Designing intelligent applications

A Creative approach for the integration of machine learning algorithms in interaction design education

Jean-François Omhover*, Carole Bouchard**, Vincent Meyrueis***

** Arts et Métiers ParisTech, LCPI, jean-francois.omhover@ensam.eu*
** Arts et Métiers ParisTech, LCPI, carole.bouchard@ensam.eu*
** Arts et Métiers ParisTech, LCPI, vincent.meyrueis@ensam.eu*

Abstract: Confronted to the integration of intelligent technologies into modern products / interactive systems, interaction designers need a way to understand and manage the integration of machine learning (ML) technologies into their projects. In order to teach the issues and practices related to intelligent technologies for interaction designers, we have designed a very short program to enlighten interaction design students to these technologies. The class has been designed as a workshop for the students to integrate these technologies into a creative concept. A formalism and a toolkit has been proposed for the animation of the workshop. In this paper we present the requirements and the framework proposed, along as a short case study of its application in our interaction design course.

Key words: interaction design, computer science education, machine learning

1. Introduction

Current products tend to integrate intelligent technologies more and more. In particular, these technologies enable the product to be self-adaptative to the behavior of its users, or to their preferences. It can be, for instance, an electronic companion that reacts to the gestures or the emotions of the user. Or it can be a user-friendly energy optimization platform for houses. Also, the perspective of the internet of things and the integration of big data let us envision the creation of high-value services/products based on data mining algorithms. We have to educate a new generation of interaction designers able to transform intelligent computing technologies into products. They will have to rely on a whole set of knowledge and issues coming from the computer science field such as machine learning (ML) or computational intelligence [4].

For this reason we have integrated a ML class in an interaction design course. This class has been designed based on a creative approach to computing education [15] and a framework derived from Computational Thinking [18].

After a short theoretical class, our experimentation consisted in the integration of ML 'cards', a map of 'intelligent verbs' representing intelligent functions or needs, and a specific formalism to specify computing solutions based on IDEF0. We have integrated the cards, the map and the formalism into a 6h creativity session (2 x 3h) with two groups of 5 and 6 students (designers and engineers in interaction design, none being computer scientists).

First, they would explore the functionalities based on the intelligent verbs map. Then they would specify the solution based on our formalism to embed the intelligent functions into the product concept. Finally they would use the ML cards to choose among the intelligent technologies. As a conclusion, students have been able to discriminate between families of ML algorithms and to apply them correctly into new intelligent services/products.

2. Elaboration of a Machine Learning class within an Interaction Design course

In this section, we describe the context of our Interaction Design course, in which we will integrate a ML class. Then we will look for educational models drawing on literature from Computer Science Education and Design Education. Finally, we will derive a framework and a methodology for the elaboration of our ML class, that we will test in the next section.

2.1. Context and Requirements

In 2011, our laboratory has opened an Interaction Design course as a Master of Research. This course is a one year program divided in three periods : theoretical classes (4 months), a short application project (1 month), a research project (6 months). In the first period – theoretical classes – the students are exposed to many different topics related to interaction design, and divided into modules :

- fundamentals of interaction design,
- interaction technologies,
- interaction, cognition and emotion,
- social and economical aspects of digital technologies
- imaginary aspects of interaction.

Into the interaction technologies module, the coordinators of the course wanted to integrate a ML class in order to enlighten the students to the family of technologies supporting the intelligent behavior of products or interactive software. Twelve hours were dedicated to this sole topic.

The issue regarding this particular class if that the students involved in this interaction design course do not have any computer science background. Aged between 22 and 27, they come from two main fields : product design, and engineering. Our experimentation took place in the period 2012-2013, there were 11 students involved (5 designers, 6 engineers).

Considering the background and skills of the students, and the time constraints, it would have been too ambitious to deploy a full scale class on ML. It would have been not only impossible but also not necessary nor expected by the students : the point of this class is not to educate ML experts per se, but to give the students the ability to investigate ML technologies, to understand the underlying technological issues and practices, and to integrate these technologies into their product design projects.

In this context, we established the following problematic for the elaboration of this ML class : *how to teach and transfer a basic understanding of the ML technologies for interaction design students ?*

2.2 Computer Science Education models

Computer Science Education deals with the issues, practices, methodologies, that are related with the instruction of computer science for young students (K12) to higher education. It received a new gain of attention recently through the expression “computational thinking” [18] : it was identified as a key concept to decode the skills engaged into computer science activities and to promote new methods for CS education [11, 6]. But the concept is not new, and it has already been explored in other disciplines as a way to understand how expert programmers deal with representations [8], resolution strategies [17, 16], structuration of problems [13,7], or cognitive aspects of programming [5, 14, 3].

In our case, the point is not to educate our students to computer science itself but only to one part of it that are machine learning algorithms. But in the process of integrating these technologies into interactive products/systems, our students will necessarily have to face some of the issues outlined in the CS education literature. More specifically, some of the skills mentioned in the literature above have been a topic of attention in our case :

- structuration of the problem [13,7] : students need to acquire the ability to structure a problem in computational terms, or more generically in terms of concerns separated within the system by functions, or components.
- formalization [8] : in order to be able to see clearly the nature of the technology required by their system, they will need to formalize its inner structure, to see the dependencies between components and relate inputs and outputs to functions.
- pet concepts [1] : for the approach to be effective, our students will need to manipulate the technology in abstraction and to have an isolated concept on which they will be able to focus and play with.
- identification [12, 1]: they will have to be able to identify the nature the computing required, and to match these requirements with technologies taken from the ML field.

In synthesis, in order to solve our problematic, we have decided to implement a creative approach in which students will engage theoretical knowledge on the ML field by integrating these technologies into pet concepts of their own. During this approach, they will have to identify the technologies corresponding to their needs by isolating the functions, specifying the perimeter of these functions and the technological building blocks, in order to relate these functional requirements to existing ML technologies. This approach also draws on the projects we have deployed in previous research, in which the complementarity between ML technologies and user needs was found by relating requirements, functional (and cognitive) needs, and technological specifications [2,10].

3. Proposal for a creative ML class

In this section, we present the detail of the ML class program by first describing the structure of the 12 hours. Then we present more specifically the creative workshop organized with the students in order to make them reflect upon the inner mechanisms of an intelligent product / system. We also describe the kind of material used in the creative workshop.

3.1. Global structure of the class

We have segmented the 12 hours in three part described in the following.

Introduction to machine learning (3 hours) – An introductory class provided a global overview of the machine learning technologies. It first briefly presented the historical context in which these technologies were developed, then quickly entered into a description of the families of technologies available. Our point of view combined two aspects : first the issues of the deployment of these technologies in terms of building process (preparation of data, learning, and use), secondly a simplified typology of algorithms in terms of supervised learning, unsupervised learning, or adaptive agents. The technologies presented were :

- supervised learning : artificial neural networks and decision trees
- unsupervised learning : principal component analysis, self organizing maps, clustering, association rules
- generic adaptation frameworks : genetic algorithms, multi-agents

For the purpose of simplicity, we decided not to extend the number of algorithms presented beyond a limited number. Furthermore, some families were presented in abstract terms and other in detail: for instance we presented “clustering” and “genetic algorithms” as a global framework because of its adaptability to many fields of study, and we presented “artificial neural networks” and “decision trees” more precisely because it enables us to present issues relative to learning capability (separation of classes, decision boundaries) and readability of knowledge.
A second part of this introduction consisted in the examination of intelligent products (figure 1) and a discussion on the identification of the underlying technologies considering the functions provided by these products. This would be a first step for identifying the functional purpose of these technologies within existing products.

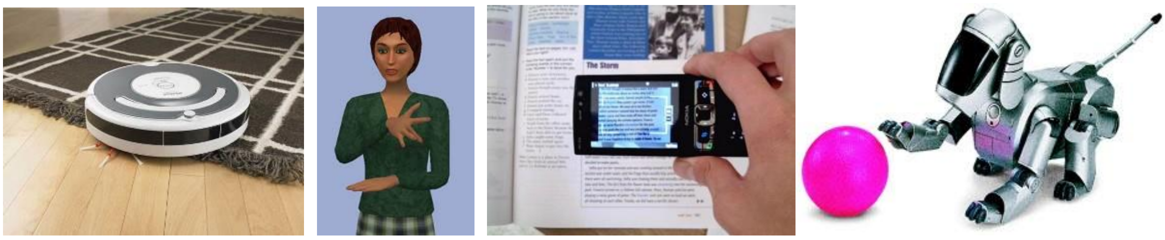

Figure.1 Some of the examples of “intelligent” products/systems used as illustration in class,
from left to right : the Roomba vacuum cleaner, an IBM avatar translating speech into sign language,
the Nokia OCR Scanner, the Sony AIBO

Creative workshop (6 hours) – The main part of the class consisted in a creative workshop. This workshop was organized in two 3 hours sessions (one on the morning, one in the afternoon) engaging the students into the creative design of an intelligent system of their choice.

Machine learning in design research (3 hours) – In order to extend the purpose of the class to research projects, a last theoretical class was inserted to present research projects, activities, or academic systems benefiting from machine learning technologies to extract knowledge in the design process (for needs analysis to mechanical engineering).

3.2. Structure of the creative workshop

This workshop was structured as shown on Table 1.

Table 1. Work plan of the creative workshop

N°	Phase	Time	Given material	Objective
	Preparation	0h20	None	Presentation of the objectives of the day, division into subgroups.
1	Definition of an objective	0h40	Paperboards and post-its	Each group would brainstorm on an intelligent system they would like to work on in the following steps. Outcome : a 5 mins presentation of the objective
2	Function definition	1h	Mindmap of “intelligent verbs”	The group was asked to reflect upon the intelligent functionalities provided by the system. They were provided paperboards and post-it to mark down ideas in a brainstorming they would organize. Outcome : a cartography of functions to debate with the instructor
3	Global structuration of the concept	1h	Structuration framework	The group was asked to describe the expected system into a given structure composed of 5 points: environment, sensors/data, system perception, system memory, decision

				Outcome : one paperboard summarizing the concept by the 5 points
	Lunch break			
4	Formalization	1h15	Intelligent system formalism	The group was asked to formalize its concept using a given formalism to express functional blocks of intelligent functions, their use of specific inputs, and their production of outputs. Outcome : a formalized description of the system
5	Identification of ML algorithms	1h15	ML cards	Using the formalization of the system, the group was asked to identify what kind of algorithms (type of learning, then family of algorithm) that would be used to support each functional block. Outcome : a proposition of ML technologies to support the design of the system.
6	Collective presentation / debriefing	0h30	None	A collective presentation of the result of the workshop and an open discussion between participants and instructor.

On the morning, the students were asked to define an objective of their own as an intelligent system they would like to design (phase 1). The “intelligence” of this system was to be reflected upon by the students and would be validated by the instructor before the next steps. Then they would reflect upon the functionalities they would like to propose to the user (phase 2). In order to help them to orient their reflection toward intelligent functionalities, a prepared mindmap of “intelligent verbs” was introduced (see section 3.2). Picking some of these verbs as inspirational material would help them characterize the “intelligence” of their system. Finally, they would start to separate the different aspects of their concept (phase 3) based on a structuration framework provided by the instructor (see section 3.2).

On the afternoon, each group would start to think more precisely about the technical aspects of their concept. They would first formalize their concept technically (phase 4) by specifying inputs, outputs, and the internal mechanisms related to their system’s learning capabilities. Finally, using this formalization, they would propose algorithms matching their system’s learning capabilities by using ML cards capturing basic main families of ML technologies (as presented in the introductory class).

3.3. Material introduced in each phase of the workshop

We describe here each material given to the students in each phase of the workshop. The reader of this paper has to know that these material were originally written in French, for readability reason they have been translated in English on the following figures.

Mindmap of “intelligent verbs” - The students were provided a mindmap prepared by the instructor to brainstorm about functionalities of their intelligent system. This mindmap was intended to capture many different “verbs” users attach to an intelligent system, and was designed from the instructor’s knowledge of such systems. The mindmap is presented figure 2 and is globally segmented into three main categories : interaction between the user and the system, interaction between the system and its environment, and finally relation between the user and data. Each time, the verb is to be understood as in the sentence “the system’s purpose is [verb]”. These verbs should not be understood as an exhaustive list but more as an inspirational material for the workshop.

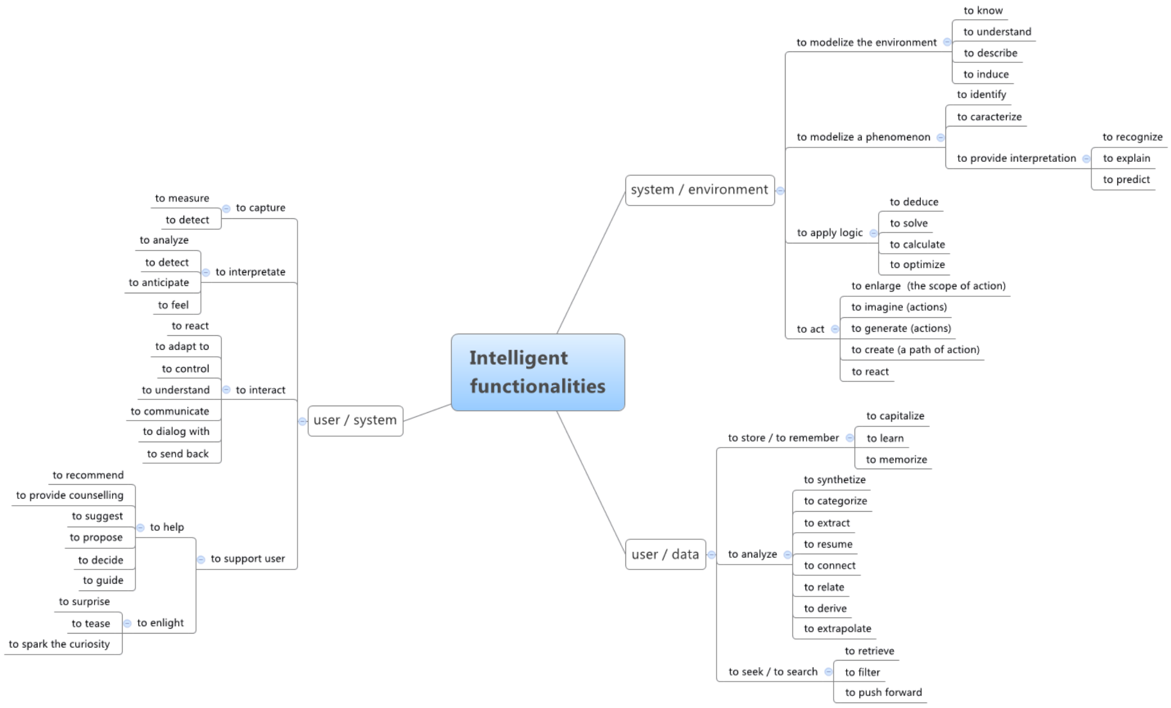

Figure.2 Mind map of "intelligent verbs" provided to the students

Structuration framework – The main difficulty we envisioned for students was to be able to separate the different concerns of their system's design. In order to tackle with this issue we have prepared a framework of questions the students would have to fill up in the phase 3 of the workshop. These were simple questions they would have to answer for the synthesis of the concept of their system :

- environment : what are the elements, contexts, users located in the environment of the system ?
- sensors / detection : what are the data, sensors, capture processes the system has to use / can benefit from (in relation with the environment) ?
- system perception : what does the system need to "perceive" ? or, how would you characterize the "perception" of the system ?
- memory : what does the system need to memorize ? (data, knowledge, models…)
- decision : what kind of decisions the system will be able to do ? or what can of actions he will be able to decide to do ?

Intelligent system formalism – In order to support the formalization and the specification of their system, the students were proposed a simplified formalism based on inputs, outputs and functional blocks (very basic version of IDEF0, see figure 3). They would specify their system considering two phases : first the learning phase in which the system would learn something from input data or sensors, second the application of learning in which the models or knowledge learned by the system would be applied to inputs for expected output or actions (actuators). There could be multiple components, or learning functional blocks that could be interconnected, in this case they would be asked to show the connection between these functional blocks. For instance, a system that would learn to recognize user emotions – that would be a first functional block - could use this recognition as an input for the adaptation and the automatic generation of good responses to these emotions – a second functional block. It would be important to specify the difference between the two blocks as the ML technologies within these blocks would be different.

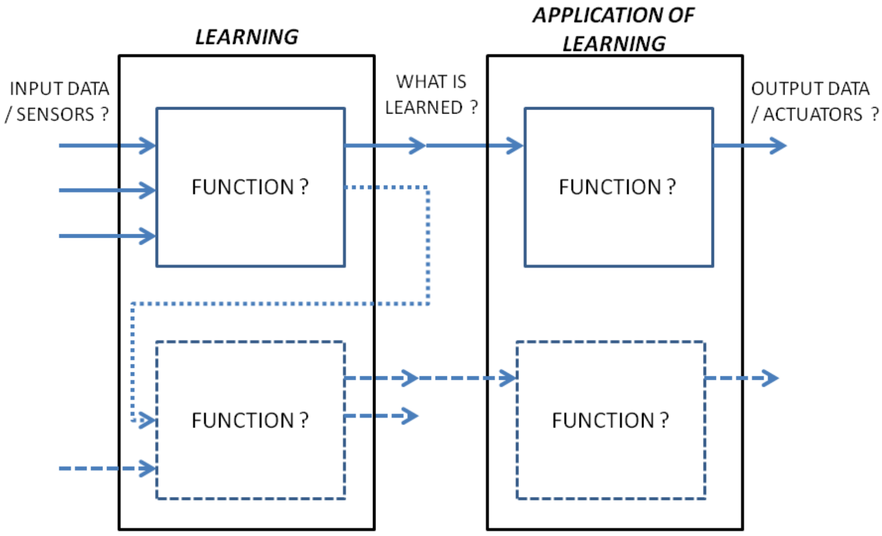

Figure.3 Generic formalism proposed to the students

Machine Learning cards – a compiled list of algorithms or families of algorithms was compiled as ML cards in which each algorithm/family was described in functional terms : the objective of the algorithm, the process associated to the algorithm, and its expected result. The same cards were used in the introductory course (see section 3.1) and in the workshop. They would be used for identifying the underlying technology needed to design the system.

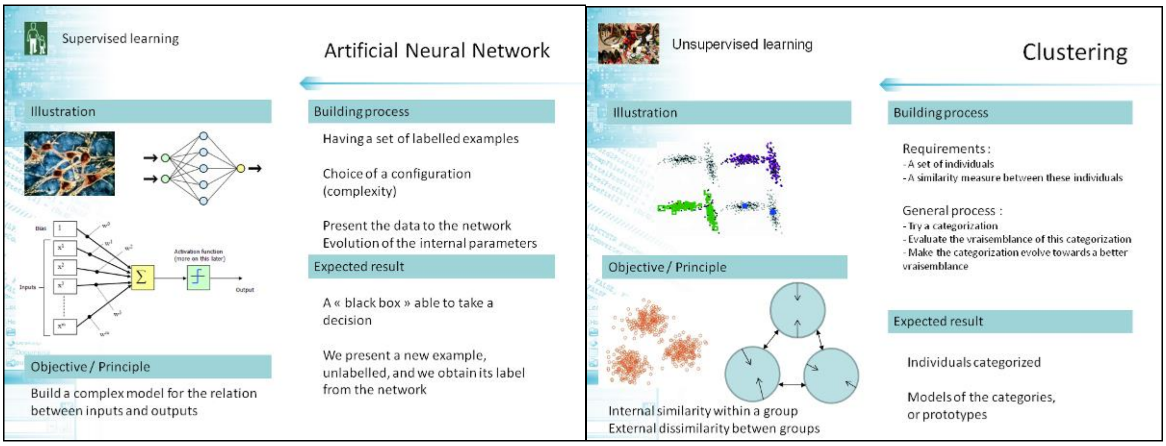

Figure.4 Two of the ML cards from the introductory class

3.4. Roles of the instructor

The instructor has a very important role to play during the workshop. We expect him to have roles similar to the four roles described by [10]. First, he has to take the role of a *team manager*, assessing the risks and feasibility of the system. This is particularly true in the phases 1 and 2 where the objective of the system is being defined. The instructor would use his knowledge of the field in order to limit the scope of the system or to provide a better definition of the objective : the point here is to be able to define some system that would be challenging enough (for the exercise to be non-trivial) and simple enough (for feasibility). Second, he has also to take the role of a possible *customer/user* of the system in order to understand where the concept is going and how it would be applicable to a given user need. Third, he has to be an *architect* of the solution. Part of this role is already embedded into the formalization and structuration material provided to the students, but it has also to be maintained through the session by iteratively assessing the structure of the solution and providing a proper separation of concerns when needed. Finally, he has to take the position of a *mentor*, more particularly engaging his knowledge of the ML field in order to help the students to identify the possible algorithms needed for the design of a system (in phase 4 and 5), to clear some misunderstanding about the ML technology identified by the students or to assess the proper use of a given algorithm in the context of the students' concept.

4. Application and results

In this section, we provide an overview of the results obtained by the participants to the workshop. We also examine how the provided material was used by the participants and underline some of the limits observed, that could lead to potential refinements of the method.

4.1. Results of the workshop

As mentioned earlier, the workshop took place this year with 11 students (5 designers, 6 engineers). Two grups were formed at the beginning of the workshop and the two groups were composed of mixed profiles (3 designers and 3 engineers in one group, 2 designers and 3 engineers in the second group). At the beginning of the workshop, each group chose a subject for the day. Group 1 chose to work on a personal assistant that would help me to choose my activities and organize my day (it was called the "27h a day assistant"). Group 2 chose to work on a reading assistant that would make reading book more interactive and intelligent (called "intelligent book").

Through the phases of the workshop, each group managed to obtained a precise definition of the system, along with a good understanding of the inputs (ex: light sensors, time tables, personal preferences, etc) and outputs (warnings, adapting the interface of the system, proposing books, etc). The second part of the workshop was more difficult, and required a constant presence from the instructor that was to answer many questions, first on the architecture of the functional blocks, then on the choice of technologies. Many questions were related to the understanding of the ML cards : how one technology worked, if it would be applicable to the system designed, etc. It showed a good engagement of the students and a need to understand the differences and issues of each technology.

Through the workshop, it was necessary to relax what would be perceived as "constraints" by the participants. More specifically the formalism was first perceived as a difficult part, and the participants were first trying to match the formalism exactly. It was then told that an exact formalization was not mandatory but needed only to be able to communicate about the functioning of the system to the instructor. We decided to relax the consistency of the formalization in order to gain more understanding of the purpose of the formalization (separation of concerns and global architecture).

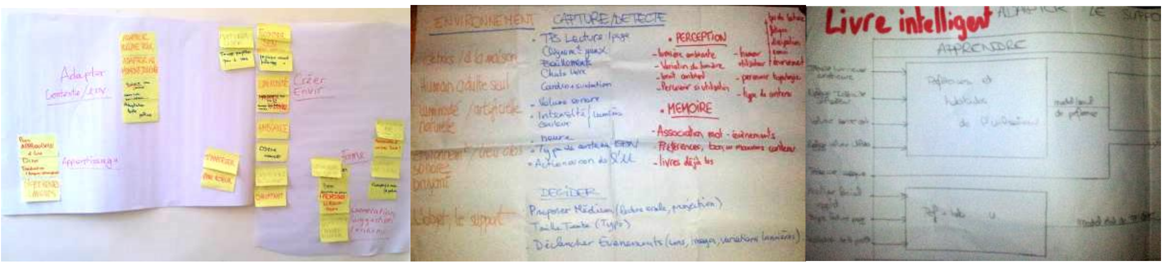

Figure.5 Illustration of the results obtained by group 2 – "intelligent book"

In the debriefing that followed, the students first expressed the satisfaction to have an application class instead of a theoretical class on this topic. It was perceived by them as more enjoyable and fun. The formalization proposed was perceived as difficult but also structuring. At the end of the debriefing, several students asked for a possible development of their system, and asked for tutorials and complementary materials on the ML algorithms involved in their system. It would be seen as an appetite for these technologies, and a positive impact of the workshop.

4.3. Limits of the approach and future refinements

With the feedback of the students, and the formalization of our methodology, it shows that some aspects were overlooked. It would have been more proper to install a measurement protocol through the workshop. But considering the experimental nature of the class it was considered as a possible hindrance for the performance of the students. This first session helped us to envision possible difficulties and issues relative to each phase of the workshop, it will lead to the formalization of a questionnaire that will let us measure a posteriori the difficulties and competencies gained through the session. Of course, it also has to be repeated next year to get a broader assessment of the performance of the toolkit, and maybe extended to other courses in other contexts.

5. Conclusions

This paper summarizes a first experiment in the integration of a creative ML class into an interaction design course for the creation of intelligent products / systems. This creative class has been designed by taking into account the background of the students, and issues coming from the literature in computer science education. We have formalized a toolkit for the animation of this session based on the characterization of "intelligent verbs", on the formalization of the solution, and the identification of ML technologies via ML cards. The results are promising and show a good engagement from the students and an overall understanding of the difficulties encountered by the students in mastering these technologies. These results need to be developed in the next session and confirmed by a specific protocol based on this session's feedback.

6. References

[1] Baker, A. & Van der Hoek, A. (2010) *Ideas, subjects, and cycles as lenses for understanding the software design process*, Design Studies, 31(6), pp.590–613.

[2] Bouchard, C., Mougenot C., Omhover, J.F., Aoussat, A. (2007) *TRENDS : A Kansei based information retrieval system based on the Conjoint Trends Analysis method*. In Proceedings of IASDR 2007. Design Research Society, ed.

[3] Eckerdal, A., Thuné, M. & Berglund, A. (2005) *What does it take to learn "programming thinking?* In Proceedings of the 2005 International workshop on Computing Education Research - ICER'05, ACM Press, pp. 135–142.

[4] Essinger, S.D. and Rosen, G.L. (2011) *An introduction to machine learning for students in secondary education*, In Proceedings of Digital Signal Processing and Signal Processing Education Meeting 2011. IEEE, pp. 243.248.

[5] Guindon, R., (1990) *Knowledge exploited by experts during software system design*, International Journal of Man-Machine Studies, 33(3), pp.279–304.

[6] Hambrusch, S. et al. (2009) *A multidisciplinary approach towards computational thinking for science majors*, ACM SIGCSE Bulletin, 41(1), pp.183–187.

[7] Ikeya, N., Luck, R. & Randall, D. (2012) *Recovering the emergent logic in a software design exercise*, Design Studies, pp.1–19.

[8] Jackson, M. (2010) *Representing structure in a software system design*, Design Studies, 31(6), pp.545–566.

[9] Kim, J.E., Bouchard C., Omhover J.F., Aoussat, A. (2010) *Towards a model of how designers mentally categorize design information*, CIRP Journal of Manufacturing Science and Technology, 3(3), pp.226–218.

[10] Kurkovsky, S. (2008) *Four roles of instructor in software engineering projects*, ACM SIGCSE Bulletin, 40(3), p.354.

[11] Lee, I. et al., (2011) *Computational thinking for youth in practice*, ACM Inroads, 2(1), pp.32–37.

[12] Muller, O., Ginat, D. & Haberman, B. (2007) *Pattern-oriented instruction and its influence on problem decomposition and solution construction*, ACM SIGCSE Bulletin, 39(3), p.155.

[13] Parnin, C. & Rugaber, S. (2012) *Programmer information needs after memory failure*, In IEEE, ed. 20th IEEE International Conference on Program Comprehension (ICPC). IEEE, pp. 123–132.

[14] Parnas, D. (1972) *On the criteria to be used in decomposing systems into modules*, Communications of the ACM, 15(12), pp.1053–1058.

[15] Rothermich, J. (2009) *A perspective on computational intelligence education*, IEEE Computational Intelligence Magazine, 4(2), pp.50.51.

[16] Tang, A. & Van Vliet, H. (2012) *Design Strategy and Software Design Effectiveness*, IEEE Software, 29(1), pp.51–55

[17] Visser, W. & Hoc, J.-M. (1990) *Expert Software Design Strategies*, In Psychology of Programming. pp. 239–274.

[18] Wing, J.M. (2006) *Computational thinking*, Communications of the ACM, 49(3), pp.33.34.