

Second assignment guide Knowledge Based Systems

Artificial Intelligence laboratory

2012/2013 FALL TERM



Master in Information Technology

Departament de Llenguatges i Sistemes Informàtics



FIB

Facultat d'Informàtica
de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Contents

1	Organization, evaluation and delivery of the assignment	2
2	Goals of the assignment	3
3	The problem	4
4	Assignment Guide	6
5	Task planning	9
6	Evaluation Rubric	11

Organization, evaluation and delivery of the assignment

This is the guide for the second laboratory assignment, in this document you can find:

- The learning goals of the assignment corresponding to the course syllabus
- The description of the problem that you will have to solve
- The points that you have to include in the report to deliver as the result of the assignment
- The weekly planning of the assignment including the goals to fulfill each week and the estimated work load
- Evaluation rubric of the assignment

The assignment has to be solved by a **team of three** students. Do not try to solve the assignment alone, if you can not find a teammate talk to your laboratory professor.

You must develop the assignment using the **CLIPS** language. How to use this language will be explained in laboratory class. You have also additional documentation about this language on the laboratory webpage.

It is important to plan the development of this assignment and do not do all the work at the last moment. This assignment guide will give you directions about how to develop the assignment and will help you to plan your work.

The grade will depend on the quality of the analysis of the problem, the adequacy of the coverage of the ontology and the quality of the solutions that can be obtained. Read chapter about the evaluation rubrics for a detailed description of the grading criteria.

The written report and the code are due by **December 23rd**, the delivery will be electronically following the instructions that will appear in the racó.

There will be also a face to face presentation of the assignment during the days of **January 9th, 10th and 11th**, your laboratory professor will publish the schedule of presentations.

Goals of the assignment

The goal of this assignment is to deal with a problem that needs expert knowledge to be solved. To solve the problem it is necessary to address problems of analysis and or synthesis

The specific goals of this assignment are:

- To analyze the nature of the problem and to identify the sources of knowledge that are needed to solve the problem.
- To use the methodology for the analysis and implementation of knowledge based systems explained in the theory classes.
- Formalize and represent the knowledge needed to solve the problem. A methodological approach to the building of the ontology has to be used and the decisions taken have to be properly justified. The result has to be an ontology that can be used for the solution of the problem.
- To identify the kind of problem solving methodology more adequate for the problem and to define the solution of the problem using this methodology.
- To implement the solution of the problem using a rule based language.
- To apply the software engineering methodologies for rapid prototyping and incremental development
- To choose adequate and demonstrative test cases to show the functioning of the solution and to explain the results of the tests.
- To have contact with languages for knowledge representation and knowledge based systems development. A certain proficiency and maturity on the use of the CLIPS language has to be shown.

With respect to the **evaluation** of the assignment, you have available a rubric with the criteria that will be used and a description of the grade levels of each criteria.

The problem

In order to be prepared for the usual new year resolutions (get fit/lose weight), the gym chain *No pain no gain* wants to develop a KBS able to recommend training programs for its future clients.

For the recommendation process, the first step is to establish the general physical conditions of the client. To do that we need a set of basic data like weight, height (from these two we can compute the body mass index), age and blood pressure (high and low). We can also ask the client to perform a set of simple exercises, like running fast during a minute or climbing stairs at a normal rhythm, to obtain some information like the heart beat rate per minute, sensation of tiredness, dizziness or muscular stress.

We want also information about their daily routine, we want to know about the types of physical activities that the clients engage during work time (sitting, standing, walking, repetitive movements, weight lifting, heavy muscular activity, ...), out work activities (static (watch television, reading, ...), household activities (ironing, cleaning, dish washing, ...), displacements (walking while shopping, leisure walking, walking pets, ...)). From all these activities we are interested on their frequency and duration. To engage in more or less physical activity during the daily activities will give us an idea about the initial intensity that the user can endure and what goals are already covered by the daily routine.

We need also information about their health, like for example skeletal and muscular problems (back pain, joints pain, cervical pain, ...) or diet habits (vegetables-fruit consumption, excess of salt, eating between meals, ...).

Apart from this information it will be also necessary to know the goals of the exercise program, for example to keep in shape, to regain shape, lose weight, gain muscle/strength, gain flexibility, improve balance, ... We need also to know the time available to perform the exercises (at least 30 minutes per day).

The system knows about a set of exercises that can be performed inside a gym that can cover the different goals that a user can look for (an exercise could cover different goals), such as exercises with apparatus (static bicycle, running machine, rowing machine, stepper, weight lifting, ...), exercises with or without weights for the different muscular groups, floor exercises, exercises for stretching/flexibility, ...

All these exercises have a set of characteristics and constraints, for example, the number of calories burn for unit of time, duration (minimum, maximum), number of repetitions (minimum, maximum), muscular groups trained, if it is not adequate for specific conditions (high blood pressure, muscular/joint problems, ...), if it can alleviate certain conditions (back pain, limited mobility, ...), if it is not adequate from certain age, ... We have also information about what exercises combine better with other exercises. The difficulty of an exercise (moderate, normal, high) can be from the exercise itself, related to the number of repetitions or related to the physical shape of the user (to perform 5 abdominal curl-ups can be easy for somebody in a normal shape and slim, but difficult for a over-weighted person)

The system has to generate a training program for a week, each day will consist of a sequence of

exercises adequate for the time the user has available. The exercises have to be selected taking in account the health and physical constraints of the user. The exercises should be mainly oriented to fulfill the goal that the user has indicated, taking in account the initial physical shape of the user. The exercises included have to be diverse enough from day to day and during the whole week to avoid monotony.

Apart from your own experience and other sources of knowledge that you can access, you can obtain information about exercises using Google, you can try with keywords like *exercises & fitness*, *exercises & calories*, ... You can also have an idea about the possible solutions looking the keywords *exercises & routines*. You do not have to be exhaustive with the list of exercises, but there have to be enough to solve the different scenarios that you propose. The values of the attributes for the exercises do not have to be exact (you can just make them up) but try to use values that make sense, use your commonsense.

You have to implement this system and write a report about it that has to include:

- An analysis of the problem **following** the steps of the knowledge engineering methodology explained in class (identification, conceptualization, formalization, implementation and testing). You have to detail these steps in your documentation. Think that the conceptualization and formalization steps include the ontology, so, you have to explain with detail how it has been designed.
- The ontology created to represent the knowledge that the system needs (you can generate the description and the graphical representation for the documentation using Protégé)
- A set of non trivial tests that demonstrates how your program solves different cases.

Additionally to the documentation of the assignment you must include the commented source code of the system you have implemented.

The grade will depend on the quality of the analysis of the problem, the adequacy of the coverage of the ontology and the quality of the solutions that can be obtained. Read the evaluation rubrics for a detailed description of the grading criteria.

Assignment Guide

First week: CLIPS/Assignment description (October 29th)

This first week has to be devoted to read and comprehend the assignment description and to have the first contact with the CLIPS language.

You have to use this week to read the documentation about the CLIPS language and to look into the simple examples that you have available. You have a FAQ (in spanish/catalan) linked from laboratory web page. This FAQ has been elaborated by past students from the course and probably will answer the majority of your questions during the development of the assignment, you also have some practical advice about how to develop your program.

You have to keep in mind that CLIPS is different from the programming languages you are used to, because of that, is important that you get acquainted with the language as early as you can and look into the examples. This will save you time when you start the development of the assignment.

While you are making yourselves comfortable with the language it would be interesting that you start thinking about the domain of the assignment and on the ontology that you will need.

You can start designing it on paper, thinking about the concept and attributes that could be interesting to have. You can read the paper “Ontology Development 101” that you have in the theory web page on the additional material on theme 3 to have some points about what methodology use.

If it was important during the first assignment to plan your work ahead, in this assignment it is even more important. Do not let the time pass and start working too late, seriously.

Second week: The specification of the problem (November 5th)

This week is assigned for the specification of the problem. You need to have a clear vision about the elements that you will need to build the ontology for the domain of the assignment. Given that the idea is to use an incremental development approach it has not to be the final ontology.

This week in the laboratory class the functioning of the CLIPS environment will be explained. It is important that you look into the examples that you have available, specially the solutions of past semesters assignments. You can also try to execute them, the description of the assignments they correspond to are available on the “other material” web page.

This week the tool Protégé will also explained. It is fairly easy to use. You can use it to design the ontology and to export it to the language that CLIPS uses. You can also generate the documentation of the ontology exporting it to HTML. It exports the ontology hierarchy and the description of the classes and attributes. Obviously you have to include the documentation of every element in Protégé first.

You can also browse the ontologies of past semesters assignments.

Before beginning the development of the ontology read first the documentation available about how to develop an ontology.

Third week: The ontology (November 12th)

During this week you should have a good idea of what are you going to include in the ontology. As it was said before, it has not to be the final version, probably you will need to make some changes while development advances.

You will need also to create some instances. It is not necessary to include all the instances you need from the beginning. You can have as first goal to solve a limited set of simple problems and to decide what concepts from the ontology will be required, creating the instances accordingly.

Keep always in mind the philosophy of incremental development, divide the problem in modules and subproblems and focus on building an initial functioning prototype that you can extend to the final system.

Fourth week: Subproblems decomposition/Initial prototype (November 19th)

One of the key points in order to start the implementation of the system is to obtain a decomposition of the problem in subproblems that give the different steps that lead to a solution.

You should keep in mind that some of the problem solving methodologies explained in theory classes has to be applied in the implementation of the assignment. This means that the problem decomposition has to be embedded in the phases of these methodologies. You can use the methodologies as a guide form the decomposition.

When you write the assignment report you will have to explain the decomposition of the problem (conceptualization) and how the subproblems are embedded in the problem solving methodologies (formalization).

From this decomposition you can implement a few rules that solve a simplified version of the problem. This will be the initial prototype that will allow you to evaluate the decisions taken so far and give some guidance about how to extend and complete it. The goal of this initial prototype is to detect in the early stages of development any problems in the ontology or in the design of the system.

It is a good idea to choose some test cases that the system should be able to solve and focus the development in creating the rules to solve them.

Fifth week: Extension of the prototype (November 26th)

At this time you should have a pretty good idea of the design of the system and should have validated all the design decisions taken.

- The ontology should be final
- The problem decomposition should have resulted in the rule modules of the system
- You should have a set of test cases to develop the initial prototype and its extensions

From now on you have to extend the prototype to include new cases. The cases should be chosen in order to obtain a set of test cases diverse enough to test the different possible scenarios of the problem.

Do not forget to work on the assignment report, it is not a good idea to wait until the last moment. You can start documenting the final ontology, do not only describe its classes and attributes, explain how you designed it, indicate the necessary justifications for your decisions and do not omit any decision you have made.

Sixth week: Final prototype (December 3rd/December 10th)

At this time all the design decisions should be final.

You should think about the test cases and see that the result that the system gives are correct. The test cases should be representative of the different scenarios that the problem can have and your system can solve.

The test cases should be diverse enough respect the possible elements from the domain and their complexity. You can use these test cases also as the test cases you will show in the documentation of the assignment, so you are killing two birds with a stone. Save the results that you obtain and document them adequately. You should be able to explain the results of the test from the knowledge that has the system and the information that represent the case.

Do not forget about the documentation, do not let the time pass and think that it has to show the work you have done.

Seventh Week: Test cases and documentation (December 17th)

This week you need to have the results of the test cases that you have decide to include in the documentation. The results have to be documented adequately, explaining the test case and the results that the system obtain.

The rest of the documentation has to describe the development of the assignment. The documentation has to be structured following the phases of the development methodology that has been explained in theory class.

Task planning

This planning can be useful as an orientation about how organize and divide the work of the assignment. Notice that a careful planning of the work will benefit the quality of the result.

The workload in hours of the assignment, considering the ECTS credits it has, is 25 for each member of the team plus 7 hours of laboratory class. This make a total of 111 hours during six weeks. Notice that these are ECTS hours, so this is the workload that the average student has to use to solve the assignment to an acceptable level (lets say a B+ grade = 8 over 10 points) to obtain an A grade (10 over 10 points) you will need some more hours of dedication.

This table includes all the task to be solved during the weeks described in this guide. The approximate weekly hours of dedication of the group are also indicated.

Week 1 (3h lab + 8 hours of work)
1. Begin to learn how to program using the CLIPS language 2. Browse and run the examples 3. Read the CLIPS snippets document 4. Read and understand the description of the assignment 5. Think about the domain of the assignment from the expert point of view, look for sources of knowledge
Week 2 (3h lab + 10 hours of work)
1. Look the flash presentation about how to use Protégé and the CLIPS environment 2. Build an informal version of the domain ontology (first each member of the group separatedly, then merge both designs together) 3. Look into the solutions published on the web of the assignments of past semesters 4. Think about the resolution of the problem from the expert point of view
Week 3 (3h lab + 12 hours of work)
1. First version of the ontology in Protègè + its documentation also using Protégé 2. Find a set of problems that should to be solved by the final system and think about how the experts would solve them 3. Describe some of these problems using the ontology 4. Plan the incremental development of the assignment from the the problem and decide the goals of the first prototype 5. Document the problems and the development plan

Week 4 (3h lab + 12 hours of work)
1. Think about the resolution of the problem from the knowledge engineer point of view (problem resolution methodologies, problem decomposition, modularization of the resolution process) 2. Document the resolution methodology that will be used and how the problem decomposition can be embedded in the methodology 3. Development of an initial prototype able to solve an instance of a simple problem
Week 5 (3h lab + 15 hours of work)
1. Extend the initial prototype to solve more problems
Week 6 (3h lab + 10 hours of work)
1. Complete the prototype 2. Generate the test problems with the problems used to develop the system 3. Decide some additional test problems and execute them on the system 4. Document the results (justify the test problems, explain their solution)
Week 7 (3h lab + 8 hours of work)
1. Complete the documentation of the test 2. Finish the documentation of the assignment

The two first hours of laboratory class will help you to understand how CLIPS and Protégé work. The rest of laboratory classes have to be used to ask questions/doubts about the assignment, to synchronize your work with your teammate and to develop the assignment.

It is important that the tasks of the assignment are divided fairly and adequately between the two members of the group. Notice that most of the tasks can be performed in parallel. And remember that three people in front of the same PC for an hour are just an hour of work, not three.

Chapter 6

Evaluation Rubric

This is the evaluation rubric for the assignment. The grading of the assignment will use these criteria and will follow the indication for each grade level.

You must follow these criteria when you write the report for the assignment to explain the work that you have done and how you have done it.

Indicator	level	Inadequate	Needs improvement	Adequate
Use of the design and implementation methodology	Identification	The description of the problem is a verbatim copy of the description from the assignment.	- The description of the problem is a verbatim copy of the description from the assignment. - A brief analysis of the viability of the construction of a KBS for the problem is included. - A clear description of the goals of the problem and the expected results from the system to be implemented is included.	- The description of the problem is more elaborated than the one from the description of the assignment - A detailed analysis of the viability of the construction of a KBS for the problem is included. - The identification of the knowledge sources is included. - A clear description of the goals of the problem and the expected results from the system to be implemented is included.
Conceptualization		- There is no description of the concepts involved in the domain conceptualization. - There is no description of the problems and subproblems involved in the resolution of the problem. - There is no informal description of the whole resolution process.	- Description of the concepts of the domain. - Brief description of the problems and subproblems involved in the resolution of the problem.	- Description of the concepts of the domain. - Detailed description of the problems and subproblems involved in the resolution of the problem. - Informal description of the whole resolution problem and how the problems and subproblems participate in the resolution.

Indicator	level	Inadequate	Needs improvement	Adequate
Formalization		- There is no explanation about how the ontology has been built. - The ontology has a few concepts, a few simple attributes and no relations. - The ontology is not documented. - There are no algorithmic explanation about what each identified subproblem does. - There is neither an explanation about what problem resolution methodology is more adequate nor a description about how the subproblems identified are embedded in the methodology steps.	- There is a brief explanation about how the ontology has been built. - The ontology includes most of the relevant concepts, a few simple attributes and some relations. - The documentation of the ontology is superficial (only is stated what each concept means). - There is a detailed algorithmic description of each identified subproblem. - The more adequate problem resolution methodology is only mentioned.	- There is a detailed explanation about how the ontology has been built. - The ontology includes all the relevant concepts, all the necessary attributes and all the relevant relations. - The documentation of the ontology is detailed (all the elements of the ontology are described), and a graph of the concepts hierarchy is included. - A detailed algorithmic description of each identified problem is included. - A justification of the problem resolution methodology is given and an explanation about how the subproblems identified are embedded in the methodology steps is included.
Implementation		- The ontology is implemented inadequately. - The resolution of the problem is not modularized. - The representation of the resolution process using rules does not follow the steps of the chosen problem resolution methodology. - No development methodology has been used.	- The ontology is implemented adequately. - The resolution of the problem is not modularized. - The representation of the resolution process using rules follows the steps of the chosen problem resolution methodology. - The development methodology is not incremental and no intermediate prototypes has been implemented.	- The ontology is implemented adequately. - The resolution of the problem is modularized following the identified subproblems. - The representation of the resolution process using rules follows the steps of the chosen problem resolution methodology. - An incremental development methodology has been used, intermediate prototypes have been implemented and all this is adequately described in the documentation.

Indicator	level	Inadequate	Needs improvement	Adequate
Testing Coverage		- A few test cases (<4) and all very similar. - There is no explanation about how the test cases have been chosen.	- A few test cases ($\geq 4, <6$) but fairly different. - Only the use cases used in the development have been tested. - There is no explanation about how the test cases have been chosen.	- Many test cases (≥ 6) and fairly different. - Some test cases are different from the use cases used in the development. - There is an explanation about how the test cases have been chosen.
Explanations		Only the raw output is included.	The output is included and only a brief explanation of the results is included.	The output is included and the results are adequately explained from the inputs of the test and the knowledge that the system has.
CLIPS implementation				
Knowledge of the language		- Mainly unordered facts are used. - The rules are not organized. - Use of global variables.	- Deftemplates and objects are used. - There are modules and basic control mechanisms. - The consequent of the rules does almost everything and the rule declarative paradigm is not used.	- Deftemplates and objects are used. - There are modules and elaborated control mechanisms. - The implementation uses deduction chains and uses the rule declarative paradigm.
Quality of the solution				
Coverage		Superficial coverage, only the simplest cases are treated.	Average coverage, only the most common cases are treated.	Advanced coverage, common cases and exceptions are both treated.
Documentation		Bad presentation, no coherently structured, explanations difficult to understand, careless.	Passable presentation, coherently structured, not much elaborated explanations, imprecise and vague.	Good presentation, adequately structured and following the steps of the development methodology, well elaborated explanations, concise and precise.
Adjustment to the assignment specifications		- Some elements of the problem have been simplified - Not all the constraints of the problem have been taken in account - The recommendations about the implementation of the assignment have not been followed		The assignment has been developed accordingly to the documentation indications

level		level	
Indicator	Inadequate	Needs improvement	Adequate
Elaboration of assignments documentation Spelling and grammar	The document has lots of misspelled words and grammatical errors. The document is of very low quality in this aspect.	The document has a few misspelled words and some grammatical errors.	The document is grammatically correct and have no misspelled words.
Organization	- The document is incorrectly organized. - There is not an introduction. The development of the document does not follow a coherent line of reasoning. The document changes from a subject to another without a logical order. There is neither a final summary nor a conclusions section.	- The organization has some aspects that can be improved. - The introduction does not describes completely the problem, or the different sections do not follow well a correct line of reasoning, or there are no final summary and/or conclusions.	- The document is well organized. - The different sections of the document follow a clear line of reasoning and make easy to understand the content of the document. - The document has an introduction describing the problem, some sections developing the problem and a final section summarizing the results and giving the final conclusions.
Clarity	The document is difficult to understand; the sentences are too long and confusing; the reader frequently has to reread some parts of the text to understand it and in some cases he does not understands it at all.	There are some sentences that are too long and/or confusing, sometimes they have to be read repeatedly to finally understand their meaning.	- The contents of the document are clear. - The sentences are short and easy to understand.
Use of graphics and figures	Graphics and/or figures are not used to make the explanations easier to understand, or the ones used do not clarify anything.	- In some parts of the document some graphics or figures to clarify the explanations should have been used. - Some of the graphics or figures do not clarify the explanations or are unnecessary.	- All the graphics and figures used are illustrative and help to the explanations. - Only the necessary graphics and figures are used.