The main implication for ICT curricula, such as Media Technology, is that factual knowledge, book knowledge and the how-to use particular programming languages and tools are gradually becoming less important whereas the ability to keep track of developments in the research area (and not just 'on the market'), the ability to do research, how to develop conceptual solutions and how to translate concepts into software are becoming more and more important.

Human Centred Creative Technology
Media Technology is part of the school for Communication, Media and Information Science of Rotterdam University of Applied Science. MT presents itself as Human Centred Creative Technology in the sense that we educate technical engineers but not for technology itself but to create technical solutions to support real people in the real world. Naturally, this requires a more then fair amount of creativity. As a result, the MT curriculum is characterised by the common technical topics such as requirements engineering, agile design and UML but it is extended with creative techniques such as creative research tools, exploratory design and co-design, and it is extended with a focus on user-aspects by means of user-centred design, usability evaluation and accessibility engineering. To facilitate the human-centred creative technology, increasingly often special workshop forms like pressure cooker settings are used as well as design laboratory facilities including a open data lab ("city lab"), a sensor lab and a fabrication lab (see: http://project.cmi.hro.nl/2011_2012/sensorlab/).

The open data lab is the physical and virtual place where open data is accessible, for instance, about the city traffic situation and the whereabouts of items like public toilets and trees. The sensor lab houses a collection of sensory toolkits like Phidgets, Arduino and Myriane to support experimenting with sensors, and the fab lab - short for fabrication lab, houses a variety of tools to enable the actual building and prototyping of design solutions, including laser cutters, 3D printers and construction material.

Develop This goals
Develop This (DevThis) is a research and development module in the Media Technology Bachelor which tries to achieve two educational goals at once: teach third year students to do scientific research and introduce them to the technological, design and scientific developments in the area of Human-Computer Interaction (or rather ubiquitous computing, pervasive design and ambient intelligence), such as social media and co-creation, personalisation, context sensitivity and location-based services, agile development, co-design, and the use of living labs and emergent design practices.

In addition to teaching students about doing research and introducing them to new developments in the Media Technology area, DevThis is also intended as a vehicle for performing research. Also here, there are two different goals: first, students learns how to do research by actually doing research. Instructing participant to investigate particular problems or a particular problem area, enables us to focus research on the most relevant questions that follow from our research programme. For example, as part of a research programme, the group investigated how to employ ICT to solve social problems, such as how to increase the coherence among the inhabitants of socially deprived urban areas. Consequently, DevThis students were asked to develop a social, mobile and context-sensitive system (read: app) to increase social cohesion, with the results that knowledge was gathered about how to do this, see: Broos et al. (2011), Bastian et al. (2011) and de Haan et al. (2010). Likewise, when the new sensorlab/fablab was opened in 2011, students were asked to create intelligent sensory applications to help manage the laboratory building and its facilities and devices. In this case, the research projects were used to acquire knowledge about how to do this. Also, a number of demonstrators were build to demonstrate what one can do with the laboratories.

Gathering knowledge is only half of what was intended with DevThis. In education, loads of effort are invested into projects whose only purpose is learn particular skills, such as writing software, building apps or creating
toy products. After the projects are finished and the marks handed out, the products are generally thrown away or used in student portfolios. This is a clear waste of effort, creativity, knowledge and actual work, and, in order to make the best of all the sweat, the second research purpose of DevThis is to facilitate the cumulation of research and development. According to Troxler and Wolf (2010), the concept of cumulation of knowledge (or ideas, creativity ...) is common to fabrication labs aiming to serve as innovation ecologies, and indeed, most fab labs - including ours - require their users to leave behind examples and documentation for future users to re-use the design of the products made there.

Develop This setup
The basic design for DevThis was that the module consists on the one hand of theoretical instruction about ubiquitous computing, including learning to read, study and possibly write scientific papers, and on the other hand it consists of a practical component where students work on a ubiquitous computing project in which they have to develop a prototype or demonstrator and are obliged to motivate their choices considering software architecture, the frameworks used, the development methodology chosen, etc. etc.

The classical teaching takes place in a number of lectures about developments in the research area and about research methodology which are supported by regular student assignments; so much as possible, assignments are setup to allow the students to link the our teaching at the scientific- and technological levels to recognisable elements in their own 'everyday' level of their experience, which is very much oriented towards the developments on the market and in the media.

In addition to classical teaching, arrangements have been made to have students teach each other by means of lecturing their classmates, and presenting their own specific areas of expertise in workshops. Following Pask (1975) this mode of learning is referred to as 'teachback', assuming that learning is facilitated by actively explaining and working with learning material.

Finally, students learn by being actively engaged in team research projects; in the manner, students are engaged in a professional situation which prepares them for the final project and working life, and they have the opportunity to do and learn from doing their own research project.

Students work together in teams of between 3 and 6 people on a assignments. Students are requested to apply their newly acquired knowledge in the context of the open data/sensor/fablab environment, the CityLab, on problems like "how to increase the social-coherence of a community with a social mobile and context-sensitive tool" or "how to intelligently manage the CityLab facilities by means of social media, intelligent software and wireless sensor networks". The application should derive its usefulness from a user community and it involve a mobile platform and adapt to features of the environment, such as the user's location, the hardware used, sensory data or the characteristics of the user, etc. Finally, provided that software development involves programming languages and frameworks, hardware and operating system platforms, analysis and design methods, as well as plans, specifications, and progress reports, students are urged to justify and document all of their main choices.

In this way, we try to acquire three different goals in one educational module: to introduce students to scientific literature as a preparation for their final thesis, to familiarize students with research and development projects for external parties, and to have students gain experience in applying scientific knowledge in practical applications.

On the basis of a literature study, student teams develop their own conceptual solution and present these as a research poster about halfway though the module. Next, making an substantiated choice of development platform (Arduino, Apple, Android, etc.), tools, frameworks and the design and development methods (co-creation, Scrum, XP, etc.), the teams develop a demonstrator or prototype as a proof of concept. Finally, the teams present and document their findings in a report for re-use in follow-up research projects; in doing so, they receive a bonus if they also write a scientific report or a short paper alongside their regular report.

By educating science by actually doing research, we improve our students' knowledge and research capabilities - showing in their thesis projects. Also, we cumulated how-to-do-it knowledge in prototypes and manuals for re-use by different users of the CityLab. Furthermore, we have completed part of our research programme by allowing student teams to exercise their creativity to find solutions to our research questions. As a final result, a number of papers and demonstrators have been presented at conferences and workshops.

Reading materials
Research requires that students start with getting familiar with what the research area look like, what the main questions are and who is involved. From early on in the curriculum, students are gradually acquainted with scientific publications; light-weight in the first year until in DevThis all the teaching is based on scientific literature: no study books but publications only.

There is a distinction between two types of using written material. First, in lectures, a number of key scientific papers are discussed about human information processing, ambient technology, sensory interfaces, etc. as an
introduction to scientific developments in the area. Mostly, this is material that must be studied. In addition, a number of publications are used, mostly relatively short conference papers and technical papers to exemplify how the research from the overview literature may be applied to solve ‘real-life’ problems. Most of these papers need to be understood only.

Some examples of the overview papers are the following: Weiser's (1991) Computer for the Twenty-First Century is of Palaeozoic age but an excellent genesis. Being Human: HCI in the Year 2020, by Harper et al. (2008) is used as a main text about the current developments. Michahelles et al. (2007) text on RFID and the Hammersmit Group (2010) text on The Internet of Things (IoT) are used to provide an overview of the main issues and several IoT examples in a text that fits a single class meeting. Essentially, the text from Hammersmith summarises Mike Kuniavsky's Smart Things which is a bit to much (Kuniavsky, 2010). Finally, we use Sanders and Stappers (2008) Co-creation and the new landscapes of design, albeit tough reading it excels in explaining the shift from user-centred and iterative design to types of co-creation and exploratory design methods that better fit the IoT. These publications are used together with examples of publications deriving from DevThis itself, such as Broos et al. (2011) which also serve as benchmarks for students to compare their own work with. Some of these papers are literally from a previous century; this is because education rather then scientific novelty is the main concern.

It is not easy to teach theory and theoretical skills to practise oriented students. In general, among students, there is a tendency to focus in doing rather then understanding, particularly among students from the middle-level applied education. In the first year, it is a major task to get students to realise that design is not about creating what would like, but rather about creating a usable design for the target group of end-users. Comparable to this, in the third year, it is a major task to get students from realise that design is not just creating something that they might consider good but rather about creating a design on the basis of deliberate choices. Without this insight, ‘science’ is easily regarded as a waste of time rather then a set of tools and techniques to inform design and to keep up and stay ahead of the crowd. In DevThis we use three ways to alleviate the pain of science:

1. practical student-oriented assignments are used to package the theory into practical questions
2. teachback is used as a principle method to have students educate each other
3. doing ones own research and development project is used to have students do research

1. Student-oriented assignments
   The purpose of assignments is to ensure that students by being actively engaged with something, learn ho to do it. This is a fairly classical way of using assignments. Learning to write is no fun but being able to write is empowering. As such, assignments are used to ensure that students process the reading material and actively use it to prove themselves and their own insights. For example, after a lecture and papers about technologies for object and for person identification, such as RFID, QR codes and object recognition, students are requested to imagine a number of new types of applications based on (wireless) RFID tags. Not all students are able to surpass known or Google solutions and suggest to paying for ones drink at the Baja Beach Club, but others are more creative and suggest ideas such as interactive advertisement leaflets, indoor localisation, and liking everyday objects analogous to liking virtual things in Facebook.

2. Teachback
   Teachback is an essential element in the conversational theory of learning by Gordon Pask (1975). It states that explaining (or doing) what one has learned to a third party stimulates learning itself. In DevThis, students are requested to give a mini-lectures or workshops. For 20 minutes or longer if so desired, students are invited to present a lecture about tools or techniques, theoretical subjects or to provide a hand-on experience. The lecture
and workshop topics have to meet a few requirements, namely that whatever the topic, it should be something innovative and it should belong or be related to the MT curriculum. In addition, the topic should help to increases the applied knowledge of a students' classmates. In addition, it is stimulated that the topic should others to successfully complete their projects.

Most students choose to give a lecture about a topic that they know, such as eye tracking, speech recognition and generation, specific tools such as Layar for creating augmented reality application, and some choose to present their own project. Only three of sixteen teams chose to give a workshop with one about the psychology of design and two workshops about programming the Arduino sensor platform. The Arduino workshops turned out to as exceptionally useful; not only because Arduino is our default platform for application development with sensors but also because the student teams involved were much better experts in the subject matter than the teacher was.

Teachback turned out as very successful: students really like doing it and presenting some of their special knowledge. Some also appreciated their teachers more, having experienced that teaching is not that easy. In addition, teachback was especially useful for the less outspoken students as an opportunity to show something useful to their classmates and gain appreciation. In this respect, it is also a useful tool to bring hidden talent out into the open.

It may be useful to videotape the lectures to build a library, following the idea of Jim Foley's video- or weblectures (see: http://gvu.gatech.edu/foley). Currently, many universities are videotaping their most successful lecture series into for online presentation, but here the idea is not to present complete lectures but rather short explanations of particular subjects and techniques.

3. Learn to do Research
In DevThis we use scientific publications as a steppingstone, not only to introduce the state of the art in the in the domain, but also, by discussing issues like how and where to find research papers, how they are structured and formatted, which rules guide the use of references, publications are used to exemplify and explain how (applied) scientific research works.

Literature research starts with a description of what science is all about, from getting acquainted with a field of research to gathering competing ideas, creating one's own 'better' idea to investigating the value of ideas by means of empirical research, opinion research or by showing the utility of ideas in practice, in the form of a demonstrator or prototype. The process, as taught, ends with reporting the results of the investigation, by means of writing a scientific publication or presenting the demonstrator. The following is a short list of the methodological issues that are discussed.

- why do scientific research?
- types of research: experiments, questionnaires, demonstrations
- the context of discovery and the context of justification
- find publications, magazines, conferences etc. that suit one's interests
- understand the structure of publications: where is find what information
- how to read a scientific publication
- how to quick-scan and select suitable publications
- the process of peer- and metareviewing publications for e.g. workshops
- how to actually review a paper
- extracting and comparing ideas and concepts
- fraud, plagiarism and data massage
- how and when to use references and why
- where to submit one's demonstration of paper
- formatting papers and using publication templates
- standards for references, such as APA and Harvard
- create a research poster or demonstrator
- rework one's project report into a (short) paper

In the first year classes about design and evaluation, students have to realize that user interface design is not building user interfaces according to their own taste and likings but rather building interfaces that suit the end-user. In similar vein, students in DevThis have to learn that science is not about investigating their own great ideas but in systematically gathering and investigating the usefulness of ideas and solutions. A striking insight
of many participants is that science is not about finding the ONE solution for THE problem but rather to gather and investigate ideas as possible contributions to solutions.

In DevThis, students are requested to apply their newly acquired knowledge in the context of the open data/sensor/lab environment on problems like "how to increase the social-coherence of a community with a social mobile and context-sensitive tool" of "how to intelligently manage the open data/sensor/lab facilities using social media and/or sensory data".

In order to find possible answers to such questions, student teams have to proceed along a number of steps: first, they have to make a choice about the specific idea or concept they are going to pursue, which requires the teams to determine the state of the art around. For example, in order to use sensory data to manage a research lab, one has to know what data is available, what to do with such data, and which techniques to use to actually utilise the data. In addition, students have to make explicit where the concept comes from. Provided that it is not sufficient to 'have a good idea', they have to develop the concept out of ideas in the relevant literature or to find support in the literature to ground the concept. Next, the teams have to decide how to embody the concept in a demonstrator, which means making choices about design approach, implementation techniques, platform, framework, etc. with an additional argumentation to substantiate the choices. In principle but often not in practice, the next step is to implement the concept into a working application or demonstrator. Finally, the projects have to be completed by means of writing reports and papers, designing presentations, etc.

It should be noted that all students are familiar with whatever is required for each of the steps and the sub steps of the process. What is new is that all the steps have to be processed in one project, and, often more problematic, that results from a variety of activities have to be integrated to arrive at a feasible solution. In other words, students have done research before, they have programmed a variety of applications, they have worked with various sorts of requirements, and they have done some research. For practical purposes, the projects are split up in three more or less separate part: creating a concept, designing and implementing a demonstrator, and finally, demonstrating, documenting and publishing the solution.

Concept development

On the basis of a literature study, student teams develop their own ideas and conceptual solution and present these as a research poster about halfway through the module. Initially, concept presentation was only a classroom presentation but classroom presentations tend not to provoke much discussion. As such, in order to stimulate a critical attitude, we switched to public poster presentations in combination with elevator pitches and group discussions.

Peer reviewing of presentations did not work out well; even when we used the formal review tools to evaluate posters and demonstrations from the annual EACE conference (see: www.eace.net) the review results showed a strong bias towards sexy technology and presentation skills instead of the quality of the concepts and solutions.

Building access control
Indoor climate control
Ubiquitous gaming in & with the sensorlab
<text block: list of concepts for research projects>

Demonstration and publication

Next, making an substantiated choice of development platform (Arduino, Apple, Android, etc.), tools, frameworks and the design and development methods (co-creation, Scrum, XP, etc.), the teams develop a demonstrator or prototype as a proof of concept. The demonstrator is taken as a working application which demonstrates the concept behind the design. It is not required that the demonstrator is complete or ready for the market but is should demonstrate all the different functional parts of the application, as such, a Flash prototype which merely demonstrates the proposed visible workings of the concept is insufficient, but teams are allowed to take shortcuts using, for example, a local database instead of a fully functional database with remote access. The demonstrators are presented during the final project presentations of the student teams in a manner that is comparable to presenting demonstrations at workshops or conferences. For later use, the presentations are videotaped and some of these are presented on Youtube (see, for example: Toolenaar, 2011).

In most projects, students chose to continue to work on topics that they are interested in and often these are topics that they have worked on before, for example in education projects or in the non-obligatory programme. In this way, students have the opportunity to work on topics that they are really interested in and, be engaged in preparing for their final thesis research.

Finally, the teams have to write a report about their research and development project, summarising what they have done and which choices were made (and why!) in the process. In addition, the teams may pursue for a
Discussion
This paper describes the 'philosophy', the intentions and the setup of DevThis as an education module to introduce novel developments to Media Technology students, alongside learning to do research. Students learn about new developments in research projects which bring together all that has been learned in the preceding years in one module that prepares students for their individual research projects for their final thesis. In addition, the paper focuses on the three types of learning that are used in the module: classical learning in combination with student-oriented assignments, students teaching each other in teachback, and in actually doing research as a means to learn how to do research.

DevThis evolves and education modules around it are developing. The content of the DevThis module moves towards the use of sensors and interactive installations and will increasingly make use of open data and computational intelligence.

We assume that it is not the hardware or technical possibilities which let us down but rather we assume that the main obstacle is our lack of creativity. Facilities to enable students and staff with easy and affordable ways to experiment with ubiquitous computer applications and facilities to support creative fabrication are in place but, by themselves, they are not sufficient. According to Troxler and Wold (2010) and van der Helm et al. (2010), to enable creative design is not something that automatically emerges when a design lab is put in place; rather, in addition to the environment, this requires building a community of designers who are able to stimulate and build upon each others' work. Consequently we need to work on creating a lab community and to find better ways to creatively cumulate design knowledge - not only between students from different years but particularly between lab users with different backgrounds.

DevThis continues to develop both to improve and among the most prominent improvements to be made is the need to extend using the laboratory facilities into developing a creative design community that may extend and reuse the cumulated insights and knowledge from DevThis and other research and design undertakings.

References

this is a draft version of november 2012

to do:
add pictures of posters, concepts, demonstrators if space is available
format the text blocks
proofreading and detailed formatting