USING BIG DATA FOR IMPROVING STUDENTS’ SKILLS IN THE DEVELOPMENT OF SCALABLE DECISION SUPPORT SYSTEMS

Iván García-Magariño¹,², Inmaculada Plaza³,², Jorge Delgado Gracia⁴, Raquel Lacuesta¹,², Raúl Igual⁵,², Javier Bielsa-Hernández¹, Diego Álvarez-Arguedas¹

¹ Department of Computer Science and Engineering of Systems, University of Zaragoza (SPAIN)
² Instituto de Investigación Sanitaria Aragón. University of Zaragoza. Zaragoza (SPAIN)
³ Department of Electronics Engineering and Communications, University of Zaragoza (SPAIN)
⁴ Department of Applied Mathematics, University of Zaragoza (SPAIN)
⁵ Department of Electrical Engineering, University of Zaragoza (SPAIN)

Abstract

In most courses of computer science grades, students develop systems that are usually tested only in reduced data sets. These systems are still normally far from the common kinds of systems that current society needs. In particular, many current systems deal with huge amounts of data. Several existing works recommend teachers to promote the competence of collaborative development of solutions that can be applied in realistic scenarios. Acquiring the skill of collaborative development of scalable solutions can be useful to prepare students for the real problems that they will address in their professional careers.

The present work belongs to the current trend of works that refer to some concepts of the “big data” field as the key to teach the aforementioned skill. More concretely, the current work proposes a guideline for designing collaborative learning activities to develop solutions that can receive input from big data. These activities normally encourage students to present novel solutions for solving certain problems in realistic scenarios. Students are gathered in teams, and each of these works collaboratively to reach to the corresponding solution. In most of these activities, the teacher suggests the main topic. However, students decide which particular subtopic they will address as long as the teacher agrees with them.

The presented guideline has been experienced with a pilot study in the course of Decision Support Systems (DSS) in the computer science grade of the University of Zaragoza. In this course, the teacher showed some of the main free available big data repositories to the students. Each group of students selected a problem that could be solved by a DSS after they reached a consensus with the teacher. Students worked collaboratively to develop scalable software applications that were robust and efficient enough to use big data for supporting users’ decisions. The teacher supervised the developments of all the teams assisting them in overcoming technological problems and taking some design decisions. At the end of the course, each team of students performed a presentation of their DSS to the other students and the teacher. In addition, all the students replied a short questionnaire for measuring (a) their perception about the learning activity, and (b) the learned knowledge about DSSs. Furthermore, the classmates evaluated each DSS with two well-known user experience scales: Usefulness, Satisfaction and Ease of Use (USE) questionnaire, and the System Usability Scale (SUS).

This paper presents one of the DSSs developed by the students as an example of the final work product that students can develop in learning activities designed with the present guideline. The paper also analyzes the results of the questionnaires.

Finally, this paper discusses the possible future impact of the presented guideline and determines the next steps to extend the current approach to other courses and grades. These future steps aim at obtaining an interdisciplinary guideline for designing collaborative learning activities in which students learn to propose scalable solutions in realistic scenarios by means of big data from free available repositories.

Keywords: Big data, collaborative learning activity, decision support system, software development education, teamwork.
1 INTRODUCTION

Computer science engineering grades usually include competencies related to the collaborative
development of software applications in realistic scenarios [1]. In order to test these applications in this
kind of scenarios, the applications should use large amounts of real data.

The current work proposes to design software development activities that use either data from big
data repositories or websites with big data [2]. In this way, students can learn to develop software
applications for realistic scenarios. In addition, they can also improve their collaborative skill as these
activities are recommended to be teamwork activities.

This work introduces a guideline for designing this kind of activities. This guideline is intended for most
subjects of computer science engineering grade. However, the current article presents its application
only in the Decision Support Systems (DSSs) subject.

The current work follows our previous research line for assisting teachers in designing collaborative
activities, either with an agent-based simulator [3, 4] or through a 3D virtual world [5]. However, none
of our previous works considered the use of big data resources in learning activities as the current
work does.

2 GUIDELINE FOR DESIGNING LEARNING ACTIVITIES WITH BIG DATA

The main goals of the proposed guideline are that students acquire (a) the necessary development
skills to obtain software applications in realistic scenarios, and (b) collaborative skills for teamwork.

The teacher is recommended to schedule very few learning activities in the corresponding subject of a
semester, so that students have enough time for developing applications that use big data. An ideal
option would be to plan only one activity for the whole subject. However, teachers should be aware of
two facts when planning the schedule of a subject with only a learning activity. First, the teacher
should support and evaluate the whole development process. In this way, the teacher can advise
students about good practices in programming and software development, allowing students to learn
from their errors before reaching the end of the course. Second, the learning activity should be useful
to cover all the relevant aspects of the subject. Thus, the activity should integrate the practice of the
relevant concepts of the corresponding subject.

This guideline suggests that the teacher explains certain concepts surrounding big data and provides
the necessary information to access and manage information from either big data repositories or
websites with large amounts of data. The teacher introduces a list of big data repositories that include
the most relevant ones such as the national official ones for USA (data.gov) and UK (data.gov.uk).
These kinds of repositories include large amounts of data in different file format such as CSV (comma-
separated values) and XLS (Microsoft Excel file format). The teacher should explain how to import
data from at least one of these formats. In particular, students can learn how to import CSV files into a
database using any database management system such as MySql.

Besides the big data repositories, there are other websites with large amounts of data like some real-
estate websites (e.g. idealista.com). These could be also used by students in this learning activities.
Usually, it is not so easy to access to large amounts of data from these websites. However, these data
commonly have the advantage of being less exploited, so students may have the opportunity of
developing novel applications using these data.

The teacher can explain how to efficiently manage these data in an application for final users. More
concretely, all the raw data can be aggregated in less amounts of data that summarize certain
meaningful properties of the global data set. The aggregation of data can be performed beforehand.
Then, the user application can access the aggregated data for obtaining a proper performance in
terms of time response.

The activities are recommended to be performed by teams of several students to address complex
software developments and to improve the collaboration skills.

Furthermore, the teacher should reinforce the motivational aspect of the activity, since this kind of
activities may require high effort from students. A way of increasing the motivation of the students is to
leave the decision of the final domain of the application to them, although the teacher requires some
general aspects about the kind of application. In this manner, each team can select its preferred
domain. Before starting the development, each team should discuss the specification of the
application with the teacher, so that the development effort does not exceed the common amount of
work of a subject.

Finally, it is worth noting that the teacher is responsible of finding the way of integrating the concepts
of a specific course into a learning activity when following the presented guideline. The next section
shows an example of application of this guideline in a specific course.

3 APPLICATION OF THE GUIDELINE IN THE SUBJECT OF DECISION
SUPPORT SYSTEMS

A learning activity was designed with the presented guideline for the subject of Decision Support
Systems (DSSs). In particular, this activity was the only one in that subject for the first semester of the
course 2016-17. The teacher introduced some big data repositories and taught how to extract data
from these. He also introduced other websites with large amounts of data. In particular, he taught how
to import CSV files into MySql databases. He also provided examples of how aggregating data with
SQL queries.

The teacher asked students to develop a DSS that used data obtained from either a big data
repository or a website with large amounts of data. The teacher recommended students to aggregate
data to obtain a knowledge base with meaningful information. The teacher introduces several
examples of DSSs that used aggregated information. In addition, the teacher taught the common
techniques for developing DSSs as part of the subject theory, explaining the background about
supported decision-making.

The teacher discussed the possible domains with each team of students. The teacher solved the
doubts of students during the development and advised them in some aspects of the development.

The teacher advised students to use the Java programming language and the MySql database
management system, however he gave them the possibility of using any other programming language
or database management system.

The next section presents one of the DSSs developed of students so that readers can have an
example of the kinds of applications that can be developed with this learning activity.

4 AN EXAMPLE OF DSS DEVELOPED BY STUDENTS: RECOMMENDATION
OF REAL-ESTATE PRICES

4.1 Introduction

This DSS helps a user to rent or sell a flat for the most appropriate price, according to several
parameters such as the number of rooms and the number of bathrooms among other examples.

This project has an academic interest, on the one hand, as well as a research one on the other hand.
The students got a learning result by acquiring practice and also taking the opportunity to work with
large amounts of data. From a research point of view, the project presents a system that is built from
scratch and aims at improving the existing ones.

There are a few examples of this type of systems, but they lack access to prove their operation for
free.

The development of this DSS prototype pursued several goals: (a) to learn how to build a new
software that supports decisions, (b) to build a knowledge base, (c) to build a DSS that recommends
prices according to some parameters, (d) to program a new software that will be the base for other
improved applications and for freeware, and (e) to look up a big data base for producing the
knowledge base.
4.2 DSS representation and implementation

The DSS presented in this work has been based on the idea of making new software that provides the best recommendation of the price of a flat that a person owns, depending on whether it is for rental or for sale.

The first step was to quantify the flat given by the user, establishing which features of the flat were relevant. These features are also referred as dimensions from this point forward. In particular, we selected the following features:

- Number of rooms
- Number of bathrooms
- Area
- Lift (yes or no)
- Type of housing unit (Flat, Duplex, Attic, Cottage and Rustic)

After defining the dimensions, it was necessary to assign importance to each one of them. For this, a market survey was carried out for three age groups, taking a sample of 50 students and administrative and service personnel from the University of Zaragoza in Teruel. The results of the survey can be seen in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>18 – 35 years</th>
<th>36 – 50 years</th>
<th>51 – 100 years</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rooms significance</td>
<td>50 %</td>
<td>50 %</td>
<td>30 %</td>
</tr>
<tr>
<td>Bathrooms significance</td>
<td>10 %</td>
<td>20 %</td>
<td>10 %</td>
</tr>
<tr>
<td>Area significance</td>
<td>30 %</td>
<td>20 %</td>
<td>30 %</td>
</tr>
<tr>
<td>Lift significance</td>
<td>10 %</td>
<td>10 %</td>
<td>30 %</td>
</tr>
</tbody>
</table>

Note that one of the most important dimensions when making the recommendation is the number of rooms, while the area has a relatively similar relevance in the three groups.

In the study of each dimension the equals are crossed in this dimension, or neighbors if there were none, and an average market value is assigned to that dimension. For this purpose, the size of the type of dwelling has been taken into account when establishing this average value; because the type dimension is a non-quantifiable dimension and that the fact of adapting the rest of dimensions to it occupies many computational resources (especially RAM memory) and may not be needed.

After explaining the dimensions used in the project, we then go on to explain the data structure used. It is a ADT (Abstract Data Type) stored on an indexed map, where each element has a reference to its next neighbor in the dimension. This data structure was chosen because of its performance in the search operation.

It is necessary to mention that when this search is performed, the entire dimension is scrolled, visiting the elements with a value. In this way, a tree structure was not chosen because it would have implied higher computational cost and memory usage, which is not desirable. This ADT was specifically programmed for this project. Note that it was done manually, without using the libraries of the programming language.

For the implementation of the DSS, it was decided to use the Java language for the following reasons among others:
- It includes a nice user interface framework (i.e. Swing).
- The implementation of the ADT was relatively simple.
- The language was known by the developer team.
- It is multi-platform, and it is desirable to execute the application in different operating systems.
- There is much documentation available for this language.

The students designed the DSS and presented the UML (Unified Modeling Language) class diagram of figure 1 as a result.

![Diagram](image)

**Figure 1. Class diagram of the DSS for price recommendation.**

It is worth mentioning that there may be more than one knowledge base for this project, because each one is loaded from only part of the data, such as sales or rentals of a specific zone.

The rental recommendation part is not functional in this version of the project, because it contains three non-quantifiable dimensions, and during development, an adequate solution to the treatment of these was not found due to the lack of time.

The graphical part of the application was programmed using the graphic library of Java Swing, obtaining the graphical interface of the application shown in figure 2.

It is worth mentioning that "Recommendation deviation" refers to a discount applicable to each dimension considering Gaussian bell, the explanation of which will be expanded in section 4.4.

After searching information for the realization of this project, the students selected the website www.idealista.com (hereinafter Idealista) for extracting the necessary data for their project.

To obtain data, the Idealista website offers an Application Programming Interface (API) for academic or research work. Access to this API is done using the Curl protocol, similar to the Get protocol. The implementation of this protocol is done by downloading the software corresponding to the protocol itself and a certificate of validation.
However, establishing the connection to the API was not possible in the given development time because the documentation was not clear enough according to the students. Then, it was decided to obtain the data manually to carry out the project, and creating the knowledge base from these data.

### 4.3 Works related to this DSS

Auctioneer [6] is a plugin for the famous videogame “World of Warcraft”. This plugin helps the player to set up the price of sale for an object of the game. This is based on similar or equal objects that exist in an auction. In figure 3, one can see how the plugin works. In this case, the data that is included on the data base must be traversed whenever they share the same features. The plugin proposes a recommendation of the price that considers it will be sell considering objects with similar or equal features. This algorithm is well known, and its normally referred as the K-Nearest Neighbors algorithm [7]. This system recommends a price of sale with virtual money.
A subpage of Idealista allows one to ask for a price recommendation for a flat based on certain parameters, somewhat similar to the main objective of this project, being the disadvantages of this software created by Idealista in comparison to this project:

1. One needs to pay to get the price recommendation.
2. It takes about a week to get the price recommendation.

In spite of these important disadvantages, it has a very important advantage:

1. The system sends a pdf file to the user's email, and the information of this file is very detailed.

However, the current DSS offers several improvements with respect to Idealista:

1. The application is free.
2. The recommendation is done automatically and instantly.
3. The price is shown in the graphical user interface of the application in which the input data have been introduced.

### 4.4 Testing and assessment of the DSS

Figure 4 shows an example of execution of the DSS, in which a price is recommended based on the features of a house and the target age for a certain city area.

![Figure 4 – Example of a recommendation of the DSS](image)

The students performed an analysis of the results obtained in the tests of functionality of the application. After seeing and analyzing the results, the students discovered three facts, which are described in the subsections of this section.
4.4.1 The “Type” dimension

When assigning the recommendation, the quantifiable dimensions "collided" with the non-quantifiable dimensions, so, when going through these dimensions, if the non-quantifiable dimension matched, it was more important when assigning a value for the recommendation.

Due to this fact, a house with the same characteristics will have different recommendation depending on its type.

4.4.2 Considerations about the Gauss Curve

When studying a quantifiable dimension, it is seen that there can be an arrangement of elements with series of values similar to this Gauss bell effect (see figure 5). The regions of the extremes could be neglected. However, in this case they cannot be neglected because they are part of the market, and failing to do so, could produce certain bias in the analysis.

![Figure 5 – Generic Gauss Curve](image)

The DSS provided an "discount" input parameter to calibrate the output of the system in order to obtain a fair and a good price that might not be necessarily the minimum available one in the market. This parameter allowed users to determine a discount over the common price for a house with certain features.

4.4.3 Strange results when not any similar house

When analyzing a dimension, if there are no elements with the same characteristic, the most similar elements will be traversed. This causes that the value that is given by rooms to a house that has twelve rooms, may be smaller than a house that has three rooms.

It is believed that these facts have appeared because the study sample was limited, and that, applied to a wider sample of data, this effect would be reduced and almost imperceptible.

5 RESULTS OF THE EXPERIENCE

Six students conducted the course of DSSs. They conformed three teams. All the students finished their projects about the presented learning activity. In addition, all the students passed the course with a grade of seven out of ten or above.

As the result of the teaching experience, section 5.1 presents the perception of the students that developed the DSS of the presented case study. Section 5.2 indicates the results about the usability and satisfaction validated scales for this DSS obtained from peer-evaluation of the classmates.

5.1 Perception of the students

The students of the presented DSS wrote the following perceptions about their experience with the presented learning activity:

*"The expectation of creating a DSS from scratch was good initially because the team of project developers believed that there would be a lot of information on how to do the code for such a system, but the reality was not so. After spending countless hours looking for information, the expectations..."
went from good to bad for being aware of the huge effort necessary for developing systems for realistic scenarios.

Fortunately for the team, the teacher of the subject guided in the programming of the knowledge base and the recommendation engine, obtaining a functional application that met the expectations in recommending a price based on the characteristics of the flat that the user wanted to buy or rent.

The realization of this work has brought a great effort for the development, mainly for the lack of experience when programming this type of applications. Nevertheless, it was gratifying to see that the DSS finally worked and incorporated certain advantages over similar systems.

5.2 Peer-evaluation about usability and satisfaction

Four classmates evaluated the presented DSS by means of the validated System Usability Scale (SUS) [8] and the dimension of satisfaction of the Usefulness, Satisfaction, and Ease of Use (USE) questionnaire [9]. In particular, the usability of the system obtained an average mark of 70.6 in the 0-100 range of SUS. The classmates expressed a satisfaction mark of 58.9 in the 0-100 range of the corresponding dimension of the USE questionnaire.

6 DISCUSSIONS AND CONCLUSIONS

The proposed guideline has allowed a teacher to successfully design a learning activity for guiding the work of students in the subject of DSSs. Although the students could get too high expectations at the beginning, it is probably positive that students confront realistic scenarios before ending the computer science engineering grade. This can make students really prepared for their professional careers.

The current experience shows the possible drawback that this kind of activities might be too demanding for students. However, this is only based on the perception of students, which might not be accurate. In order to assess this aspect, the spent time should be measured and compared with the standard recommendations.

The fact that all the students successfully finished their projects in the learning activity reveals that this kind of activities might motivate students appropriately.

Despite the great efforts on behalf of the students, they felt satisfied with their work done for their projects obtaining novel and useful software applications, which in the current experience were DSSs.

Furthermore, the final DSS product of the selected case study was evaluated as usable as one can observe in the result of the corresponding validated scale.

The current work is planned to be experienced with more students for the same subject, to obtain results that can be considered representative. In addition, the presented guideline is planned to be applied to more subjects from the computer science engineering grade and the electronic and automation engineering grade, to determine its utility in different subjects and different engineering grades.

ACKNOWLEDGEMENTS

This work acknowledges the research project “Diseño de actividades de aprendizaje colaborativas con Big Data” with reference PIIDUZ_16_120 funded by University of Zaragoza.

REFERENCES


