POLITECNICO DI TORINO
Repository ISTITUZIONALE

Training Engineers for the Ambient Intelligence Challenge

Original
Training Engineers for the Ambient Intelligence Challenge / Corno, F., DE RUSSIS, L.. - In: IEEE TRANSACTIONS ON EDUCATION. - ISSN 0018-9359. - STAMPA. - 60:1(2017), pp. 40-49. [10.1109/TE.2016.2608785]

Availability:
This version is available at: 11583/2648065 since: 2017-02-03T09:58:46Z

Publisher:
IEEE

Published
DOI:10.1109/TE.2016.2608785

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

12 September 2026

Training Engineers for the Ambient Intelligence Challenge

Fulvio Corno, *Member, IEEE,* and Luigi De Russis, *Member, IEEE*

***Abstract*—The increasing complexity of the new breed of distributed intelligent systems, such as the Internet of Things (IoT), which require a diversity of languages and protocols, can only be tamed with design and programming best practices. Interest is also growing for including the human factor, as advocated by the *Ambient Intelligence* (AmI) research field, whose focus is on transparently and intelligently supporting people. These new design methodologies are increasingly needed in the toolbox of new electronic and computer engineers, and teaching strategies should be devised that allow students to acquire a systems-level view instead of getting lost in technology-oriented approaches. This paper describes a study carried out over two academic years, in a course in ambient intelligence at Politecnico di Torino, Italy. In the course, a project-based learning approach was adopted, in which students design and prototype an AmI system, and their progress is closely monitored throughout the semester. The paper presents the learning goals and teaching strategies, analyzes the learning outcomes from the qualitative and quantitative points of view, and highlights the lessons learned in the process.**

***Index Terms*—Ambient intelligence, computer engineering, computing skills, multidisciplinary teams, Internet of Things, project-based learning**

I. Introduction

COMPUTER Engineering (CE) education is struggling to keep pace with the emergence of new technologies, devices, and execution environments. Educational institutions need to maintain core CE skills and competences, and at the same time introduce students to new topics such as mobile computing (including wearable computing) and the Internet of Things (devices, protocols, architectures, and applications).

The Internet of Things (IoT) paradigm is based on a network of devices interconnected with Internet infrastructure, over a variety of protocols, domains and applications. The co-existence of hardware, software, networks, and the mix of programming languages requires a systems-level view, difficult to reach in vertically-oriented courses.

Politecnico di Torino, one of the leading technical universities in Italy, takes a traditional educational approach to on disciplinary courses: each course focuses on one specific vertical technology. The systems-level approach, where students integrate different technologies, is not part of the normal course offering. This situation is similar in other engineering education institutions in Italy.

To provide its students with systems-level skills, in 2014 Politecnico di Torino initiated an elective course course, "Ambient Intelligence," taught in English and offered to all students enrolled in the third year of the Bachelor's degrees. To the authors' knowledge, it is the first university-level course in Italy about Ambient Intelligence. In this experimental and project-based course, a teamwork and design-driven paradigm is applied to teaching AmI system design; it aims to exploit core student skills acquired in students' previous engineering courses in multidisciplinary project work.

The main topic of the course is Ambient Intelligence (AmI), and in particular the design of AmI systems. This entails a strong focus on the application and on user needs. Students spend more than half of the course hours in the lab (and many more working on their own). Each student team must propose a specific project topic, and develop it according to a predefined set of milestones. Successful projects are presented to the general public (students, professors and entrepreneurs).

In the design, preparation, delivery, and evaluation of the course, the authors addressed the following Research Questions, applied to the AmI domain:

RQ1 *Can the complexity of an AmI system be tackled by thrid-year undergraduates with limited background knowledge?*
Since AmI systems integrate a wide variety of technologies (see Sections II and IV-A), students are required to reach a basic to intermediate skill level across a wide set of topics, instead of developing deep knowledge in one specific area, as is the case in most other courses;

RQ2 *Can students, design, prototype, and demonstrate a complete (if small) project within a single semester, and with their limited experience?*
Students are expected to focus on the completion of the whole project, instead of smaller and more focused assignments.

RQ3 *Is the collaboration of students from different backgrounds fruitful in AmI systems development?*
AmI systems are multidisciplinary, and the breadth of student degrees (Table II) and countries (Table III) could help teams to reach a more complete vision.

II. Ambient Intelligence: Short Overview

Ambient Intelligence is defined as "a digital environment that proactively, but sensibly, supports people in their daily lives" [1]. As a field, it is at the intersection of artificial intelligence, human-computer interaction, ubiquitous computing, and sensor networks.

Four cyclical steps comprise the behavior of an AmI system: *sensing* from the environment, from the users, and/or from the Internet; *reasoning* about sensed data; *acting* upon the

F. Corno and L. De Russis are with Politecnico di Torino, Italy.
E-mail: fulvio.corno@polito.it and luigi.derussis@polito.it

Manuscript received ...; revised .... The authors contributed equally to this work.

TABLE I
KEY FEATURES QUALIFYING AN AMI SYSTEM

Feature	Description
Sensitive	able to sense from the environment and its occupants able to process sensor data
Responsive	able to respond to user needs able to act on the environment
Adaptive	able to infer a situational context able to adapt to the context
Transparent	computing elements should disappear
Ubiquitous	computing elements are everywhere, distributed over the ambient and over different people
Intelligent	artificial intelligence is an enabler for context awareness, adaptivity, and proactivity

environment and/or towards the users; and *interacting* with the users, to keep them always in the loop.

Every AmI system should additionally possess a set of key features, Table I. A comprehensive definition of AmI may be found in [1], [2]; this does not prescribe any specific implementation technology. However, many required features may be realized by resorting to state-of-the art Internet-based distributed systems, by mixing mobile devices, web applications, cloud computing, wearable devices, and of course IoT devices.

III. LEARNING OBJECTIVES

Interdisciplinary work is increasingly needed in engineers' daily activities, and the Ambient Intelligence course is designed to help undergraduate-level students develop their first multidisciplinary professional skills.

The educational goal is to outline the multidisciplinary nature of AmI design and to provide a strong set of competencies. The course design, and the choices concerning contents and teaching methodologies, were informed by the state-of-the art (Section VII), in particular problem-based learning [3], and mainly sought to exploit the advantages of introducing real-world problems to the classroom [4], [5].

The learning goals are that students should be able to:

- Design and manage complex and distributed systems; AmI offers an intriguing, modern application domain.
- Work in multidisciplinary and international groups, and manage different backgrounds for the project's sake.
- Learn technologies on an as-needed basis: every project uses particular technologies, that are not part of the class material, but must be mastered by the team.
- Follow a structured design process, with well-defined (and strict) deadlines, deliverables and expected outcomes; in particular, to be able to separate product choices from technical ones, and avoid delaying all work until the last day.
- Identify and propose a project topic, of their own choice, within specified constraints; this may be the first time students need to define their own design goal, rather than working on pre-defined assignments.
- Take a system-design view instead of a technology-driven one; this requires focusing on user needs and the overall behavior, rather than on specific devices or technologies.
- Understand the importance of end-users in the conception of a new system and throughout the design process.
- Realize a working prototype and be able to demonstrate it: the end result should not simply please the teachers, or pass an assessment rubric, but should actually work in multiple occasions (in the lab, during the exam, at the showcase).

These objectives force undergraduates to adopt graduate-level methods of work, pushing their limits and increasing their self-confidence in the technical material they have learned.

The above learning objectives are common to project-based courses, especially on pervasive and ubiquitous computing. However, actually realizing the course in the context of the Italian engineering education system, and in particular at Politecnico di Torino (POLITO), presents a set of challenges:

- the course must fit within a rigid teaching structure in terms of available space, time (four time slots per week, of which two may be used for lab work), schedule (fixed time slots, set 14-weeks course duration), and effort;
- compared to other studies (Section VII), resources are very scarce: just three instructors, for a total of 60 teaching hours, to teach 70 enrolled students working in 15 or more different groups;
- in previous courses, POLITO students seldom had to produce working systems, so had to learn a result-oriented mindset;
- in their prior career, POLITO students were never expected to face multidisciplinary problems;
- students are not accustomed to following formal design processes, as usually they only need to deliver a final product, without intermediate checkpoints;
- teaching in Italian universities is usually in-class (lectures tend to cover all the needed material); in this course, students need to organize their own individual learning, identifying what they need to learn, and finding sources;
- since the course is open to all engineering students (not just electronics and computer science – a rare exception in Italian universities), some of them may lack the necessary background, and must plan to catch up;
- students meet each other for the first time in the course; their having no prior acquaintanceship slows the creation of working groups.

Overcoming these challenges requires pushing students to their limits, providing continuous support, and bending some of the institutional rules.

IV. COURSE OVERVIEW

A. Learning Framework

The general approach of the course is project-based learning (PBL), defined as "a learner-centered approach that empowers learners to conduct research, integrate theory and practice, and apply knowledge and skills to develop a viable solution to a defined problem" [3], built on the constructivist framework. The course employs an extended form of PBL, which combines project and team-based learning [6] with delivering fundamental knowledge via lectures and assignments, as suggested by [7], [8] and successfully experienced by [9]. Therefore,

TABLE II
ENROLLED STUDENTS' BACKGROUND

Academic Program	2014 (52 students)	2015 (70 students)
Computer Engineering	46% (24)	55% (38)
Electrical Engineering	33% (17)	29% (20)
Mechanical Engineering	11% (6)	8% (6)
Industrial Design	10% (5)	8% (6)

TABLE III
ENROLLED STUDENTS' NATIONALITY

Nationality	2014 (52 students)	2015 (70 students)
Foreign students	21% (11)	24% (17)
Italian students	79% (41)	76% (53)

the class emphasizes practical design activities and supervised word, taking 65% of course hours.

Three instructors are involved in the course (one professor and two post-docs from the Department of Control and Computer Engineering), and they alternate in classroom and laboratory sessions according to their specific skills. The course amounts to 60 hours of teaching (in class or in lab), nearly equally split among the three teachers.

The course spans 14 weeks in the second semester (March-June) of the third year of the Bachelor's degrees, with an average of three 90-minutes classes per week. Classes are videod; these videos are available to enrolled students.

The distribution of theory and practical lessons is carefully planned, with a higher density of the former in the first half of the course, when students need to gain knowledge about AmI, and a higher frequency of practical activities (lab exercises and supervised workgroup) in the second half of the course. The first weeks are also used to bootstrap teamwork activities (forming groups, discussing projects).

Course topics are diverse and aim at providing a wide view of AmI systems. view of AmI systems. Having been taught for two years, the course structure is quite settled[1], covering:

- Internet of Things and Ambient Intelligence: definitions and approaches (15% of class hours).
- Requirements and design methodology for AmI (5%).
- Design, analysis and requirements for an AmI system and its user interaction (20%).
- Practical programming of AmI systems: Python, Raspberry Pi, Web protocols and architectures (HTTP and REST), Android development, Web-based APIs, and collaboration tools such as Git and GitHub (60%). All technical notions in these topics are needed in project development. If additional notions are required by a subset of projects, additional hand-on sessions are organized.

Tables II and III report enrollment statistics for the 2014 and 2015 course offerings. Enrolled students come from different academic programs: Computer Engineering, Electrical Engineering, Mechanical Engineering, and Industrial Design, mainly. Internationality is considered important to initiate ideas and achieve above-average results; thus the course is taught in English and hosts a cohort (20-25%) of students from foreign universities.

On top of technical content, students are forced to learn and follow a rigorous design process, ensuring that all requirements and specifications have been expressed and validated, before designing the system architecture and starting to code. In choosing the most suitable design process for the course,

[1]Course material available at http://bit.ly/polito-ami, last visited Aug. 2016

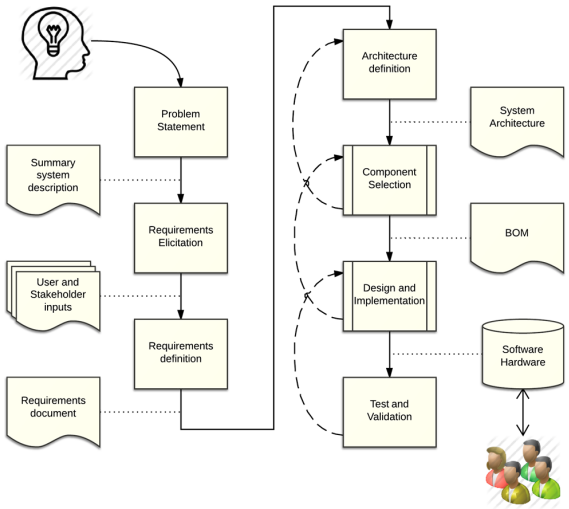

Fig. 1. The proposed design process for AmI systems

two key constraints must be taken into account: first, students limited prior knowledge of topics such as Python and Web development prevents an early start of prototyping and development activities; and second, the limited available time, that only allows time for one project iteration. Although interactive systems often adopt agile methodologies, due to the above constraints the chosen process has a more classical "waterfall" nature, Fig. 1. A major shortcoming of the planned process is the lack of adequate time for requirements elicitation and user involvement in the initial design concept phase: such activities, essential for a real product, are not included so as to allow time to complete the development of the working final prototype.

From the very start of the course, students form three- to four-people teams, and are guided to define the requirements for an AmI system, and then to design and implement it. This system and the "deliverables" produced throughout the semester are the focus of the course exam, which also includes a presentation of the team projects and an oral discussion on the same topic. Each team must find an idea suitable for the course and in line with AmI features.

Criteria for project selection are: Teams are encouraged to develop open and reusable solutions, which *must* involve sensing (environmental, user, social, cloud), actuation and interaction (on the environment, user, social, cloud), and intelligence (i.e., they should not be simply deterministic or preordained). Projects *cannot* be mobile-only, software-only, or hardware-only solutions, but *must* exploit different platforms and mix hardware with software and user interaction. All the projects *must* implement all the four main AmI steps (Section II), and *should* include most AmI features, Table I. Projects *must* be

focused on some "core" functionality, and *must* be relevant to the *theme of the year*.

Every year, a theme is chosen for the projects; in 2015 it was the Smart "Cittadella Politecnica" (the local POLITO campus), i.e., AmI systems on campus. Target environments could be classrooms, libraries, offices, hallways, open spaces, bars, laboratories, etc. Target users could be students, teachers, staff and/or visitors. The theme was sufficiently wide to generate some 20 projects, but sufficiently well-defined to determine whether a project fits.

After approval by the teachers, teams start developing their ideas according to the proposed methodology. Projects should be accomplished using the material learned in class, but should also motivate a search for additional ideas and tools.

The process has four main steps, each constituting a project milestone, and requiring the submission of a deliverable, on the public project website. The steps are:

1) Vision and goals definition. Deliverable D1 is the creation of the project website, with a one-page vision document outlining the goals of the proposed system and its benefits for the users.
2) Functional and non-functional requirements elicitation. Deliverable D2 is a list of such requirements.
3) System architecture design and component selection. Deliverable D3 consists of describing the overall system architecture, and the more detailed software, hardware and network sub-architectures.
4) Practical realization of the prototypical system. This is the longest phase, and Deliverable D4 corresponds to the material needed for the exam: a short video presenting the system, an oral presentation, and a live demo.

After each milestone, the teachers check all deliverables, and give detailed feedback to each group on their issues and shortcomings, during the following laboratory session. Students can evaluate their performance thanks to the feedback given by the teachers after each milestone. The importance of providing students with the means for self-evaluation has been discussed, for example, in [5].

Each group is required to publish and maintain the source code related to their project in a Git repository. Project repositories are part of a GitHub organization and must have a corresponding website created by using the "GitHub Pages" hosting. The presence of the source code and the frequency of commits are also evaluated during the exam.

Activities are not limited to "traditional" study and homework: students are forced to be proactive and teams are required to give presentations, to update the public website that presents their work, to prepare a pitch-like video for sharing the idea underlying their work, and so on.

In the laboratory, students have access to a wealth of different devices for their projects (covering IoT and smart home technologies). In the component selection phase (D3), they can select the technologies and components that best suit their project requirements (as expressed in D2).

B. The Exam

The exam consists of the presentation of the project. No additional written or oral examination is required. Ideally, the project would be finished right after the end of the course, so that foreign students can pass the exam before returning to their home university.

During the exam, two different aspects are subject to evaluation, with equal relevance:

1) the uploaded material, in terms of quality and appropriateness of the Deliverables (D1, D2, D3), of the web site, of the presentation video, and of source code commits;
2) the project presentation, in terms of quality and clarity of the talk (conducted with slides, usually), of the live demo of the project, and of the discussion (questions addressed to individual students) with the teachers that follows the talk and demo. The discussion is also used to evaluate the individual contribution of each student.

The final score is determined by adding the two evaluation scores and possibly applying small increases or penalties according to each student's individual contribution.

The 52 students in the 2014 offering of the AmI course formed 13 teams, while in 2015 the 70 students split into 17 groups. A short description of all projects is in Table IV.

In 2014, all but three groups passed with success the final examination, whereas in 2015, 14 out of 17 teams passed. The failed projects (three in 2014 and three in 2015) were mostly due to a break up of the group, where students could not find a way of collaborating due to other priorities or for personal reasons. In two cases, students were not motivated enough to continue the project, and abandoned the course. In one case, a four-student group broke up, but two of its members decided to take the exam, independently, with a subset of the work.

For the successful projects, students were generally able to demonstrate, during the discussion, their individual contribution to the project. In the 2015 offering, in four projects out of 14 the individual contributions were not considered uniform, and the individual students' scores were different from the general project score; two students got a better score than their peers, and four got a worse score. The reason for the reduced score was an insufficient contribution to the technical parts of the projects (i.e., these students only worked on the presentation materials).

These results give a positive answer to RQ1 and RQ2, showing that students can indeed realize a *complete* prototype of a *complex* system. RQ3, however, must be better analyzed, because of the problems in group collaboration.

C. Industry Involvement

Internet of Things technologies, and emerging Ambient Intelligence scenarios, are hot topics in today's computing industry, and links should be created between interested companies and student groups. The course teachers, over the last few years, created a network of local industrial contacts interested in the course and in the students' skills. This interest benefited the course in several ways: some companies donated devices and equipment for the laboratory; some contributed with seminars to the students; some listened to the students' pitches (after D3) and gave feedback, and some sponsored the Student Showcase, described next.

TABLE IV
FINAL AMI PROJECTS

Year	Project	Description
2014	DoorOnPhone	Home intercom on your smartphone
2014	Set App	Smart control of light settings around each workstation
2014	Smart Butler	Voice-controlled home automation
2014	Smart Gardener	Programmable self-adaptive home irrigation system
2014	Smart Notifications	Context aware notification delivery
2014	Smart Pet Feeder	Food delivery according to schedule and consumed quantity
2014	Smart Raise Your Hand	Call teacher and queue management for in-lab TA support
2014	Treasure Hunting	City-wide game based on user localization
2014	Smart Routine Optimizer	Activity detection and adaptation to user habits
2014	Never be locked outside your apartment again!	Detect missing objects (e.g., keys) when you leave
2015	EasyPark	Direct teachers to the closest free parking place
2015	ItsYourTurn	Sense whether a teacher is in their office and available for a meeting
2015	MarcoPoli	Sensors around the campus to identify critical areas
2015	MyBikePlace	Intelligent bike stand, with anti-theft capabilities
2015	NeverLate	Detect when a lecture starts and notify through smartwatch
2015	NoNoise	Measure ambient noise and provide feedback for quieter places
2015	Smart Make Your Bag	Help students to have all items they need for today's classes
2015	SmartClassSchedule	Intelligent displays outside classrooms act as information devices
2015	TrackDown	Detect if some item was stolen from your bag
2015	WC Info	Directs you to the closest toilet with the shortest user queue
2015	Well Cleaned	Monitor toilet supplies (paper, soap, ...) and call maintenance
2015	Adaptive Online Radio	Change ambient music according to tastes of the people present
2015	MyGuide	Vibrating cane and indoor navigation for blind people
2015	PoliRoute	Campus navigation through smart bracelets and dynamic billboards

D. Student Showcase

From the university point of view, a course is over once the exam is passed. However, students hard work and the quality of many of their projects, allow course results to be transformed into a public technology event. All students who passed the exam were invited (on a voluntary basis) to present their project to the public, during a dedicated open event, the "Ambient Intelligence Student Showcase", late in September (after the first three rounds of exams).

The event, hosted by the I3P incubator, is organized in a fair-like setting, with a stand for each project, and students freely describing their work to the showcase visitors. The students usually replicated the same demo that was given during the exam, but the best teams presented a much improved version.

Invitations to the showcase were disseminated to industries and startups, to Politecnico di Torino teachers, to all computer engineering students, and leaflets were posted on the official billboards. A joint press release (POLITO and I3P) was also issued to local and national newspapers.

The first presentation event, AmI Showcase 2014, took place on September 30, 2014, with eight former AmI teams presenting their projects. The showcase saw over 100 people attending the event. Participants were evenly distributed between students, POLITO faculty, and industries.

The 2015 edition of the showcase, held on September 29, was even more successful: of 206 pre-registered participants, 131 actually visited the showcase. The three most voted-for teams were awarded with nice electronic gadgets, one per team member, sponsored by the companies.The sponsor companies also interviewed the students, for possible recruiting, during a "private" showcase session, before the public one. In some cases, the contact has led to paid internships.

V. RESULTS

The learning objectives of the AmI course span several levels, and the learning outcomes may be evaluated in three different areas: *Domain knowledge* (ability to understand and design AmI systems), *Technical skills* (required languages, frameworks, protocols), and *Soft skills* (communication, collaboration and management capabilities).

In the following, quantitative and qualitative results for these aspects are presented; both outcome-oriented measures and user satisfaction results are included. In this evaluation, the focus will be on the 2015 edition of the course, for which appropriate measurements were recorded (for the 2014 edition, only student satisfaction data is available).

A. Student Satisfaction

In POLITO, at the end of each course, students are requested to complete an anonymous questionnaire, containing 18 questions on a four-point response scale, and a free area for comments and suggestions. Table V reports the percentage of students who were satisfied (scoring 3 or 4) across the most relevant questions. The response rate was 54.05% in 2014 and 56.36% in 2015. Most satisfaction indicators improved from 2014 to 2015, thanks to a better level of maturity of the course.

The main student concerns ($< 50\%$ satisfaction) were about the work load. This is somewhat expected, since the required activities are very demanding, especially when compared to the previous courses taken by the students, that were mainly *disciplinary*. This impression is confirmed by the free form comments, where students said that it was one of the most difficult courses, but also one of the most interesting of their career. Other students commented that they like these topics, to get more engaged, and increase their readiness for the market.

TABLE V
RESULTS OF ANONYMOUS QUESTIONNAIRES (% OF SATISFIED RESPONDENTS)

Measure	2014	2015
Exam organization and rules are clear	68%	88%
Course development is consistent with the syllabus	68%	81%
My prior knowledge was sufficient to take the course	47%	35%
The work load matches the given credits	37%	23%
Learning material is adequate	58%	69%
Laboratories are useful for learning	68%	73%
Teachers respect timing schedule	89%	96%
Teachers available for additional explanation	84%	88%
Teachers stimulate interest in the topic	79%	85%
Teachers are clear	79%	85%
I am interested in the topics of this course	95%	85%
I am satisfied by this course	53%	62%
Following the classes was useful	68%	54%

According to [10], ranking the course as challenging as well as highly satisfying indicates a successful course design.

B. Domain Knowledge

For domain knowledge, each project is scored against the six "AmI features," Table I, that represent desirable qualities of an AmI project. Each project is given a score (a subjective evaluation from the authors, from one to three stars) in each feature. The scores, Table VI, show that all projects have addressed the requirements for an AmI system, and tried to incorporate most features. In particular, sensitivity and transparency were mostly satisfied, while adaptivity and responsivity were harder to incorporate. The low score on the intelligence feature is due to students having no background in Artificial Intelligence.

C. Technical Skills

The evaluation of technical skills is more complex, since the skills are at the individual level, while the exam is at the group level. Different skills are needed to complete the project; all students must understand at least the basic level of each skill, but each skill can be mastered only by one or two students. In a multidisciplinary team, in fact, students naturally assign themselves the tasks they are more confident with.

In this evaluation, therefore, a skill is considered satisfactorily acquired if the group as a whole (i.e., at least some of its members) has been able to show mastery in that skill, within the context of the developed project. With this assumption, the success of the group is deemed more important than the abilities of its individual members. Actually, data reported in Section IV-B shows that the majority of students (92%: all but four out of 52, in 2015) were able to demonstrate a sufficient level of individual technical contribution.

To evaluate this outcome, the set of prior skills (before taking the course) is compared with the set of actually-used technologies. Prior skills are self-declared by students through an anonymous questionnaire that was delivered on-line before the beginning of the course. The set of actually-used technologies are determined by the analysis of project source code and demonstrations. Since questionnaires are anonymous, it is not possible to reconstruct the skills learned by each student, and the comparison is therefore made at the project level.

Table VII reports the quantitative results demonstrating the increase of technical skills. Skills are split into two groups: general and programming languages. For each skill, the prior knowledge declared by students is reported in the first column: for most topics, the overwhelming response is "Low," confirming that the topic was not known by the student. The only exceptions are programming in general and the C language, since nearly all students had taken a basic programming course. The last column reports the number of projects (out of the 14 successful projects in 2015) that actually used those skills or those programming language in their project. Most projects needed and successfully used most skills, showing they were learned during the semester.

D. Soft Skills

Some of the key technical skills students learn might be ineffectual unless accompanied by the professional soft skills necessary to bring them into play [11]. The course constraints prevented a proper inclusion of these skills in the course hours, therefore the students had to learn them on their own, through experimentation and through some information sources selected by the teachers.

To evaluate soft skills, a variety of factors were considered, Table VIII: multidisciplinarity, internationalization, team building, team management, presentation capabilities, and external interaction. The table reports how many of the 2015 projects met each of these soft skills. The most important of these relate to the successful operation of the team, and its interaction with teachers and external campus or industry experts; this aspect is often neglected in engineering courses, and proved to be a challenge for many of the students. The team approach has been claimed to benefit weaker students [10].

The evaluation of soft skills was not used in grading the projects, as it is outside the specific learning objectives, but is an important ingredient for students' professional growth.

E. Showcase Results

During the student showcase in 2015, a questionnaire was submitted to the attending students and professionals, with six simple questions, graded on a 5-point Likert scale. Out of the 131 visitors, 85 completed and returned the questionnaire (64.88% response rate), whose results are reported in Table IX.

The first two questions aimed at assessing the self-awareness of the visitors to IoT and AmI technologies, respectively. Professionals self-rated with high confidence (i.e., the target audience was competent on the topic), while students were obviously less confident. The next four questions aimed at evaluating the showcase impact, and the overall quality across all projects, according to their technical nature, their originality and degree of innovation, their presentation (demo and students descriptions), and shown posters. All quality facets were ranked quite highly, both by professionals and students.

TABLE VI
AMI FEATURES IN 2015 PROJECTS

Project	Sensitive	Responsive	Adaptive	Transparent	Ubiquitous	Intelligent
EasyPark	**	***	*	***	**	*
ItsYourTurn	***	**	**	***	*	**
MarcoPoli	***	**	***	***	***	**
MyBikePlace	**	*	*	**	**	*
NeverLate	**	**	**	***	***	**
NoNoise	**	*	*	***	**	*
Smart Make Your Bag	**	*	***	**	**	*
SmartClassSchedule	*	*	**	**	***	**
TrackDown	***	**	**	**	***	**
WC Info	**	*	*	***	*	*
Well Cleaned	***	**	**	***	*	**
Adaptive Online Radio	**	***	***	***	**	*
MyGuide	**	**	*	**	*	*
PoliRoute	*	***	***	**	***	**
Total stars	30	26	27	36	29	21

TABLE VII
LEARNED TECHNICAL SKILLS

Topic	Low (1-2)	Average (3)	High (4-5)	Projects
Programming (in general)	13.21%	**41.51%**	35.85%	14/14
Web Architectures	**58.49%**	16.98%	15.09%	13/14
Mobile development	**83.02%**	3.77%	3.77%	8/14
Source Control management	**86.79%**	1.89%	1.89%	14/14
Software requirements specification	**75.47%**	11.32%	3.77%	14/14
Python	**86.79%**	0.00%	3.77%	14/14
HTML/CSS	**67.92%**	13.21%	9.43%	14/14
JavaScript	**81.13%**	5.66%	3.77%	12/14
Java	**73.58%**	11.32%	5.66%	8/14
C	13.21%	18.87%	**58.49%**	3/14

TABLE VIII
APPLIED SOFT SKILLS

Measure	Definition	Projects
Multidisciplinarity	teams composed of students coming from different degrees	10/14
Internationalization	teams with students from different countries or speaking different languages	5/14
Team building	fairness of contribution by all members, which is an explicit criterion during the exam	12/14
Team management	groups not requiring team reshuffling and with no students leaving	12/14
Process	following a development process, respecting milestones, producing deliverables, handling feedback from milestones and deliverables	13/14
Presentation	quality of public deliverables on the project website, and quality of public presentations of the project to an open audience	13/14
External interaction	the amount of interaction that the project team had with campus experts (professors or technical managers) needed for project integration	8/14

TABLE IX
SHOWCASE QUESTIONNAIRE

	Professionals (35 total)			Students (40 total)		
Question	Low (1-2)	Average (3)	High (4-5)	Low (1-2)	Average (3)	High (4-5)
How much do you know IoT?	4	6	**25**	**18**	15	17
How much do you know AmI?	8	8	**19**	11	**18**	13
Projects: Technical quality	0	3	**52**	0	10	**40**
Projects: Originality/innovation	0	6	**29**	2	15	**33**
Projects: Presentation quality	0	2	**33**	5	9	**36**
Projects: Posters	1	8	**26**	6	13	**31**

VI. Discussion

Due to the ambitious goals set for this course, the teachers were prepared to face many challenges, and to have to make significant extra effort. Some unexpected issues were discovered during the course delivery that forced the instructors to consider new aspects or to adapt some activities. After the analysis of the course results, the findings related to the research questions can be discussed, and additional lessons learned can be shared with the research community.

A. Research Questions

The verified learned technical skills, Table VII, and soft skills, Table VIII, show that the response to RQ1 is positive. When properly motivated and supported, third-year undergraduate students can understand and design complex systems.

The success rate of the projects (24 teams out of 30) and the verified learned technical skills (Table VII) show that the response to RQ2 is also positive. The exam is bound to the completion of the project, and most groups could complete the prototype (by simplifying the initial design, if needed).

From the course experience and the student comments, the response to RQ3 is not entirely positive. While a definite advantage was observed from the collaboration of computer engineering (CE) students with electronic engineering (EE) ones (mixed groups realized better projects that CE-only or EE-only groups), the *other degrees* did not integrate so well. While engineering students (mechanical, electric, management) normally contributed to the projects to some extent, the industrial design and architecture students provided very limited contributions. In Ambient Intelligence systems, the design component, in particular when linked to the user experience, is very important, but those students were deterred by the complexity of programming skills and did not provide a useful contribution from their own specialty. This could be due to the limited time, or to insufficient guidance, and must be further investigated. On the other hand, no such effect was detected with respect to students' nationality.

B. Lessons Learned

In this section, the most relevant issues are presented and analyzed, and recommended corrections are proposed (under the "*Recommendation*" heading). In some cases, these solutions are already being applied in the 2016 offering of the course, underway at the time of writing, and are described under "*Actions*".

a) Failed projects: in the two offerings, six of the 30 projects (20%) failed the course, a typical failure rate at the university. Analysis of these six cases reveals that the causes for failure were always related to team-building issues: some students were not able to work constructively as a team, some were not committed enough to the course, or some did not contribute enough effort. On the other hand, when the students worked well together, the project result was always good to excellent. *Recommendation:* more effort should be devoted, at the beginning of the course, to forming groups, or to exclude from the course, as soon as possible, the students who are not interested or committed enough. *Actions:* students were asked, in the first two weeks, to declare their role in the group: this forced them to start thinking critically about the contribution they could bring to the project. Also, stricter rules for group composition were enforced (i.e., ensuring that sufficient programming skills were available). As a result, this year ten out of 81 students dropped the course before the first milestone.

b) Project ideas: students found the initial part of the course harder than expected, where they had to define the project idea. Much effort was required to nurture non-trivial project ideas (many proposals were dropped because they were too simple, i.e., just a mobile app, or a predefined threshold-based actuation), or ones that fell within the well-defined course constraints (many project were not AmI systems, since they lacked some of the key steps, or were not relevant to the course theme). *Recommendation:* instructors should plan to spend a lot of time in revising ideas (through after-the-class discussions, emails, and shared documents), and constructively help students reshape their projects. *Actions:* instructors met with each group for at least 20 minutes every two weeks.

c) Thinking at the systems level: under the influence of disciplinary teaching, students invariably tended to think about the technology (or the devices) instead of the overall system functionality. *Recommendation:* help students to overcome this tendency of not being able to see the wood for the trees, through discussions and tutoring. *Actions:* force the students, both in informal discussions and in documents, to refrain from using any technology-related term, and express their ideas with user-centric technology-neutral concepts, at least until the main AmI system functionality is defined.

d) Waterfall process: the design process presented in Section IV-A is the result of a set of constraints, and is not particularly suited to interactive systems such as AmI. This is indeed a weak point in the AmI course, that cannot be overcome unless more time is available or more prepared students enroll. *Recommendation:* a spiral-based iterative process should be adopted, to allow more than one design iteration. *Actions:* starting from year 2017, the course will be opened to M.S. students, as this will increase the input skills and should allow a faster pace of development for realizing at least two project iterations.

e) Trust and appreciation: the course is strongly based on periodic feedback, by the teachers, to student groups. In some cases (as emerged in the satisfaction questionnaires, in 2014), some student felt that teachers were being "unappreciative" of their work, due to the criticisms. *Recommendation:* instructors must find the right balance between highlighting design errors (in the process or in the technologies), while at the same time being "on the side" of students, and not "against" them. *Actions:* special care in written communications, as well as in discussion, was taken to speak in positive (constructive) terms, without watering down the message, but avoiding a "judgemental" attitude. This technique gave good results, as in 2015 no student complained about the issue.

f) Too early for hiring: some phases of the course (the pitch-like presentation, the industrial stakeholders, the public showcase) are explicitly designed to create a link between

students and work opportunities. However, it turned out that most students want to continue their study career with their Master's degree, and therefore are not interested nor ready to evaluate work offers. *Recommendation:* when external actors are involved, always help them to understand students' plans, to avoid excessive or inappropriate expectations. *Actions:* the university is trying to mitigate this problem, and increase the engagement with the industrial world, by offering the AmI course to Master's-level students, too, in 2017.

g) Bending the rules: needless to say, the dynamics required by the course, the teaching and mentoring effort, and the variability of project needs, made it very difficult to strictly follow all university regulations and constraints. A suitable degree of flexibility has been accepted (or rather, tolerated) by the university; the instructors are exploiting this forebearance, and are pushing it further every year. *Recommendation:* some colleagues will be enthusiastic about your work, some will be envious, most will try to dismiss you; the important thing is to focus on ones goals as an educator, that that prioritize helping students to learn. *Actions:* the Faculty Dean and the Department Director are always kept in the loop, invited to project presentations and showcase, and informed about course developments. This flow of information, bordering on stalking, helps to ensure that students continue to make significant learning progress from the course.

VII. Related Works

Other teaching experiences, similar to the Ambient Intelligence course at Politecnico di Torino, are being studied at various academic institutions. This section gives an overview of some notable or representative related experiences. The review is limited to courses whose information is publicly available, and balances European and U.S.-based experiences, and scholarly published papers and information provided by institutional websites. Reported experiences include Ambient Intelligence courses, as well as those in closely related fields, such as Ubiquitous Computing and Internet of Things.

One of the first systematic attempts at analyzing the education best practices and methodologies took place at the Ubicomp conference in 2003, within a dedicated Education Workshop, whose main results are presented [12] by the workshop organizers. A group of instructors in pervasive computing systems and ubiquitous computing shared their experience, and outlined similarities and differences in approaches taken in the various course offerings: general agreement was reached on the need of learning through experience, and in particular design projects. The AmI course, designed nearly ten years later, falls into this category.

Project-based learning was also described in [13], where undergraduate cadets at the U.S. Military Academy engage in a nine-month project development, to a client's requirements. The focus is on successful collaboration and learning between students of different disciplines (computer science and electronics), on learning the needed theories and technologies, and on delivering a complete product (not necessarily related to pervasive computing). The very demanding learning path is shown to increase student awareness of their capabilities, and to improve their employability. The AmI course develops in a narrower environment, where the available time is just 3.5 months and the goal is merely a working prototype, instead of a product; however, the issues of team building and multidisciplinary collaboration are very similar.

The exploration of the optimal mix between theory and experimentation is discussed in [14], where a course focusing on the applications of Multi-Agent Systems in smart environments is presented. The results highlight that, depending on the chosen projects, some theoretical classes might never be needed, and some useful notions might be missing. In the AmI course, all theoretical classes are useful for project development, thanks to a more constrained definition of the project topics and system architectures.

The Technische Universität Darmstadt, in Germany, offers an elective course "Ambient Intelligence" [15]. The aim of this course is to provide students an overview of a new vision for Human-Computer Interaction (HCI), in which people are surrounded by intelligent and intuitive interfaces embedded in everyday objects. In particular, the course addresses the emergence of Ambient Mobility and the ubiquitous and pervasive information access on mobile devices. Practical exercises account for half of the class hours and are organized like small "internships", with a bias towards the creation of "smarter" interaction with the environment. The AmI course, mainly due to timing constraints, does not include structured HCI topics, even if they would be important for the Ambient Intelligence vision.

The Mobile Computing and Internet of Things [16] course at the Karlsruher Institute für Technologie (KIT), Germany, is an elective undergraduate course for computer science students. The course aims at providing an introduction to methods and techniques of both mobile computing and the Internet of Things. The course exercises (about 25% of the class hours) deepen the knowledge acquired during the lectures through a practical project, which focuses on the creation of user interfaces. Compared to the AmI course, the class project play an auxiliary, rather than central, role.

The University of New Hampshire has offered the Ubiquitous Computing Fundamentals [17] course for electrical and computer engineering students since 2010. The course takes an interdisciplinary look at the foundations of ubiquitous computing (ubicomp), covering topics like system software for supporting ubicomp, human-computer interaction in ubicomp systems and sensors used in these systems. Students, in small groups or individually, realize a project, follow a requirements document (from topic identification to its completion) and report at the end of the semester through a written document and an oral presentation. Such a project accounts for 70% of the final grade. The general structure is very similar to the AmI course, even if ubicomp is a wider topic and there is more variability in the type of designed projects.

The Interactive Device Design [18] course of UC Berkeley, California, is more focused on "learning by making" and IoT. The course aims at giving students the concepts and skills required to design, prototype, and fabricate interactive devices, i.e., physical objects that intelligently respond to user input and enable new types of interactions. It is a hands-on course,

with enrollment limited to 30 students, and covers relevant techniques in 3D modeling and fabrication, electronics and circuit board design, sensing and actuation for interaction, wired and wireless communication with mobile devices and computers, among others. Half of the course is dedicated to a design project, carried by groups of students, in specific areas: health, energy technologies, connected communities, people and robots. Such a project counts for 40% of the final grade. The design and "making" of objects is tacked by a minority of AmI projects, and strongly depends on the students' skills, since these topics are not explained in the course, that aims more at a systems-level (rather than device-centric) view.

Several years of experience in a course on design of pervasive computing are reported by Martin et al. [19], where a rich team of five teachers, from different departments, coordinate students from electrical and computer engineering, industrial design and marketing in designing a pervasive computing product, stressing the contribution of each discipline. Besides the educational and technical aspects, the authors discuss the organizational aspects, and the integration of deeply interdisciplinary topics in the academic curricula, including the importance of faculty mind-sets and of institutional barriers. While the AmI course is developed with more limited resources, the reported issues of internal barriers are very similar, and "non-conventional" solutions are found in both cases.

VIII. Conclusions

The "Ambient Intelligence" course exploits a project-based learning approach to introduce students to the complexity of distributed interactive systems, where technical knowledge mixes up with user understanding and organizational and communication skills. The course has been offered for two years (2014 and 2015) and the 2016 edition is ongoing.

Learning outcomes have been quantitatively measured, and show that the adopted course strategy is able to deliver the needed hard and soft skills. At the same time, students are satisfied with the course organization, although they feel it to be very demanding. This paper posed research questions about project-based learning in AmI, and, after presenting the results of the course, discussed them. Finally, it analyzed in detail the lessons learned by the teachers, and discussed recommendations and corrective actions.

References

[1] D. J. Cook, J. C. Augusto, and V. R. Jakkula, "Ambient intelligence: Technologies, applications, and opportunities," *Pervasive and Mobile Computing*, vol. 5, no. 4, pp. 277–298, Aug. 2009.

[2] J. C. Augusto, V. Callaghan, D. Cook, A. Kameas, and I. Satoh, "Intelligent environments: a manifesto," *Human-centric Computing and Information Sciences*, vol. 3, no. 12, pp. 1–18, 2013.

[3] J. R. Savery, "Overview of problem-based learning: Definitions and distinctions," *Interdisciplinary Journal of Problem-Based Learning*, vol. 1, no. 1, pp. 9–20, 2006.

[4] C. Dean, T. Lynch, and R. Ramnath, "Student perspectives on learning through developing software for the real world," in *Frontiers in Education Conference (FIE)*, 2011, pp. T3F–1–T3F–6.

[5] C. P. Lee, A. S. Uluagac, K. D. Fairbanks, and J. A. Copeland, "The design of NetSecLab: A small competition-based network security lab," *Education, IEEE Transactions on*, vol. 54, no. 1, pp. 149–155, feb 2011.

[6] L. K. Michaelsen, A. B. Knight, and L. D. Fink, *Team-based Learning: A Transformative Use of Small Groups*. Greenwood Publishing Group, 2002.

[7] N. Hosseinzadeh and M. Hesamzadeh, "Application of project-based learning (pbl) to the teaching of electrical power systems engineering," *Education, IEEE Transactions on*, vol. 55, no. 4, pp. 495–501, Nov 2012.

[8] R. Rengel, M. Martin, and B. Vasallo, "Supervised coursework as a way of improving motivation in the learning of digital electronics," *Education, IEEE Trans.*, vol. 55, no. 4, pp. 525–528, Nov 2012.

[9] G. Goncalves, S. Nicola, L. Afonso, M. Ferreira, T. Ferro, and J. Mendonca, "Problem based learning: An experience in computer engineering," *Journal of Education and Human Development*, vol. 3, no. 4, pp. 271–281, 2014.

[10] T. Hassner and I. Bayaz, "Teaching computer vision: Bringing research benchmarks to the classroom," *Trans. Comput. Educ.*, vol. 14, no. 4, pp. 22:1–22:17, Feb. 2015.

[11] T. Hissey, "Education and careers 2000. Enhanced skills for engineers," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1367–1370, Aug 2000.

[12] G. D. Abowd, G. Borriello, and G. Kortuem, "Education training - report from the ubicomp education workshop," *IEEE Pervasive Computing*, vol. 3, no. 1, pp. 94–98, Jan 2004.

[13] G. A. Jacoby and J. R. Blair, "An undergraduate success story: A computer science and electrical engineering integrative experience," *IEEE Pervasive Computing*, vol. 5, no. 3, pp. 11–15, 2006.

[14] S. Bader, T. Kirste, and C. Burghardt, "Teaching smart environments and co-operative ensembles," in *Proc. of the 2nd Workshop on Pervasive Computing Education*, W. G. Griswold, Ed., 2009, pp. 17–25.

[15] Ambient intelligence. Technische Universität Darmstadt. Last visited Aug. 2016. [Online]. Available: http://www.gris.tu-darmstadt.de/teaching/courses/ws1415/ambient/index.en.htm

[16] Mobile computing and internet of things. Karlsruher Institute für Technologie. Last visited Aug. 2016. [Online]. Available: http://pcs.tm.kit.edu/530_534.php

[17] Ubiquitous computing. University of New Hampshire. Last visited Aug. 2016. [Online]. Available: http://hciunh.org/2015-ubicomp-course

[18] Interactive device design. UC Berkeley. Last visited Aug. 2016. [Online]. Available: https://bcourses.berkeley.edu/courses/1376830

[19] T. Martin, E. Coupey, L. McNair, E. Dorsa, J. Forsyth, S. Kim, and R. Kemnitzer, "An interdisciplinary design course for pervasive computing," *IEEE Pervasive Computing*, vol. 11, no. 1, pp. 80–83, January 2012.

Fulvio Corno has been an Associate Professor at the Department of Control and Computer Engineering of Politecnico di Torino since 2002. He is also Associate Editor of the IT Professional IEEE Magazine and of the Journal of Reliable Intelligent Environments.

Luigi De Russis is a postdoc research assistant at the Department of Control and Computer Engineering of Politecnico di Torino. He received his Ph.D. in computer engineering from Politecnico di Torino, in 2014. He is an ACM member.