From GreedEx to GreedEx Tab v2.0: Tool for Learning Greedy Algorithms on iPad Following CIAM Mobile Methodology

Yoel Arroyo, Manuel Ortega Cordovilla, Miguel A. Redondo, Ana I. Molina,
María del Carmen Lacave, and Manuel Ortega Cantero(✉)

Escuela Superior de Informática, Universidad de Castilla-La Mancha,
Paseo de la Universidad, 4., 13071 Ciudad Real, Spain
{Yoel.Arroyo,Manuel.Ortega1,Miguel.Redondo,AnaIsabel.Molina,
Carmen.Lacave,Manuel.Ortega}@uclm.es

Abstract. The growth in the use of mobile devices have allowed for the development of new educational modalities within e-Learning. Because of this, we have shown the evolution experienced by GreedEx, a desktop application that facilitates the learning of greedy algorithms. Two iPad versions were developed (GreedEx Tab v1.0 and v2.0). The first iPad version adapted many functionalities to mobile devices, whilst the latest one adds several pedagogical usability characteristics. For that, we followed the CIAM Mobile methodology, improving the educational and collaborative aspects. Our main aim is to verify that this methodology is useful for the production of new methodological approaches aligned with the principles of MDE (Model-Driven Engineering) that gives support to learning activities within a group. Therefore, this work can be considered a preliminary m-CIAM test case .

Keywords: Mobile usability · Mobile learning · Methodology · Greedy algorithms learning · Knapsack problem · iOS · iPad · Model-driven development

1 Introduction

The growth in the use of mobile devices have allowed for the development of new educational modalities within e-Learning. Because of this, we have shown the evolution experienced by GreedEx, a desktop application that facilitates the learning of greedy algorithms. Two iPad versions were developed (GreedEx Tab v1.0 and v.2.0).

GreedEx Tab is an application for iPad developed by the CHICO (Computer-Human Interaction and Collaboration)[1] research group to facilitate the learning of greedy algorithms [1]. In the latest version, we wanted to solve the usability problems found in its predecessor [2] and add several pedagogical characteristics. Furthermore, the application presents groupware characteristics, allowing both students and teachers to work collaboratively on a learning task thanks to iCloud.

[1] http://chico.esi.uclm.es/cms/.

G. Meiselwitz (Ed.): SCSM 2017, Part II, LNCS 10283, pp. 44–54, 2017.
DOI: 10.1007/978-3-319-58562-8_4

Thus, GreedEx Tab v2.0 has been developed by following the CIAM Mobile (m-CIAM) methodology [3]. This new methodology assumes a quality leap over the CIAM (Collaborative Interactive Applications Methodology) methodology [4] by integrating it with the MoLEF (Mobile Evaluation Framework) usability framework [5], providing evaluation and design mechanisms to facilitate the development of collaborative mobile learning systems.

Our main aim is to verify if this methodology is useful for the production of a new methodological approach aligned with the principles of MDE (Model-Driven Engineering) that gives support to learning activities within a group. Therefore, this work can be considered a preliminary m-CIAM test case.

The structure of the paper is as follows: Section two introduces similar programming and algorithm tutoring systems that already exists and the current state of the methodologies and frameworks implied. In the next section, the evolution GreedEx experienced and the obtaining of the different versions is explained. Then, we talk about some current and future work, with m-CIAM methodology and MDE as a starting point. Finally, we close this paper with some conclusions related to this work.

2 Previous and Related Work

It is important to teach novice programmers how to analyze a given problem, identify the computing requirements, and develop an algorithm appropriate for its solution. The main problem is they are not aware of designing an algorithm before writing code. They believe that developing an algorithm will increase the time necessary to do an assignment. However, many students struggle through the programming process, and end up spending a considerable amount of time debugging numerous logical or design errors.

Nowadays, there exist similar algorithm tutoring systems such as GreedEx. For example, AlgoTutor (The Algorithm Tutor) [6] is an online, graphical and interactive tool which provides automatic assessment of student designed algorithms. In addition to the student algorithm design interface, there is an instructor interface that allows the teacher to create/edit problems and their solutions, manage student accounts, and analyze student grades as well as student activities. As far as disadvantages are concerned, it is just for novice programmers.

Another tool of this kind is ANIMAL (A New Interactive Modeler for Animations in Lectures) [7]. It offers a selection of powerful features that can easily be combined to create and display animations of algorithms, data structures and many other topics. Animations can be generated using a visual editor, by scripting or via API calls. The main advantage is that it has a good collection of algorithms of different categories, including backtracking, graph, searching and sorting, trees, hashing, etc. So, it is usually used by intermediate level programmers.

Similar systems such as JHavé [8] or AlgoViz [9] can serve the same purpose. However, if we focus on greedy algorithms visualization and experimentation, we should talk about GATutor (Greedy Algorithms Tutor) [10] and GreedEx (Greedy Experimentation) [11]. GATutor is a rule based framework for teaching greedy algorithms. It allows the user to develop an algorithm providing them stimulating questions and timely hints to real life scenarios. Later, all this data could be analyzed by the system providing insights to the teacher. GATutor is a good choice, but the learning process is relatively more complex than GreedEx. The main difference is that GATutor could be considered a visualization system program, usually designed to aid in analyzing program behavior, whilst GreedEx objective is less general, designed to aid students in understanding algorithm behavior. In Sect. 3 more details about GreedEx and its different versions are described.

The second and latest GreedEx Tab version was developed following the m-CIAM methodology. It is a methodology used in the development and modeling of m-Learning applications. It integrates the CIAM methodology [4] with MoLEF [5] usability framework factors and dimensions. CIAM provides support in the modeling of collaborative and interactive aspects of e-learning systems, while MoLEF incorporates evaluation or analysis techniques for mobile learning applications. In other words, MoLEF adds support for mobile learning applications evaluation and implementation where CIAM does not have.

Currently, we are validating the m-CIAM methodology in other scopes such as MDE (Model-Driven Engineering). As we said in section before, our main aim is to verify if this methodology is useful for the production of new methods like these. Therefore, the design and implementation of GreedEx Tab v2.0 may be considered further proof of this methodology, apart from getting an improved version of the original application.

3 From GreedEx to GreedEx Tab v2.0

GreedEx (Greedy Experimentation) [11] is a desktop application developed by LITE (Laboratory of Information Technologies in Education)[2] research group for discovery learning and experimentation of greedy algorithms. Their original intention was to develop an educational system to support any greedy algorithm. However, due to the huge difference in the visualization of the different greedy algorithms this goal should have been abandoned by designing a system that supports a fixed but extensible set of problems.

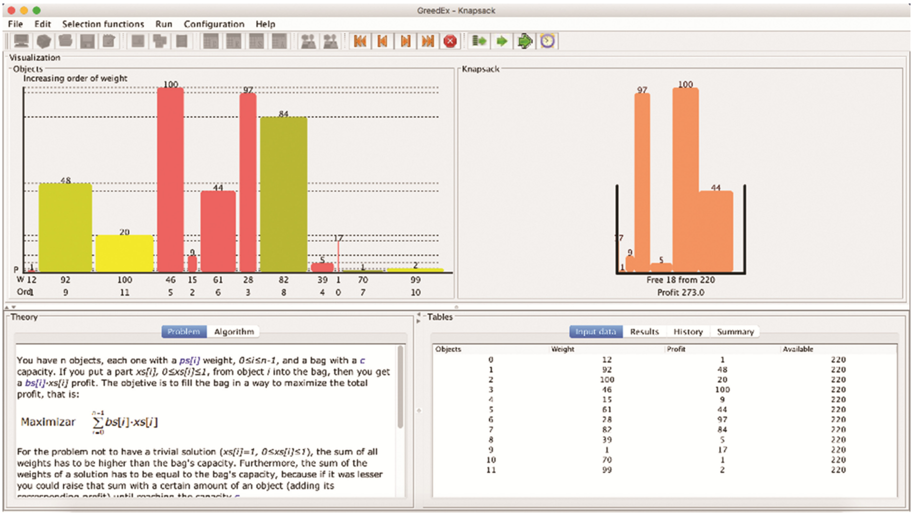
Currently, GreedEx supports six optimization problems, the activity selection problem [1] and five knapsack problems: The fractional and the 0/1 knapsack problems, as well as three variations with different maximization measures (number of objects introduced into the knapsack, weight of these objects, and number of objects introduced into two knapsacks).

The main goal is to state a selection function for one of these problems and prove its optimality. Which means, the student must figure out all the selection functions that could characterize an optimal greedy algorithm for a given problem. To accomplish this, GreedEx provides several selection functions for each problem. A selection function is defined on the set of available candidates, returning to each step the most promising candidate with respect to some measure. For example, in the knapsack problem users may select among the *increasing order of index, decreasing order of index, increasing order of weight,*

[2] http://www.lite.etsii.urjc.es/.

decreasing order of weight, increasing order of profit or *decreasing order of profit* between other selection functions. They must interact with them and seek the maximum size subset of objects to be included in the knapsack, discarding the non-optimal ones.

For that, GreedEx provides a user interface with three different areas clearly distinguishable. Figure 1 shows an interactive session with the knapsack problem. The top area is the visualization panel, which shows the graphical representation of the problem. The lower left area is the theory panel, with two tabs: one to show the problem statement written in natural language, and the other tab shows the solution to the greedy algorithm written in pseudocode. The lower right area is the table panel, with four tabs containing the input data table, the results table, the history table, and the summary table. This user interface is possible in desktop devices due to the huge size of the screen. In mobile devices, such as a smartphone or iPad as it is the case, it must be analyzed for new design and development, adapting its functionality and interface taking advantage of interaction, visualization and animation of these kinds of devices.



You have n objects, each one with a *ps[i]* weight, *0≤i≤n-1*, and a bag with a *c* capacity. If you put a part *xs[i], 0≤xs[i]≤1*, from object *i* into the bag, then you get a *bs[i]·xs[i]* profit. The objetive is to fill the bag in a way to maximize the total profit, that is:

$$\text{Maximizar} \quad \sum_{i=0}^{n-1} bs[i] \cdot xs[i]$$

For the problem not to have a trivial solution (*xs[i]=1, 0≤xs[i]≤1*), the sum of all weights has to be higher than the bag's capacity. Furthermore, the sum of the weights of a solution has to be equal to the bag's capacity, because if it was lesser you could raise that sum with a certain amount of an object (adding its corresponding profit) until reaching the capacity *c*

Objects	Weight	Profit	Available
0	12	1	220
1	92	48	220
2	100	20	220
3	46	100	220
4	15	9	220
5	61	44	220
6	28	97	220
7	82	84	220
8	39	5	220
9	1	17	220
10	70	1	220
11	99	2	220

Fig. 1. GreedEx session with knapsack problem

With this purpose, our research group evaluated GreedEx in collaboration with the LITE research group through an eye tracking tool (the Eye-Tracker Tobii ×60) available in our laboratory in the Computer Science Engineering College of Ciudad Real. The results of this evaluation [12, 13] identified some shortcomings in its user interface and learning methods that suggested that a more interactive tool could improve this learning. One of the main complaints between students were that the graphical representations was lacking in an interactive nature and were difficult to understand, so they preferred the tabular data representation to consult the results and interact with the problem. For that reason, what we considered interesting was to build up GreedEx Tab [2], an application where graphical representations takes on greater role by improving the interaction within the representation of the results, amongst other things.

GreedEx Tab [2] was developed in our laboratory for this purpose. The selected device was an iPad. The larger size of the screen was one of the motives for choosing this mobile device, in comparison with others such as an iPhone, where graphical representations could be difficult to understand and to develop.

This version was obtained following SCRUM [14] as management methodology along with OpenUP [15] and CIAM [4] as development methodologies. SCRUM and OpenUP are agile methods to work with based on the *Manifesto for Agile Software Development* [16]. For that, we considered them ideal for small projects like GreedEx Tab. More details about these methodologies and how to follow them could be reviewed [2], where the application obtained was explained.

While SCRUM and OpenUP guides the project via sprints and the typical phases such as initiation, elaboration, construction and transition, CIAM gives support to modeling the collaborative and interactive aspects. It is really useful for applications such as GreedEx Tab whose main premise is to reach a healthier computer-human interaction.

We tried to make it intuitive and fast, allowing users to interact in the simplest way possible so that learning does not become a complicated and tedious task. To achieve that, a series of usability tests were carried out [17] during its implementation process. These tests were performed in our laboratory, consisting of a series of activities to do in the GreedEx Tab application with the iPad device, registering and recording all activity with the eye tracker. Before and after performing the tests on the application, the participants had to fill out a questionnaire concerning their knowledge of greedy

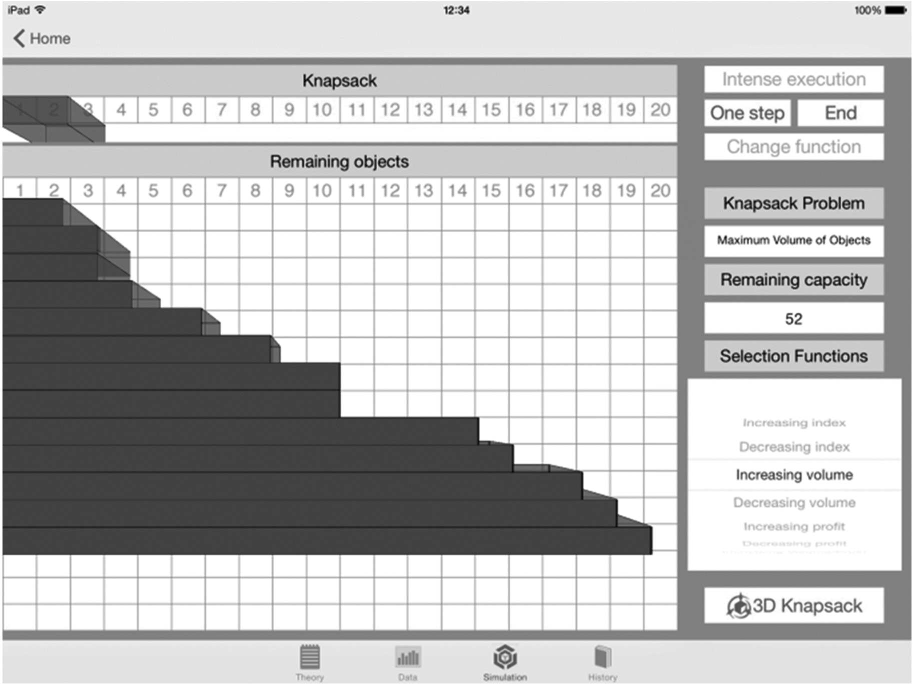


Fig. 2. GreedEx Tab session with knapsack problem

algorithms and their experience with the application respectively. All the information collected was finally analyzed, determining that the initial shortcomings in the graphical user interface and in the learning methods detected were resolved. In this version, the graphical representations as shown in Fig. 2 are the most representative and consulted, instead of the tabular representations of GreedEx.

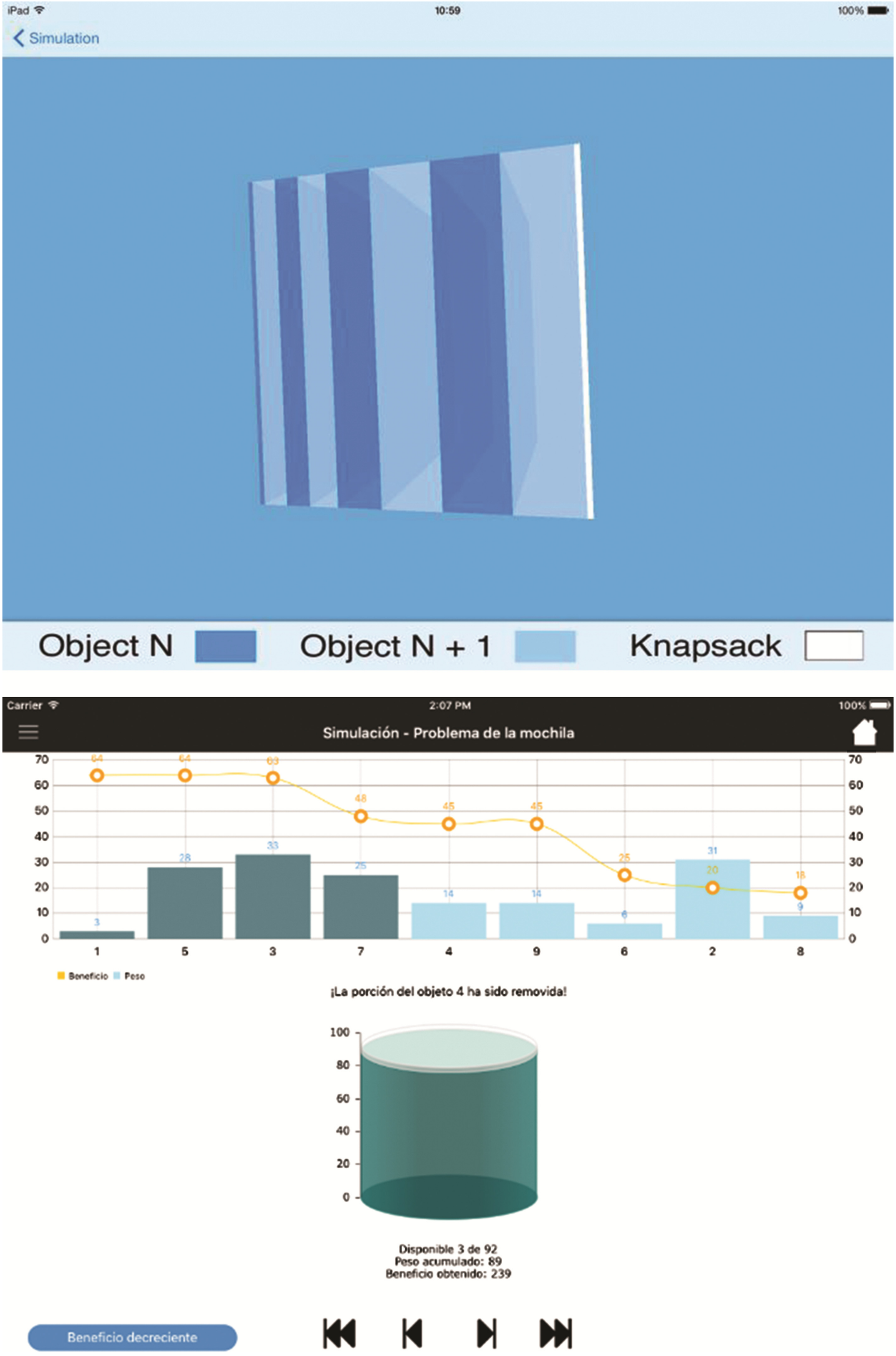
However, the results obtained were not satisfactory at all. We could appreciate that these representations increased the use of the graphical representations but continue to be difficult to understand, obtaining only acceptable values in this area. For example, users complained about the 3D visualization of the knapsack because they considered it not intuitive, making the learning process difficult. For that reason, we decided to enhance the user interface again, trying to solve these problems using other methods that could help us obtain improved pedagogical and technological values.

Moreover, looking at [2], some usability principles were taken into account in the design and development process, such as the golden rules of Ben Shneiderman [18], the established for Apple for its applications [19] and the usability patterns for mobile devices defined by Jacob Nielsen [20]. Nevertheless, these were not enough to achieve our objectives.

Thus, we must mention the addition of MoLEF [5] in the implementation of GreedEx Tab in its second version. MoLEF is a design and evaluation framework for collaborative m-Learning systems. It provides some guidelines that developers must take into account to obtain improved pedagogical and user interface usability values. In contrast with CIAM, MoLEF is focused on the development of mobile learning applications. That is the way m-CIAM arose, with their integration, providing an updated and more powerful methodology.

The results were satisfactory in general, solving several of the complaints about GreedEx Tab. For example, the knapsack of the problem was replaced by a barrel metaphor (see Fig. 3). It is not only more attractive and comfortable, but more interactive, providing information of its state whenever the user wants just by touching it. It also provides a more detailed legend. Furthermore, the screen space is well exploited. As shown in Fig. 3, the state of the problem is always known because of the objects implied and the knapsack are in the same place, facilitating the learning process.

To achieve that we followed the m-CIAM steps as described [3]. To verify that this methodology is useful in obtaining better pedagogical and user interface usability values, an exploratory evaluation with real users was performed in our laboratory. Sixteen undergraduate students from our school took part in it. In this case, the students had to compare the first GreedEx Tab version with the latest one. For that, they accomplished a series of scheduled tasks with both applications while recording and registering all data with our eye-tracking tool. After that, they had to fill out the CECAM (Questionnaire for the Evaluation of the Quality of m-Learning Applications) questionnaire [21], which is part of m-CIAM and provided by MoLEF. This questionnaire is made up of 56 items, 29 of them corresponding to pedagogical usability and the rest in the user interface usability. It is really useful as an evaluation checklist for usability of m-Learning systems, but it can also be used as an heuristics checklist for its design, previously explained [3].



¡La porción del objeto 4 ha sido removida!

Disponible 3 de 92
Peso acumulado: 89
Beneficio obtenido: 239

Fig. 3. The representation of a knapsack on GreedEx Tab v1.0 and GreedEx Tab v2.0

The results obtained were positive. The new version received better pedagogical and user interface values, fulfilling the stated objective, making it more comfortable and usable in both pedagogical and technological aspects. In a comparative way, as shown in Table 1, the evolution experienced on some characteristics by the applications involved. In this table, we can see thanks to the follow up of m-CIAM, most of them were updated and improved in the latest version. Whilst, for example, the multilingual support on GreedEx Tab v2.0 is scheduled as future work.

Table 1. Comparative table between versions and its characteristics

Characteristic	GreedEx	Gx Tab	Gx Tab 2
Concepts explained and presented in a clear manner	X	X	X
Activities reflects practices relevant to professional life	X	X	X
Intuitive and consistent icons for actions	X	X	X
Appropriate feedback information	-	-	X
Navigation simple, familiar and intuitive	-	-	X
Adjusted and adapted information for screen size	X	X	X
Pleasant and attractive aesthetics	-	-	X
Interactive charts selected to facilitate learning	-	-	X
Collaborative functions for students and mentors	-	X	X
Interactive knapsack provided	-	X	X
Multilingual support	X	X	-

Thus, after joining all collected data and verifying the results, we can conclude that the stating students were more comfortable and satisfied with the latest version. The first iPad version introduced several updates and enhancements in the user interface, whilst the latest version also improved the learning methods. For that, we considered this methodology interesting for the development of applications of this kind.

4 Applying MDE for Generalizing Our Approach

In recent years, we are also working to provide a new methodological approach aligned with the principles of MDE (Model-Driven Engineering) that gives support to learning activities within a group. It is focused in the CSCL (Computer Supported Collaborative Learning) [22] paradigm, which exploits the advantages provided by groupware systems in the field of e-Learning.

The development of such systems is not a simple task. There are aspects such as the support to model collaborative processes, the roles supported by the system or the existence of spaces for sharing information, which has become key in design and implementation requirements [21].

That is the reason why CIAM methodology was firstly developed. It covers some groupware usability requirements (i.e., incorporating coordination and communication tools, a design-based on models of users and tasks, role modeling, etc.), providing a series of notations and stages to design collaborative systems. Nevertheless, it only supports the analysis and design phases, and other important aspects such as the implementation of communication mechanism, the access control to shared resources, the support to coordination, the management of sessions or the inclusion of awareness elements are not technologically supported.

For that reason, the SpacEclipse framework [23] was also developed. The main advantage is that, integrating it with CIAM, allows designers to generate a collaborative modeling tool adapted to any domain. More simply put, we could obtain collaborative synchronous modeling tools specialized for the domain of application previously modeled.

However, as we are focused in the CSCL paradigm it is necessary to consider its educational and pedagogical aspects during the different phases of the method. To achieve that, we used MoLEF as a reference, which provides these necessary criteria and guidelines. For instance, by integrating CIAM Mobile (giving support to the analysis and design phases) with the SpacEclipse framework (giving support to the implementation phase), full support for the development of CSCL tools may be obtained.

Currently, groupware engineers use these methodologies and frameworks in a 'manual' way. Thus, our further objective is to facilitate their work by providing a systematic technological support that takes into account pedagogical usability aspects. To achieve that, we are carrying out more exhaustive studies, with evaluations that include comparisons between tools developed using the method and without using it, and comparisons between tools generated using different characteristics of the method to achieve that. However, the results of these studies are outside the scope of this paper.

5 Conclusions and Future Work

In this paper, we have shown the evolution experienced by GreedEx, a desktop application that facilitates the learning of greedy algorithms. Two iPad versions were developed (GreedEx Tab v1.0 and v.2.0). The first version adapted many functionalities to mobile devices whilst the latest one improves user interface by adding several pedagogical usability characteristics, thus enriching the educational and collaborative aspects.

To achieve that, the latest version has been developed by following the CIAM Mobile (m-CIAM) methodology. This methodology assumes a quality leap on the CIAM methodology by integrating it with the MoLEF usability framework, which provides some evaluation and design guidelines to facilitate the development of collaborative mobile learning systems.

The results obtained were positive, obtaining better pedagogical and user interface usability values in general. The students were more comfortable with it. Moreover, the CECAM questionnaire and data collected with our eye-tracking tool verifies those results. Thus, we consider this methodology interesting for developing applications of these kinds.

Nevertheless, we are working to provide a new methodological approach aligned with the principles of MDE (Model-Driven Engineering) that gives support to learning activities within a group. This method is focused in the CSCL paradigm so it is necessary to consider its pedagogical usability dimension. Thus, we also must take as reference

MoLEF, and for instance, the m-CIAM methodology, which provides these necessary criteria and guidelines.

To reach our goal, we are carrying out more exhaustive studies, with further evaluations that includes comparisons between tools developed using the method and without using it, and comparisons between tools generated using different characteristics of the method.

Acknowledgments. This research has been partially funded by the Ministry of Science and Innovation through the project TIN2015-66731-C2-2-R and the JCCM Project PPII11-0013-1219.

References

1. Cormen, T.H.: Introduction to Algorithms. MIT Press, Cambridge (2009)
2. Ortega Cordovilla, M., et al.: GreedEx Tab: tool for learning Greedy algorithms on mobile devices. In: IARIA 2016, pp. 28–35 (2016)
3. Arroyo, Y., Navarro, C.X., Molina, A.I., Redondo, M.A.: CIAM mobile: methodology supporting mobile application design and evaluation applied on GreedEx Tab. In: Luo, Y. (ed.) CDVE 2016. LNCS, vol. 9929, pp. 102–109. Springer, Cham (2016). doi: 10.1007/978-3-319-46771-9_14
4. Molina, A.I., et al.: CIAM: a methodology for the development of groupware user interfaces. J. UCS **14**(9), 1435–1446 (2008)
5. Navarro, C.X., et al.: Framework to evaluate M-learning systems: a technological and pedagogical approach. IEEE Revista Iberoamericana de Tecnologias del Aprendizaje **11**(1), 33–40 (2016)
6. Pettey, C., et al.: A tool for promoting algorithm development in introductory CS classes. In: Proceedings of ED-MEDIA (2009)
7. Rößling, G., Schüer, M., Freisleben, B.: The ANIMAL algorithm animation tool. ACM SIGCSE Bull. (2000). ACM
8. Naps, T.L.: JHAVÉ: supporting algorithm visualization. IEEE Comput. Graph. Appl. **25**(5), 49–55 (2005)
9. Ullrich, T., Fellner, D.: AlgoViz-a computer graphics algorithm visualization toolkit. In: World Conference on Educational Multimedia, Hypermedia and Telecommunications (2004)
10. Lahoti, M.: GATutor: Intelligent Tutoring System for Greedy Algorithms. Indian Institute of Technology Bombay, Mumbai (2014)
11. Velazquez-Iturbide, J.A., et al.: GreedEx: a visualization tool for experimentation and discovery learning of greedy algorithms. IEEE Trans. Learn. Technol. **6**(2), 130–143 (2013)
12. Molina, A.I., et al.: Evaluación basada en eye tracking de las técnicas de visualización de programas soportadas por el sistema GreedEx
13. Molina, A.I., et al.: Assessing representation techniques of programs supported by GreedEx. In: 2014 International Symposium on Computers in Education (SIIE). IEEE (2014)
14. Schwaber, K.: SCRUM development process. In: Sutherland, J., Casanave, C., Miller, J., Patel, P., Hollowell, G. (eds.) Business Object Design and Implementation, pp. 117–134. Springer, London (1997)
15. Kroll, P., MacIsaac, B.: Agility and Discipline Made Easy: Practices from OpenUP and RUP. Pearson Education, Boston (2006)
16. Beck, K., et al.: Manifesto for agile software development (2001)

17. Ortega Cordovilla, M.: Usability Tests of GreedEx Tab. CHICO Wiki. http://chico.inf-cr.uclm.es/greedextab/wiki/index.php/Pruebas_de_usabilidad. Accessed 29 Dec 2016
18. Shneiderman, B., Plaisant, C.: Designing the User Interface: Strategies for Effective Human-Computer Interaction, vol. 5. Addison-Wesley Publishing Co., Boston (2010)
19. Apple Inc.: iOS Human Interface Guidelines. Apple Developer Website. https://developer.apple.com/ios/human-interface-guidelines/overview/design-principles/. Accessed 10 Jan 2017
20. Nielsen, J., Budiu, R.: Mobile Usability. MITP-Verlags GmbH & Co. KG (2013)
21. Molina, A.I., et al.: A Model-Driven Approach for the Development of CSCL Tools that Considers Pedagogical Usability
22. Koschmann, T.D.: CSCL, Theory and Practice of an Emerging Paradigm. Routledge, New York (1996)
23. Gallardo, J., Bravo, C., Redondo, M.A.: A model-driven development method for collaborative modeling tools. J. Netw. Comput. Appl. **35**(3), 1086–1105 (2012)