

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# AUTOEVALUATION IN 8086 ASSEMBLY LANGUAGE PROGRAMMING

**Marie-Martine Paget<sup>a</sup>** (Marie-Martine.Paget@lip6.fr)  
**Marie-Claire Nogier<sup>b</sup>** (Marie-Claire.Nogier@ufr-info-p6.jussieu.fr)  
**Hélène Giroire<sup>a</sup>** (Helene.Giroire@lip6.fr)

<sup>a</sup>LIP6, <sup>b</sup>UFR d'informatique - University of Paris 6  
4, place Jussieu, F-75252, PARIS cedex 05 - FRANCE

**KEYWORDS:** Pedagogical resources, autoevaluation, 8086 assembly language programming, computer aided teaching.

## Abstract

*Autoevaluation (self assessment) is more and more developed for teaching at university. We present our contribution to this pedagogical resource. Our exercises are intended for students in second year at the University of Paris 6 for the module "Computer Architecture". These numerous exercises complete the classical courses and allow students to check their know-how. We have experimented with success our set of exercises for the past 2 years with numerous students in UTEs<sup>1</sup>. Uteval software has both a student interface (<http://upipc100.cicrp.jussieu.fr/autoeval/>) and a teacher interface. The analysis of evaluation questionnaires proposed to the students and the discussion we had with them show their interest for this type of activity.*

## INTRODUCTION

Autoevaluation is a subject which interests many researchers in various countries. A quick search on the web is enough to convince oneself of that. A great number of autoevaluation exercises go with courses in very different fields in many universities. Most of them are on line on educational sites. The RUCA<sup>2</sup> offers exercises in most of the scientific fields. In the scope of pedagogical resource production, the Joseph Fourier University in Grenoble with its ARCADE CLIPS-IMAG team has been very active in this field. Since the nineties, that team defined a concept of autoevaluation (David 1993) and developed means for autonomous learning within the framework of the CAFIM<sup>3</sup> (David and Fournier 1995). The team designed and built specialised authoring tools in order to construct exercises with an automatic generator GenEval (David and al. 1996), (David and Peyrin 2001). These realisations are its contribution to the European ARIADNE1<sup>4</sup> project.

It can be noticed that autoformation and autoevaluation are present in the same references above. We would like to distinguish between these two notions. Autoformation allows learning a discipline. Autoevaluation is a means for the learner to control his ability. In autoformation there is a progression in the sequence of the notions adapted to the discipline. In autoevaluation, the knowledge is supposed to have been acquired, hence the student controls his know-how. This is the reason why the plan of our autoevaluation exercises does not exactly follow that of the course. The student may think that he has acquired the knowledge, but his understanding can be

---

<sup>1</sup> Usage des Technologies Educatives en Sciences: <http://www.lutes.cicrp.jussieu.fr/lutes>

<sup>2</sup> Réseau Universitaire des Centres d'Autoformation: <http://www-ruca.univ-lille1.fr/>

<sup>3</sup> Centre d'Autoformation et Innovations Multimédia: <http://www.ujf-grenoble.fr/CAFIM/>

<sup>4</sup> Alliance of Remote Instructional Authoring & Distribution Networks for Europe: <http://www.ariadne-eu.org/>

superficial. The autoevaluation allows him to check what he really knows and to appropriate this knowledge. He is an actor in his own training.

In our student interface, an exercise appears on the screen. The student is invited to solve the exercise on a paper in a limited time, exactly as he does in an examination situation, but without stress. When he has finished, he makes the solution appear on the screen, in order to mark his work and compare his mark to a critical threshold. If the learner plays the game, that is does not check the solution immediately, there will be a go-back allowing him to control his personal progress in his know-how. Our students tell us that by training like this, they have understood notions that they had not assimilated during the lectures.

## **SITUATION**

At Paris 6 University, a module "Computer and Representation" is proposed to students within a training of one semester in their second year of DEUG MIAS<sup>5</sup>. 450 students are involved in the module. We offer them as usual lectures coupled with tutorial classes which prepare them for their practical classes. Because of the great number of students and the lack of lecture rooms, the students cannot benefit of hours in self service, but they work in UTES.

Other autoevaluations in computer science have been available in Paris 6 for several years. The first module offered to students in DEUG MIAS depends on a set of Dr Scheme and the autoevaluations associated with the mediated teaching analyse the students' answers by means of the evaluator devised by Dr Scheme itself (Brygoo and al. 2002). Unlike Scheme, our goal with our exercises in computer architecture is not of programming in a language but more of understanding the described notions.

The 8086 Intel family processor has been chosen as the practical support of the course. Its architecture is simple enough for teaching the basic concepts of computer architecture to beginners. We use the Debug, a debugger tool (see MS-DOS reference manual), to realise short programming sequences, but more to read directly in the memory and understand its secrets. The first part of the course deals with computer architecture, representation of integer numbers and arithmetic with Boolean algebra. Then the instructions classes and the operands format are presented. The second part concerns the jumps, the stack with procedure calls and the interfaces with the system. A third part deals with the execution time of an instruction, the arithmetic with integers and the floating numbers. A one-semester introductory computer architecture course is difficult for beginners, who are not familiarised with these notions, even with the vocabulary used. In order to help the teaching of these notions, some teachers propose to use a personal simulator (David and al. 2002). For our module, we built a structured set of exercises for autoevaluation. That set is bound to complete the traditional teaching, but by no means to replace it. Its aim is to allow each student to control his own understanding of the subject.

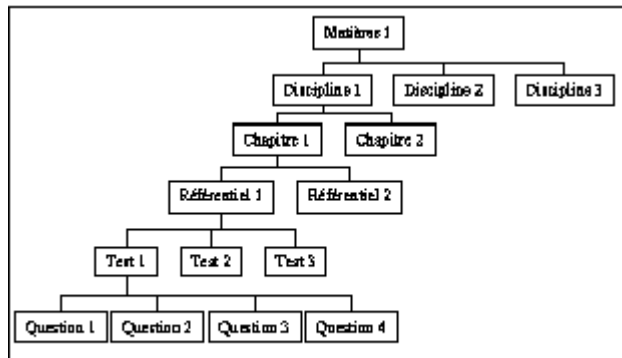
## **CONTENTS**

Teachers of two domains are involved in the autoevaluation: A.-M. Leseney (ARPe<sup>6</sup>) in biology, and our team in Computer Science. The resource is structured according to the pedagogical concept of J.P. David (1993), (Bouvier and al. 2001) as presented in the following diagram:

---

<sup>5</sup> Diplôme d'Etudes Universitaires Générales Mathématiques Informatique et Applications aux Sciences

<sup>6</sup> Atelier de Recherche Pédagogique



Example of a structure

In a Domain (Matière in the diagram), say "Computer Science", it is possible to build a set of exercises about several subjects (Discipline), say "8086 assembly language". The set of exercises is organised into chapters (Chapitre). A chapter is composed of referentials (Référentiel). A referential corresponds to a know-how and can involve several tests, each composed of a series of questions.

In our domain, two subjects have been developed: Pascal language and 8086 assembly language. The first one developed, Pascal language, involves nearly 200 exercises, corresponding to about thirty hours of work for the student. This set of exercises has been created by A. Brygoo, M.-C. Nogier and M.-M. Paget in the late nineties. It is no longer used in MIAS because we do not teach the Pascal language in this section anymore. Nevertheless students in other sections of DEUG Science still benefit from this set to control their knowledge of the Pascal language.

The subdivision in chapters we adopted for 8086 assembly language, is as follows: Numeration, Arithmetic and logic operations, Inside the processor Intel 8086, Instructions (the first ones), Addressing modes, Control structures, Format of the 8086 Instructions, Stack, Translating an easy algorithmic programme in assembly language, Representation of real numbers, Programmes (loops, recursion), (see the details at <http://www-asim.lip6.fr/~mp/Uteval.pdf>). We have chosen this subdivision in chapters to be adapted to our traditional teaching. However the choice of the structure of the know-how is here specific to autoevaluation and contains transversal references. For example the notion "sequence break" of our course appears in several chapters: control structures, programmes, translating an easy algorithmic programme in assembly language.

All the exercises cover most of the course taken into account for the examinations. The exercises are of various types that we have identified as important models in our subject: questions on the course, questions with multiple answers, programmes to read and understand (see the example below Figure 1), non complete programmes to complete, programmes to write. Solving all the exercises (about 150) in assembly language would require more than ten hours work from the student. In a proposed test in a referential, the exercises are classified from the easiest to the most complex, without any special delimitation. They are more detailed than the ones given in tutorial classes, in order to enable the student to work alone in front of his computer. The answers are intentionally very plain to allow those who have not been successful in solving the problem to find a detailed explanation there. We even sometimes propose several solutions which are then analysed.

## UTEVAL SOFTWARE

Uteval software proposes two pedagogical interfaces, one for the student and one for the teacher. A tool allowing the teachers to have a look at the contents of the exercises and print them is also available, if required.

## Student interface

File Edit View Go Bookmarks Tools Window Help

http://upipc100.cicrp.jussieu.fr/autoeval/informatique/dir2/dir1

testf.html

Assembleur Instruction Addition

informatique:

Question

On considère le programme suivant :

```

23B5:0100    ADD    CL , CH
23B5:0102    JC     0100
23B5:0104    INT    20

```

Au départ les contenus des registres sont les suivants  
**CX** = 8080    **CS** = 23B5    **IP** = 0100

1- Après l'exécution de l'instruction d'adresse 0100, donner les nouveaux contenus de **CX**, **CS**, **IP** ainsi que les indicateurs de condition **CF** (carry) et **OF** (overflow) .

2- Même question après l'exécution de l'instruction d'adresse 0102.

3- Comment se poursuit le programme par la suite ?

Temps indicatif = 300 secondes

**Solution**

**Solution**

	Note
1- Après l'exécution de l'instruction d'adresse 0100, CL + CH est mis dans CL soit 80 + 80 = 00 mis dans CL avec retenue débordante et dépassement de capacité. <b>CX</b> = 8000 <b>CS</b> = 23B5 <b>IP</b> = 0102 <b>CF</b> = 1 <b>OF</b> = 1	2
2- Après l'exécution de l'instruction d'adresse 0102, le compteur ordinal est mis à 100 car il y avait carry : <b>CX</b> = 8000 <b>CS</b> = 23B5 <b>IP</b> = 0100 <b>CF</b> = 1 <b>OF</b> = 1	2
3- Le programme se poursuit en exécutant ADD CL,CH et CL + CH est mis dans CL <b>CX</b> = 8080 <b>CF</b> = 0 <b>OF</b> = 0 Le compteur ordinal CO (ou IP) est alors mis à 0104, il n'y a pas saut car pas de carry et le programme se termine avec INT 20.	2

Seuil critique = 2    Note max = 6    Votre note (0-6) =     Suite

Figure 2

The student has at his disposal the text of the exercise and the indicative time proposed with a gauge (Figure 1). When the time is over, the gauge is full and becomes red, but the student can continue to work on the exercise, knowing however that he works too slowly. To come to an exercise he has to choose a chapter among the 12 proposed then a referential in the chapter. Finally he chooses a test in the referential and has to resolve all the questions of the test. For each of them, he can make the solution appear when he wants (Figure 2). The proposed solution gives him the detailed marking, so that he is able to mark his own work. The critical threshold indicated is the minimal mark that the student must have for this question.

When all the questions of a test have been solved and marked, a page giving the global results appears. This outcome is composed of the marks given by the student and the times used to solve the problems. The student can then determine if he has acquired the know-how associated with this referential when he compares his overall marks with the critical threshold. If at the end of one test the critical threshold of the mark is not reached, the student can have at his disposal another test where the questions and answers would be slightly different, thus allowing him to take again the same referential with an alternative.

## Teacher interface

In the first prototype the questions and answers had to be introduced directly in the HTML pages. In 2000 N. Bouzaïene and S. Brouillet have worked out a new and more convivial interface that can be used even by

teachers not familiarised with computer.

A group of referenced teachers (with login and password) in any domain can intervene in the exercises basis at every level using the teacher interface. After the welcome page, the teacher's login and password allow him to choose a domain, then a subject, here "8086 assembly

language". After that he creates or chooses a chapter, a referential, a test and finally a question. At each level, he can either add an item, either take it away, or modify it.

The proposed window to enter the exercise and its solution, is very easy to use. Two modes exist to enter the data: from an editor by Cut/Paste in the small window above, or by typing directly in the window intended for questions and answers. Pictures can also be inserted. The scale of the marking must be indicated for the complete question in an area called *Points* and it is also possible to introduce middle points so that the student can mark his work much precisely.

### **Explorator**

A very useful tool for teachers, built in 2002 by T. Gullaude, a student in training with us, shows the contents hierarchy and gives the possibility to see and print the selected exercises and their solutions, in order for example to control the accuracy of the text parts entered.

## **EVALUATION WITH QUESTIONNAIRES**

Our exercises have been used by numerous students: either at home, either in free use in UTES all the year round, either during tutoring sessions. We have opened these sessions so that we could give immediate answers to the additional questions the students would ask.

In order to have an evaluation of the work, we proposed questionnaires organised from a corpus of questions, to be filled by the students. This evaluation method is easy to manage and to exploit for the closed questions and brings precious information from the free questions. The questionnaire is composed of 33 paragraphs for evaluation. It is presented at: <http://www-asim.lip6.fr/~mp/Uteval.pdf>.

Here is the synthesis of the answers given by the questionnaires. The students have used Uteval either at UTES, during 2 hours (25%), 3 hours (35%), 4 hours (25%), 5 hours (3%) or at home. There was no problem in the use of the software. Some students (21%) have done all the chapters methodically, but most of them have not. The chapters which have not been done are different depending on the students. Probably, good students are interested in difficult exercises whereas students of a lower level prefer the easy exercises. According to the students (90%), the exercises were well chosen, the easy ones as well as the difficult ones. They also said that their number was sufficient to reach a reasonable proficiency of the subject. One student has regretted that there were not more difficult exercises, it must be said that this student had solved all the problems in order to obtain a distinction.

The detailed correction has allowed 96% of them to correct their errors easily. The solving time proposed has seemed adequate for 58% of the students. Some of the students have found the time allowed too short for some chapters. It is then difficult to take into account these answers, as the mentioned chapters are different, depending on the students. 64% have appreciated to be able to mark their work themselves. The layout of the terms of the problem (84% of the students), of the solution (88%) and of the marks in the solution (61%) suited them. It is the autoevaluation before a test or before the examination that is the main goal of the students and 95% estimate that Uteval has played its part. We can even boast to have met a student girl thanking us with this sentence: "Madam, I have succeeded the examination thanks to Uteval".

More than half of them (61%) have been led to learn new concepts. 96% have however claimed that the vocabulary used was the one of the lectures and that all the chapters seen were in the courseware. They are unanimous in affirming that Uteval has allowed them to understand

notions that were not obvious for them and that they are more able now to solve the problems (85%). For them the main assets of Uteval are the clear and explained solutions, and also the time management. Those who attended sessions at UTES have also much appreciated that one teacher was present during the autoevaluation to help them by giving another type of explanation at the very moment they needed it.

## **CONCLUSION**

The set of exercises, presented in this paper, under the domain "Computer Science", in the subject "8086 assembly language", has been used for 2 years with DEUG MIAS students. The interface is plain for these students. Their satisfaction incites us to increase the set of exercises and at the same time to improve the two interfaces of Uteval .

In installing our set of exercises in Computer Architecture, we have helped many students to evaluate themselves in order to be able to progress in their learning for their examinations. We hope then to have convinced our students that assembly language programming is not a difficult subject to understand and learn, as they often tend to think at first.

Thus implemented, the concept of autoevaluation was a success and the results can be interpreted in other contexts. Some public relations action is now needed to convince colleagues from other domains, to build a comparable structured set of Uteval exercises.

**Acknowledgments:** This work could not have been carried out without the support of Nicole Bernard, director of UTES. We thank her very much for her precious help. We are also grateful to A.-M. Leseney (ARPe) for numerous enriching discussions.

## **REFERENCES**

- BOUVIER D., BOUZAIENE N., BROUILLET S., LESENEY A-M., NOGIER M-C., PAGET M-M. (2001). UTEVAL Logiciel d'Autoévaluation, rapport interne UTES, University of Paris 6 (France): 13 p.
- BRYGOO A., DURAND T., MANOURY P., QUEINNEC C., SORIA M. (2002). Un cédérom pour Scheme. Chacun son entraîneur, un entraîneur pour tous: TICE 2002, Technologies de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie, Lyon (France): 223-231.
- DAVID J.P. (1993). Un modèle d'organisation des connaissances pour le logiciel AUTOEVAL: Technical report, Collection Grenoble Sciences mai 93.
- DAVID J-P., COGNE A., DUTEL A. (1996). Hypermedia exercises prototyping and modelling: CALISCE'96, Computer Aided Learning and Instruction in SCience and Engineering, San Sebastian (Spain), LNCS 1108.
- DAVID J-P., FOURNIER J.F. (1995). Mise en œuvre de l'autoévaluation formative. Un usage des NTE au premier cycle sciences de l'UJF: Xème colloque du CIPTE, Montréal.
- DAVID J-P., PEYRIN J-P. (2001). Training student to produce hypermedia resources, CALIE'2001: Computer Aided Learning In Engineering, Tunis (Tunisia): 112-117.
- DAVID J-P., ADAM J-M., PAUMIER J., PROMAYON E. (2002). Création de ressources Informatiques Multimédias pour l'enseignement de l'informatique (CRIMEI): TICE 2002, Lyon (France): 343-344.