

# Mobile Programming – Enhance Teaching Informatics

**Ludger Humbert**, *humbert@uni-wuppertal.de*  
Bergische Universität Wuppertal, Applied Computer Science  
Gaußstraße 20, 42119 Wuppertal, Germany  
Willy-Brandt-Gesamtschule, Fachgruppe Informatik  
Am Friedrichsberg 30, 59192 Bergkamen, Germany

**Peter Micheuz**, *peter.micheuz@uni-klu.ac.at*  
University of Klagenfurt, Software Engineering and Soft Computing  
Universitätsstraße 65 – 67, 9020 Klagenfurt, Austria

## Abstract

Teaching informatics allows students to work out solutions for given problems. If this is done with typical elements in an object-oriented manner, they will find out objects and discuss several possible ways to structure a solution by combining the work of these objects. After this there is a sequence in which objects interact to solve the given problem. Last but not least students have to test their solutions and discuss them. Implementing at sourcecode level is essential to acquiring basic insights and to help students to find out that “code rules”. Nowadays students grow up in an environment, where they use mobile phones and not desktop computers in everyday life. Mobile phones allow students to code wherever they are and at whatever time they wish. They may spend time on programming, testing and modifying their solutions without paying any fees. To provide proof of this concept, a regular course at secondary II level – K11-grade – with 29 students (19 female and 10 male) was given the opportunity to work with mobile phones alone, without access to desktop computers.

## Keywords

Mobile Programming, History of Teaching Informatics, Informatics Systems for Programming

## INTRODUCTION

Rather a long time ago programming was an art and only some folks – wizards – knew how to do it. Nowadays – since informatics systems are becoming ubiquitous<sup>1</sup> – we find informatics introduced in school as mandatory subject. Writing source code – as it remains the heart of modelling – is mostly still done with “real computers”. This means it is carried out with PCs although there are many other devices and gadgets which can be programmed as well. For example when it comes to databases teachers give students some data to play with, instead of using real data, which are available on every mobile phone – this data may consist of visited locations, contacts, played music etc.

We have to take care of real world phenomena of our students to allow them to get involved in modelling and extending what they find useful in their world. Those problems do not remain the same over time, so every teacher is able to provide examples of the modelling of real life problems with students as well as with issues of security of and reacting to emerging failures.

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it. [Weiser, 1991, p. 94]

---

<sup>1</sup> The term ubiquitous is somewhat misleading – omnipresent and calm would be much better, as Weiser pointed out – cf. [Ishii, 2004, p. 1300]

## HISTORY

### Yesterdays experience

The first time modelling and programming were practiced in German schools was in the Gesamtschule Berger Feld comprehensive school in Gelsenkirchen in 1969. When visiting informatics lessons there in the seventies, you found students working on an editor for the only informatics system in their school. They **needed** an editor for programming this system. Elements of the courses were documented and published as first textbooks for teaching informatics in school.

The primary idea – the “need” for a concrete solution for a real problem as pointed out – was not communicated, so many examples were produced and students would – should – ask: “Why should I model and program this function – it is already there in my pocket, in my application, ...?” If we look at textbooks on teaching informatics in secondary schools we will find that they do not fulfill the basic didactical requirement mentioned above: the need to “invent” something to solve a concrete real life problem.

### Todays informatics systems

We have to take into account that students (and even pupils and children at kindergarten-level) are using full fledged informatics systems in everyday life and do not think about the hypothetical chance to program their devices. Informatics as a subject has to enable them to use their systems in unforeseeable ways – i.e. to program on and with these systems. We have to take a look at solutions to program such devices without using any other systems and thus enabling the students to realize **mobile programming**.

### Mobile Programming – realizing object-oriented models with Python

At the conference EuroPython in 2004 a first reference of an upcoming implementation of Python for mobile programming – namely for the Symbian S60 operating system was presented. The implementation is based on Python 2.2 and is constantly being improved to enable functions specific to mobile phones. Not only the core functions have been extended but also several interfaces for Python have been published (cf. Laurila [2008]).

- internal databases
- accelerator functions
- bluetooth control
- setup and use WLAN
- control the inbuild camera
- ...

First discussions on mobile programming in schools in informatics courses led to a master thesis Carrie [2006]. This thesis shows the possibility of mastering programming with the mobile itself – without the use of any other informatics system. Carrie showed this by implementing first elements of a class library which is in use in regular desktop-based courses in German schools. The class library named “Stifte und Maeuse” was developed for German students to give them the chance to get an insight in a small but useful library as a didactically unique wrapper for far too complex programming language based class libraries. The developed class library is available in some programming languages – namely Oberon-2, Object Pascal, Java, and Python. The basic classes are so constructed as to allow students to construct project-based extended versions. Carrie ported some classes of this library for mobile phones so students are able to program with and on the phone. Moreover Carrie [2006] documented some examples of using elements on the mobile phones, which are not available on desktop computers. With Tuulos [2007] a textbook for mobile programming was published. It shows more than 100 mostly very short examples of how to use a mobile for programming purposes.

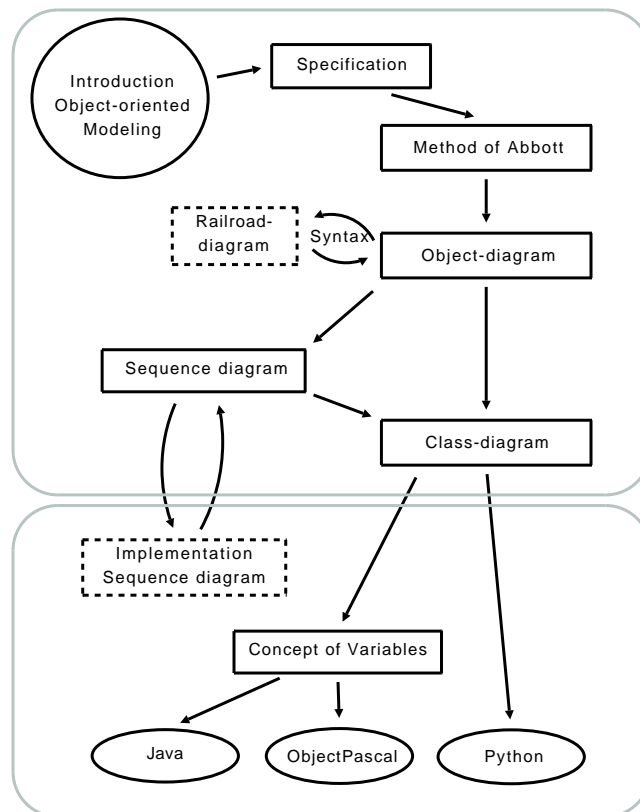


Figure 1: Teaching approach

### Modeling and implementing over time

We will not fully discuss the “generations” of teaching informatics – same procedure as trying to classify the development in technical informatics by focussing on generations of hardware. A small sketch may show the principal direction. We argue that the named effect with teaching technical informatics will take place by discussing the direction from at first technical to a nowadays more modelling approach in the sense of using informatics systems to solve (given) problems.

Nowadays the focus is set on modelling – to make it clear – primarily on object oriented modelling. But this is the backend – we have to consider the above mentioned didactical idea – to motivate students one has to look at their lives, at their using concrete informatics systems and to point out how informatics will help them to use these gadgets in a more sophisticated manner. In this manner, they learn to program their devices and use self-realized solutions for real life problems. Only informatics teachers’ help may enable students to do so – other teachers may focus on using devices (from desktop computers to whatever else).

For some years the authors of the paper at hand did some practical research to construct elements which may be used in courses at secondary II level to fill the gap between real life problems and the interest in implementing. Some of these problems were published at the biannual German conferences **Informatik und Schule (INFOS)**.

When preparing Boettcher et al. [2007] we found that it is possible to implement a solution in the programming language Python without using the term **variable** as shown in figure 1 – the other languages used to implement the solutions in some of our schools require this term (and so the concept), so one has to discuss – besides object-oriented thinking – a different concept to structure data. This puzzles novices, who are first taught to express their modelling aim in objects and later in classes – but variables? The term variable is obsolete in object-oriented modelling – so we must look for implementations which avoid using this term at all.

## From problems to implementation

We agree with Denning, who states that “Students need to see from the beginning how to connect abstractions to actions. There is little joy in worlds of pure abstractions devoid of action. The practices of programming and systems make this connection” [Denning, 2004, p. 20]. In the beginning there is a problem, for example students are confronted with a situation in which they are supposed to provide the music to be played at a party. They are also told to take into account that the guests’ favourite music styles will probably differ. As far as concrete people are concerned, this problem is then solved by hand-selecting a few pieces of music – so the students can get an idea of the context. As a second step they will use elements pointed out by Abbott (cf. Abbott [1983]) to find objects, attributes, and messages (methods). This leads to terms regarding object oriented modelling but also to well known representations as object-diagrams and, later, sequence diagrams.

This leads to terms regarding object-oriented modelling but also to well known representations as object diagrams and thereafter sequence diagrams. After being confronted with the notation: `object.method(parameter1,parameter2,...)` they are able to formulate their sequence-diagram as Python-program (cf. figure 1).

(0) Operating System	Symbian Series 60 (Symbian Series 60 (S60)) User Interface Quartz (UIQ)
(1) Filemanager	Y-Browser
(2) Programming language	Python for S60 Python for UIQ
(3) Classes—SuM	<code>sum*.sis</code> cf.Humbert [2007]
(4) Integrated Development Environment	Python editor (Ped)

Table 1: Mobile Programming – Enabling technology

While our project was going on, Wahlig [2007] published a version of Ped (Python editor) for the Operating System (OS) S60 3<sup>rd</sup> edition of mobile phones – a kind of Integrated Development Environment (IDE) for Python on the mobile that fulfils our project requirements best. To sum it up: when it comes to mobile programming we used the elements which are documented in table 1. This project could only be carried out, because our school received thirty mobile phones (three different types – namely: 5500 Sport, E61i, and E65 – Nokia) to run the course.



Figure 2: S60 – Interface to input data

The mobile phones arrived; the students unpacked them, charged the batteries, and installed elements which allowed them to use the devices for mobile programming. So the students could work with the local file system on their phones. Furthermore this solution makes it possible to send and receive files via Bluetooth. They installed the latest Python-interpretter and the Python-shell cf. table 1—(2).

## Adults and teachers are puzzled

If you look at the “keyboard” in figure 2 you may never want to type in a complete program – no matter what programming language you prefer to use. But not so our students – no one asked how to type in the found solutions we had to implement for our first problems.

## CONCLUSION

By no means has this been our first experience, but this project has shown that it is possible to use mobile programming in regular teaching informatics at secondary II level just for beginners. The students are very motivated and realize solutions with mobile phones primary in so called homework, which is carried out not only at home but also at the busstop, as a regional newspaper states: "Wenn die Informatikschüler der Willy-Brandt-Gesamtschule an der Bushaltestelle stehen ..."<sup>2</sup>

## REFERENCES

- Abbott, R. J. (1983) Program Design by Informal English Descriptions. *Comm. ACM* 26 (11), 882–894.
- Boettcher, D., Grabowsky, A., Humbert, L., Poth, O., Pumplün, C., Schulte, J. (2007) Ada – dieser Zug hat Verspätung. In: Schubert, S. (Ed.), *Informatik und Schule – Didaktik der Informatik in Theorie und Praxis – INFOS 2007 – 12. GI-Fachtagung 19.–21. September 2007, Siegen*. No. P112 in *GI-Edition – Lecture Notes in Informatics – Proceedings*. Gesellschaft für Informatik, Köllen Druck + Verlag GmbH, Bonn, pp. 217–228,  
<http://www.ham.nw.schule.de/pub/bscw.cgi/1001803> – last visited: 20<sup>th</sup> January 2008.
- Carrie, R. (2006) Einsatz mobiler Informatiksysteme im Informatikunterricht der gymnasialen Oberstufe. Hausarbeit gemäß OVP, Studienseminar für Lehrämter an Schulen – Seminar für das Lehramt für Gymnasien Gesamtschulen, Hamm,  
<http://www.ham.nw.schule.de/pub/bscw.cgi/315319> – last visited 23<sup>rd</sup> December 2007.
- Denning, P. J. (2004) The Field of Programmers Myth. *Comm. ACM* 47 (7), 15 – 20,  
<http://cs.gmu.edu/cne/pjd/PUBS/CACMcols/cacmJul04.pdf> – last visited 1<sup>st</sup> January 2008.
- Humbert, L., August 2007. »Stifte und Mäuse« on Symbian S60 3<sup>rd</sup> edition. Software available at  
<https://haspe.homeip.net/projekte/ddi/browser/python/S60-SuM> – last visited 3<sup>rd</sup> January 2008.
- Ishii, H. (2004) *Bottles: A Transparent Interface as a Tribute to Mark Weiser*. *IEICE Transactions on Information and Systems* 87 (6), 1299 – 1311,  
[http://tangible.media.mit.edu/content/papers/pdf/bottles\\_IEICE04.pdf](http://tangible.media.mit.edu/content/papers/pdf/bottles_IEICE04.pdf) – last visited 31<sup>st</sup> December 2007.
- Laurila, J. (2008) PyS60 Library Reference – Release 1.4.2 final. Manual – python symbian s60 – nokia corporation, [OpenSource.nokia.com](http://opensource.nokia.com),  
[http://downloads.sourceforge.net/pys60/PythonForS60\\_1\\_4\\_2\\_doc.pdf](http://downloads.sourceforge.net/pys60/PythonForS60_1_4_2_doc.pdf) – last visited 27<sup>th</sup> January 2008.
- Scheible, J., Tuulos, V. (2007) *Mobile Python. Rapid Prototyping of Applications on the Mobile Platform*. Wiley, Chichester, <http://mobilepythonbook.com/> – last visited 16<sup>th</sup> February 2008.
- Wahlig, A. (2007) Ped. A full-featured Python IDE for Nokia S60 platform. Version 2.xx – <http://homepage.rub.de/Arkadiusz.Wahlig/Ped/> – last visited 11<sup>th</sup> December 2007.
- Weiser, M. (1991) The Computer for the 21st Century}. *Scientific American* 265 (3), 94 – 104, reprinted in *Pervasive computing—January – March 2002* pp.18–25  
[http://www.media.mit.edu/resenv/classes/MAS961/readings/weiser\\_reprint.pdf](http://www.media.mit.edu/resenv/classes/MAS961/readings/weiser_reprint.pdf)  
<http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html> – last visited 26<sup>th</sup> December 2007.

---

<sup>2</sup> engl. translation: "When the students of the computer science course ... are standing at the busstop ..."

## Abbreviations

<b>IDE</b>	<b>I</b> ntegrated <b>D</b> evelopment <b>E</b> nvironment	<b>PyS60</b>	<b>P</b> ython for <b>S60</b>
<b>INFOS</b>	<b>I</b> nformatik und <b>S</b> chule	<b>PyUIQ</b>	<b>P</b> ython for <b>UIQ</b>
<b>OS</b>	<b>O</b> perating <b>S</b> ystem	<b>S60</b>	<b>S</b> ymbian <b>S</b> eries <b>60</b>
<b>Ped</b>	<b>P</b> ython <b>e</b> ditor	<b>UIQ</b>	<b>U</b> ser Interface <b>Q</b> artz

## Biographies



**Ludger Humbert** is teacher of informatics at the Willy-Brandt-Gesamtschule in Bergkamen (comprehensive school), teacher trainer in regions of Westphalia (Hamm and Arnsberg), and lecturer in didactics of informatics at the Bergische University of Wuppertal. He graduated as Dipl.-Inform. in Paderborn in 1976, became teacher in informatics and mathematics. He did some research on foundations of didactics of informatics at the technical university of Dortmund and got his PhD in 2003.



**Peter Micheuz** is since 1979 an Austrian teacher for Mathematics and Informatics at the Alpen-Adria-Gymnasium Völkermarkt and since 2000 in charge of teachers' education for informatics at the Alpen-Adria-University Klagenfurt. His doctoral thesis deals with a survey on informatics education in Austrian's secondary academic schools. As generalist and digital immigrant he is active and publishes in the domains of informatics education, standards and E-Learning.

## Copyright Statement

This work is licenced under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.