Educating Creative Technology for the Internet of Things - Research and Practice-oriented Approaches Compared

Geert de Haan
Mankind Inc.
Twello, the Netherlands
geert.de.haan@upcmail.nl

ABSTRACT

Develop This (DevThis) is a research and development module in the Media Technology Bachelor which tries to teach third year students to do scientific research and introduce them to the technological, design and scientific developments in the area of Ubiquitous Computing and the Internet of Things (IoT), such as context sensitivity, location-based services, co-design, and the use of emergent design practices. Ambient en Pervasive Design (AmbiPerv) is a second-year course following a practice-oriented vision on Human-Centered Creative Technology to educating Internet of Things techniques. In AmbiPerv, students learn to work with sensors and effectors in IoT applications. The paper evaluates the research versus the practice-oriented approach to educating Creative Technology for the Internet of Things and compares the results in terms of research opportunities, student motivation and effort versus outcomes to make recommendation about how to teach Creative Technology for the Internet of Things.

Author Keywords
Education; ubiquitous computing; ambient intelligence; pervasive computing; sensory applications; embedded design; curriculum development; teaching methods.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous. See: http://www.acm.org/about/class/1998/ for more information and the full list of ACM classifiers and descriptors.

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 know-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.

Media Technology is a curriculum for professional education in the school for Communication, Media and Information Science of a Dutch University of Applied Science. MT presents itself as Human-Centred Creative Technology in the sense that the curriculum educates technical engineers but not for technology itself but to design creative technical solutions to support real people in the real world. As a result, the MT curriculum is characterized 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-centered design, usability evaluation and accessibility engineering. To facilitate the human-centered creative technology, increasingly often special workshop forms like pressure cooker settings are used as well as design laboratory facilities including a open data lab (also 'the City Lab'), a sensor lab and a fabrication lab (see: http://project.cmi.hro.nl/2011_2012/sensorlab/).

DEVELOP THIS

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. Asking student participants 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: [1,2,3]. 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 accumulation of research and development. According to Troxler and Wolf [4], the concept of accumulation of knowledge (or ideas, creativity ...) is common to fabrication labs aiming to serve as innovation ecologies, and indeed, most fab labs users to leave behind examples and documentation for future users to reuse 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, to study, and possibly, to 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 [5] 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 an assignment. Students are requested to apply their newly acquired knowledge in the context of the open data/sensor/fablab environment, the City Lab, on problems like "how to intelligently manage the City Lab 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.

**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 Computer for the Twenty-First Century [6] is of Palaeozoic age but an excellent genesis. Being Human: HCI in the Year 2020, by Harper et al. [7] is used as a main text about the developments. Michalelles [9] text on RFID and the Hammersmit Group [9] 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. Finally, we use Sanders and Stappers [10] 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. This publication is used in conjunction with single subject publications like Ed Chi’s blog entry on Living Laboratories: Rethinking Ecological Designs and Experimentation in HCI [11] which
example of a location-based service in which tilting a device is the 'novel' feature. Crepaldi et al. [14] Managing Heterogeneous Sensors and Actuators in Ubiquitous Computing Environments is used as a typical example of what a technical project paper would look like. Crepaldi's text is used together with examples of publications deriving from DevThis itself, such as [1, 2], 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 than 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 than understanding, particularly among students from the mid-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 than 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. student-oriented assignments package theory in practical challenges
2. teachback is used as a method to have students educate each other
3. learning to do research lets students contribute to research programs

Teachback

Teachback is an essential element in the conversational theory of learning by Gordon Pask [5]. 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 increase the applied knowledge of the students’ classmates and it is stimulated that the topic should help others to complete their projects with success.

Most student 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 presenting some of their own 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. This is a list of some of topics of student lectures:

- Arduino development
- Psychology and design
- Semantic web
- Games using sensor data
- Digital map techniques
- Face recognition
- Nanotechnology
- User profiling with sensors
- Object recognition

Learning to do Research

In DevThis we use scientific publications as a stepping stone, 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, etc. Most importantly, 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 ones 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, concludes with reporting the results of the investigation by writing a scientific publication and presenting the demonstrator. The following is a short list of the methodological issues that are discussed:

- Why do scientific research?
In DevThis students are requested to apply their newly acquired knowledge in the context of the open data/sensor/fablab environment on problems like "how to increase the social-coherence of a community with a social mobile and context-sensitive tool". 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 parts: 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. These are some of the concepts, developed for the research projects:

- Wifi broadcasting onsite
- Access control to buildings and site
- Mobile money with NFC
- Indoor climate control
- A Bluetooth remote for old TV's
- 3D aerial photography
- Herba sense - sensing plant growth
- Ubiquitous gaming in & with the sensorlab
- Face recognition & authentication

**Demonstration and Publication**

Next, making a 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: [15]).

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 bonus mark if they succeed in writing a scientific report or a short paper alongside their regular report. In a number of instances, these papers and demonstrators have been accepted for scientific presentation [1, 2] or presented as part of a paper presentation [16].

Discussion
This section described 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, much like business cases, which introduce students to develop ideas for the commercial market. It is expected for the near future that application which utilise sensory data, location based services, tagging real world objects, and open data will increase their market share and will become a novel design focus [7][9][17]. Consequently, the content of the DevThis module shifted year by year and came to include, for instance, the use of sensors and interactive installations and the 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 [4] and [18], 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.

There is also one particularly hard issue in relation to successfully developing a module about new developments in HCI and Media Technology: the irresistible tendency to extract interesting topics out of DevThis and move them into the first and second year curriculum, which, for some, surpasses the need to create ones' own educational material. This implies that DevThis content continuously required being updated at the cost of considerable amounts of time and effort; especially with a curriculum committee which actively stimulates colleague teachers to take out as much as possible of the content into lower years. As an example, about 60% of the original DevThis teaching material has been taken out to form the MT research methodology curriculum...

In various ways, the DevThis modules have been successful in teaching students how to do research in combination with doing research about into topics related to the new developments in the field. Compared to a few years ago, students are better equipped to be professionally active in the mobile and ubiquitous computing area, and they are much more aware of the tools, techniques and frameworks for developing mobile and ubiquitous computing applications [16].

Furthermore, within the major programme, it is apparent that students are better equipped with knowledge about the mobile and ubiquitous computing area. The students have significantly improved in terms of basic knowledge of the research area and in terms of the ability to use scientific sources and methods in their projects.

**Ambient and Pervasive Design**

The ambient and pervasive design course is a second-year course which exemplifies how a course like DevThis which demonstrates various aspects of the vision of Human-Centered Creative Technology, may be used to guide the design of other, and generally, less advanced courses. In this course, students use the Arduino ([19]; see: http://arduino.cc) toolkit to experiment with the use of sensors and effectors in computers applications, either on the connected pc or as stand-alone applications on the Arduino board.

The purpose of the Ambient and Pervasive Design course is to let students get acquainted with the concept of the Internet of Things (IoT) within the areas of HCI and Media Technology. The main focus is to get acquainted with applications that make use of context-sensitive information, such as location-based services, personalized advice, adaptive software, restfull services, and especially, sensory interfaces and interactive installations.

The purpose is not to provide a complete as possible overview of Ubiquitous computing or to educate students in doing research but rather to educate student in the basic technology to build Internet of Things applications. For a number of years, the Media Technology curriculum has focused on interactive websites and, later on, on mobile phone apps. In The Ambient and Pervasive Design course, the aim has been to extend the focus to Internet of Things applications. Internet of Things applications may, of course,
employ internet and wireless technologies, but, in principle, they aim at stand-alone applications. Apart from the affordable availability of the Arduino technology, the ability to create stand-alone interactive installations has been a major reason to choose the Arduino platform.

As a second year course, AmbiPerv is not intended to teach students how to do research, as is the case in DevThis, which prepares students for the research in their final thesis project. Neither is AmbiPerv intended to provide a complete overview of the field of Ubiquitous Computing or the Internet of Things. The single most important goal is to make students familiar with IoT technology, and, as such, examples from the research area are presented in an accessible way such as to motivate students and to illustrate the technology.

**AmbiPerv Set-up**

First, students are introduced to the concept of IoT using video material from Bassett et al. [20] and other examples. IoT is related to the 'Make' movement and a link is created with the Arduino development platform using a well-known presentation by Massimo Banzi, one of the architects of the Arduino platform, at the Ted conference of 2012[19]. Throughout the course, theory and background are presented in a highly accessible way, choosing for pictorial examples and video material, rather than by reading scientific papers or understanding theoretical explanations.

In similar vain, students are advised but not required to do any background reading and they are referred to some of the easier accessible texts of the DevThis course, such as the Internet of Things Comic Book [21] and the report from the Hammersmith group [9].

Working with the Arduino platform is introduced with the building of the 'Blink' application with on an Arduino board. A LED-light is stuck on an Arduino board and a script is installed and ran on the board that makes the LED blink at a steady rate. Apart from introducing the Arduino platform, students are also taught about basics of electronic engineering. Experimenting with the Arduino kits is supported with references to Arduino crash courses and resources texts (e.g. [22]). In addition, some of the students who had already followed the first-year optional FabLab course sometimes acted as student-teachers. This is a general list of the main lecture topics of the AmbiPerv course:

- Introduction
- IoT and what the future might look like
- What can we do: Arduino and electronic engineering
- The transition from web 2.0 to web 3.0
- The Semantic Web: machines talking to machines
- IoT and product conceptualisation
- IoT and design methods
- Roundup

Subsequent lessons introduce students to the basic concepts of working with electronics and stepwise refine the blink-application to more advanced techniques, including reading sensor values, activating effectors, transferring sensor information to a pc over the serial port with PHP and Node.js (see: http://semu.github.io/noduino/), and presenting the results in a graph that may be accessible over the internet, using the 'processing' programming language (see: processing.org). Lessons are designed such that students are not merely taught about how to get things done but rather where to find additional resources and examples to find their own way.

In addition to introducing the concept of the Internet of Things, further explanations deal with various subtopics in more depth such as a brief history of Ubiquitous Computing, a short explanation of the ideas and techniques of the Semantic Web as well as Linked- and Open Data, and a variety of design issues related to designing, building and fine-tuning sensory and intelligent applications, like co-creation and co-design [10], the notion of agile design, and IoT-specific design techniques such as mashups and exploratory design [23].

In parallel to introducing Arduino and the Internet of Things, students are requested to perform three assignments of increasing difficulty. Assignments start with copying or rebuilding the Arduino applications which deal with the basic techniques for building IoT applications in general:

- Connecting the development board
- Reading sensor values
- Writing to effectors
- Transferring data
- Presenting information elsewhere

Each of the techniques is associated with an assignment task that requires students to enhance or adapt the example application in such a way that they need to understand the technique in detail rather than merely copying the programming code. Each of the three assignments should be videotaped to qualify for the project to conclude the course.

Next, once all preparatory projects have been finished, students are requested to develop their own concept idea for building an IoT/Ardunoi application; in this case, to sense whether the atmosphere is healthy, guided by a worked out example (see e.g.: http://learn.adafruit.com/tmp36-temperature-sensor/overview). For the concept it is required that multiple sensors are used in combination with social media in order to influence the behaviour of people. The idea is that sensors measure things like temperature and CO2 and other sensory measures, together with derived measurements of, for instance, the noise and crowdedness in a room, in order to calculate some health-indicator that may be used to advise people, for instance, by SMS, to move away or to open a window. To develop this concept it is necessary to do both theoretical research, to establish
what to measure for what purpose, as well as technical research, to establish how to measure and implement the concept.

Finally, students are requested to actually build a prototype of the concept in the form of a demonstrator to present the feasibility of the concept. By insisting that the concept is actually build, students are forced to determine and to finetune (read: 'to tweak') the measurement scheme to something that will actually work in reality. Students finish their project by videotaping the results of their demonstrator which is handed-in for evaluation.

**AMBIPERV VERSUS DEVTThis**

In the ambient and pervasive design course, three aspects of the DevThis approach were skipped to address the fact that AmbiPerv students are less advances in their studies then students are who follow DevThis, and because an introduction to the basic (programming) techniques for building IoT applications replaces the research goals of DevThis.

First, there is no need to provide a more or less complete overview of the new developments in HCI and Media Technology, such as agile and co-design methods, intelligent interfaces and metadata. Instead, rather 'tangible' examples are presented in an easy to grasp manner, like YouTube films.

Secondly, students are not requested to create a design based on extensive theoretical research; they are not required to read scientific papers or to argue why a particular solution has been chosen, and neither is there a requirement to write a scientific paper alongside the end-result in the form of a prototype application.

Thirdly, it is not required to follow and evaluate a particular design method and consequently, it is not possible to investigate the utility of various design approaches. Instead, students are led through a development process in which, in a step by step approach, all the essential or basic techniques to create an Internet of Things (IoT) application are presented, explained and evaluated. Student may focus on building without being bothered with scientific research.

**DISCUSSION**

The design of Ambient and Pervasive Design differs in various aspects from its predecessor Develop This. These changes have been introduced to meet the demands of presenting an IoT course in an earlier year in the Media Technology curriculum in combination with the wish to focus only on providing students with the technical necessities to start building Internet of Things application rather then providing a more-or-less complete scientific overview of the area in combination with opportunities for applied research.

A major difference between the courses is that DevThis is about four times bigger then AmbiPerv, allowing more time to present theoretical background information and teaching of research methodology issues. Nevertheless, it is remarkable that students in both courses succeed in developing suitable concepts and demonstrators. The demonstrators of DevThis are technically more advanced and seem better designed but the underlying technology does not differ too much. This supports insights from e.g. the EYA festival (see: www.eu-youthaward.org) that, in order to create interesting IoT and social media apps, does not seem to require too much beyond basic technical knowledge.

On the basis of earlier research [16] we concluded that the combination of research and development in DevThis resulted in a noteworthy improvement in both, the practical and in the theoretical knowledge of students. In Ambient and Pervasive Design it is remarkable how much better as well as easier these students perform in comparison to the DevThis students who attempted to develop Arduino applications, a few years ago. At the time, there was no teaching support for Arduino programming and the students had to do it all by themselves. Given that a little bit of explicit technical instruction makes it hard to distinguish second year from fourth year students clearly indicates that a practical hands-on approach is worthwhile.

Furthermore, whereas DevThis is focused on providing opportunities for research, for AmbiPerv the opposite is true. It is not surprising that AmbiPerv does not provide new insights that are worthwhile of publishing. It may be, and this requires further investigation, that students who are technologically better equipped, early on during their studies may provide more interesting research results, later on in their studies. Unfortunately, there is no extensive evaluation and performance data available about the differences in long-term outcomes between the two courses, and only limited conclusions may be drawn. Regarding research opportunities in AmbiPerv, we must conclude that the available time and knowledge are insufficient to establish useful relations between the science of the research field and the practical assignments of the students within the course.

Finally, a very prominent difference between the two courses is that students are much better motivated for the technology-oriented and hand-on approach of AmbiPerv then they are for the theoretical and research-oriented approach of DevThis. Students really seem to enjoy building things with their hands as opposed to studying abstract research papers. In this respect, we may conclude that a practice-based approach has a noticeable and very positive influence on student-motivation. This result presents some evidence for the notion that education should support rather than hamper the motivating value of creative exploration in children’s learning, as suggested in [24].

Note however that student motivation should not be taken as an isolated factor. First year students tend to dislike non-technical courses, whereas advanced students and graduates particularly appreciate their non-technical luggage such as
creative design techniques or professional support training. However, given that there is both an immediate improvement and given that a well-established theoretical basis does not seem to be necessary to design interesting IoT applications, it seems safe to state that, beyond motivation, presenting students with the basic techniques earlier-on in their studies will have a positive effect on the levels of creativity and technical capabilities in future professionals.

To conclude, we may attempt, albeit with caution, to formulate two recommendations. First, compared to late exposure, it seems recommendable to let students become familiar with Internet of Things techniques early on during the study curriculum; if not for motivational reasons then certainly for expected outcomes.

Secondly, it seems recommendable for educating Creative Technology Design to focus on technical and hands-on training instead of focusing on first completely mastering the abstract and theoretical knowledge that we used to see as an essential prerequisite to design meaningful Internet of Things applications.

ACKNOWLEDGEMENT
This paper is based on assignment at a University of Applied Sciences. I am indebted to my former colleague Kimmert Zelle for co-creating, or perhaps, co-designing Ambient and Pervasive Design and its’ predecessor Ubiquitous Computing in the Mobile Life minor.

REFERENCES


18. Toolenaar, J. Dev This - Socialize This Live Demo. Available from: http://www.youtube.com/watch?v=n27gsE8Hg (2011)


