ParTec and KarnUMa – Digital tools as a help for freshman learning

Valentim Freitas[1], A. Ravelo-García[2] and F. Morgado-Dias[1]

[1]Madeira Interactive Technologies Institute and Centro de Competências de Ciências Exactas e da Engenharia, Universidade da Madeira
Campus da Penteada, 9000-039 Funchal, Madeira, Portugal
[2]Departamento de Señales y Comunicaciones. Universidad de Las Palmas de Gran Canaria,
Campus Universitario de Tafira, 35017 Las Palmas de Gran Canaria, Spain
`morgado@uma.pt`

Abstract. Freshman students always present lower success rates than other levels of students. Digital systems is a course usually taught at first year students and its success rate is not very high. In this work we introduce three digital tools to improve freshman learning designed for easy use and one of them is a tool for mobile terminals that can be used as a game. The first tool is ParTec and is used to implement and test the partition technique. This technique is used to eliminate redundant states in finite state machines. This is a repetitive task that students do not like to perform. The second tool is called KarnUMa and is used for simplifying logic functions through Karnaugh Maps. Simplifying logical functions is a core task for this course and although students usually perform this task better than other tasks, it can still be improved. The third tool is a version of KarnUMa, designed for mobile devices. All the tools are available online for download and have been a helpful tool for students.

Keywords: Karnaugh Map, Partition Technique, Digital tools, self-learning, Quine- McCluskey

1   Introduction

Success rates at University are becoming more and more important for both administrations and governments. University students represent a cost and failing adds costs to the general process. Freshman students usually exhibit the lowest success rates due to the lack of study habits and youth. This low success rate sometimes ends in the student abandoning the degree or changing to a different one.

Digital systems is a course taught at the first year of Computer Engineering and Electrical Engineering and its success rate is not very high in most universities. To improve this success rate three tools were created and introduced in the teaching process at the University of Madeira. These tools were designed for easy use and one of them is a version for mobile terminals that can be used as a game.

The first tool is called ParTec and is used to implement and test the partition technique. This technique is used to eliminate redundant states in finite state machines. This is a repetitive task that students do not like to perform. For this kind of problem no tool has been reported in the scientific community.

The second tool is KarnUMa and is used for simplifying logic functions through Karnaugh Maps. Simplifying logical functions is a core task for this course and although students usually perform this task better than others, it can still be improved. The third tool is a version of KarnUMa, designed for mobile devices. For this problem several tools have been developed by the scientific community, with [6-7] being examples of the most complete tools presented. Nevertheless none of them has all the features that KarnUMa has [5]. All the tools are available online for download and have been a helpful tool for students.
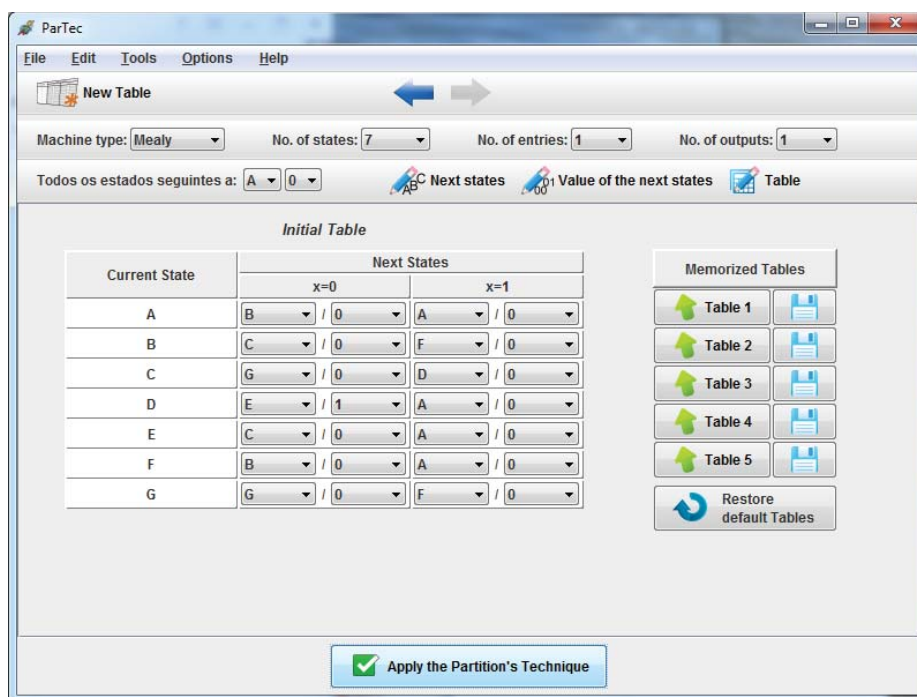

Fig. 1. ParTec main window.

2 ParTec

A poor definition of the Finite State Machine (FSM) can lead to the introduction of redundant states. This situation is more common in Mealy machines, especially be-

cause students learn about this type of machine after they learn about Moore machines. Nevertheless redundant states can appear in both types of FSM.

Figure 1 shows the main window of ParTec with an imported Partition table for a Mealy FSM. ParTec allows filling, saving and loading tables; choosing Moore and Mealy FSM, choosing the number of states, inputs and outputs and filling the table.

Once the Apply Partition technique button is pushed and the technique has been applied, the state tables and state diagrams can be analyzed. Figure 2 shows the situation obtained from the table presented in figure 1.
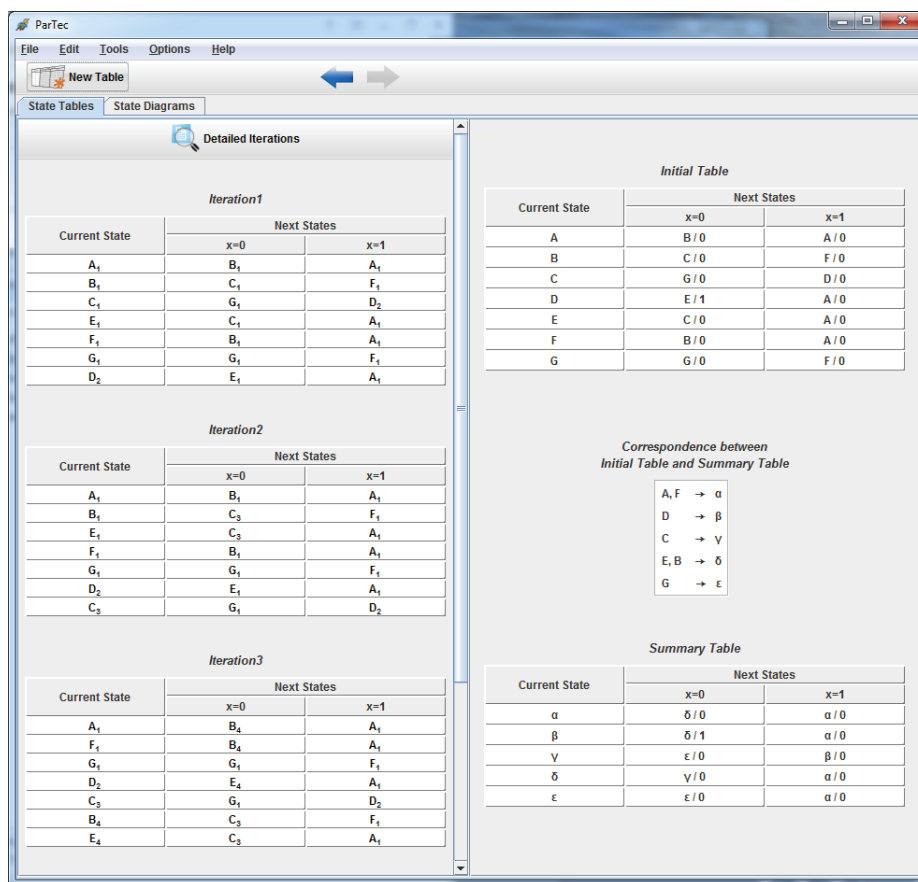

Fig. 2. State tables.

On the state table tab the application of the Partition technique can be seen. The left side shows the several iterations of the process while the right side shows a resume with the initial table and final table after redundant states are eliminated.

A step-by-step solution can also be seen by selecting Detailed Iterations on the top left of the window. This is shown in figure 3. Here each change in the partitions is

detailed through a graphical color based step-by-step procedure that will help the student to learn the technique.

In figure 4, the state diagrams can be seen. The left side shows the initial diagram while the right side shows the simplified version, after all the redundant states are eliminated. In this example two redundant states have been removed. The new state diagram is drawn using different names for the states to avoid confusing them with the original states.
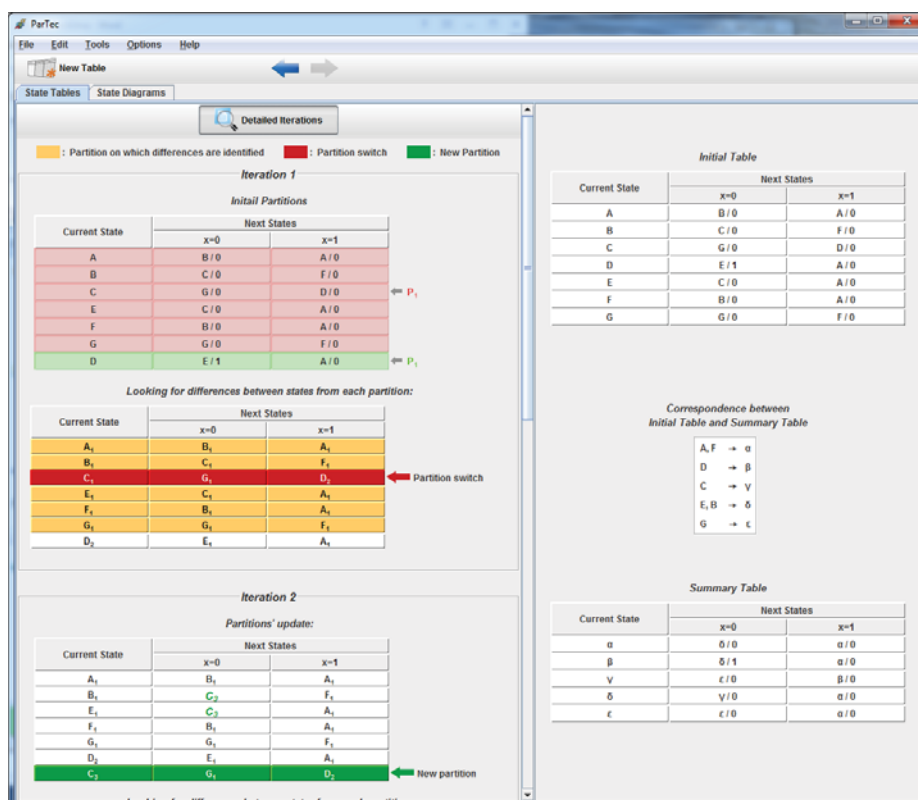

Fig. 3. State tables – detailed iterations.

The removal of redundant states can allow for using less hardware and is therefore a mandatory step in designing FSM. This tool helps students to learn the detailed step-by-step procedure and to practice since it can generate countless examples to practice on.

3 KarnUMa

KarnUMa is a tool to learn and practice the use of Karnaugh maps [5]. These maps are a graphical tool for simplifying functions. The tool was built in two versions: a Personal Computer (PC) version and a pocket version. Both versions use the Quine-Mckluskey [1-3] method and source code obtained from [4] to implement it.

Both versions of the tool are Java based, use two languages: Portuguese and English and allow Undo and Redo actions.
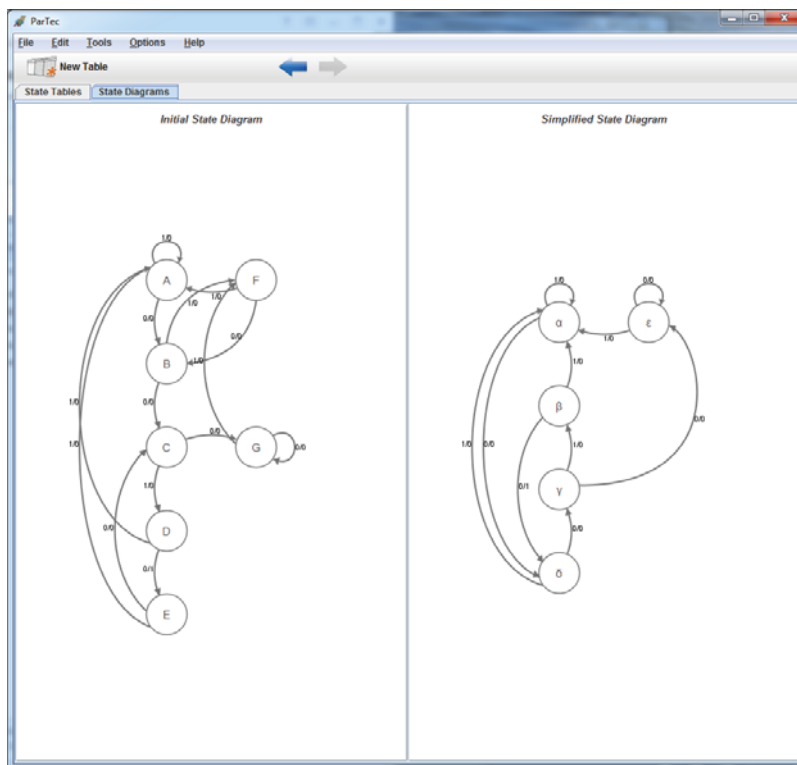

Fig. 4. – State diagram.

3.1 KarnUMa - PC Version

For the PC platform, the application was developed in Java, using Release 6. The choice fell on Java because it offers versatility, efficiency and portability. To run KarnUMa a Java virtual machine needs to be installed in the PC.

The use of KarnUMa starts by choosing the new map option, as can be seen in figure 5. In figure 6 the two options to fill the Karnaugh map are represented: fill the Karnaugh map directly or use a truth table.
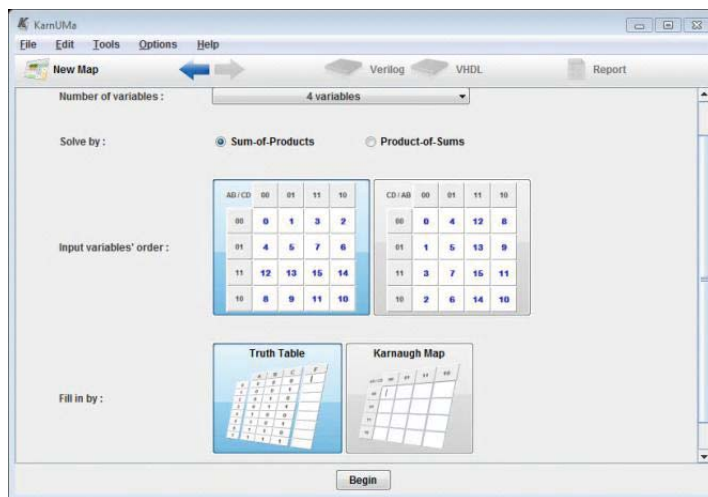
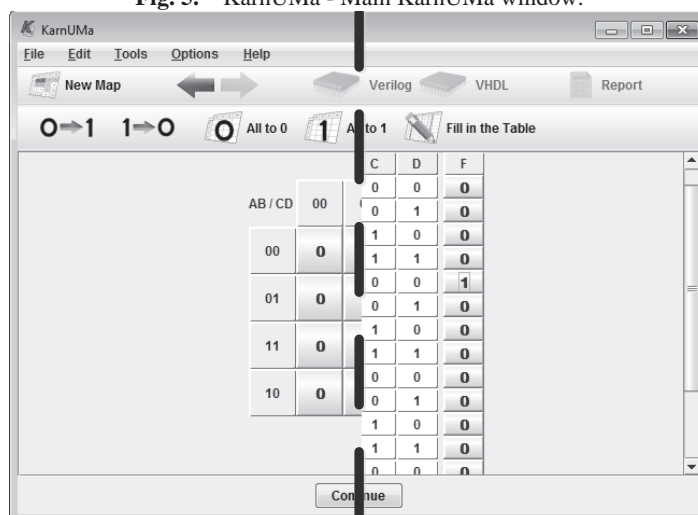**Fig. 5.** – KarnUMa - Main KarnUMa window.

Fig. 6. – KarnUMa - Fill Karnaugh map (right).

Using either option the user can have the map/table filled automatically or do it himself. The use of don't care option is also available and the position of the Karnaugh map can be chosen.

Once an appropriate map is ready, the user can then chose the learning mode and the testing knowledge mode.

For the testing knowledge part there are also two possibilities: "View solution" and "View solution in detail".

Exploring the "View solution" part we can find a set of features that will also be available in the other modes:

- If there is more than one equivalent solution the user can choose between them.

- Sum of Products or Product of Sums can be used by pressing buttons "1" and "0";

- The code for implementing the resulting circuit in Verilog or VHDL can be obtained. – A full report can be obtained that contains the function in truth table, Karnaugh map, the logical equations in several formats including the color image of the implicants and the Hardware Description Language code.

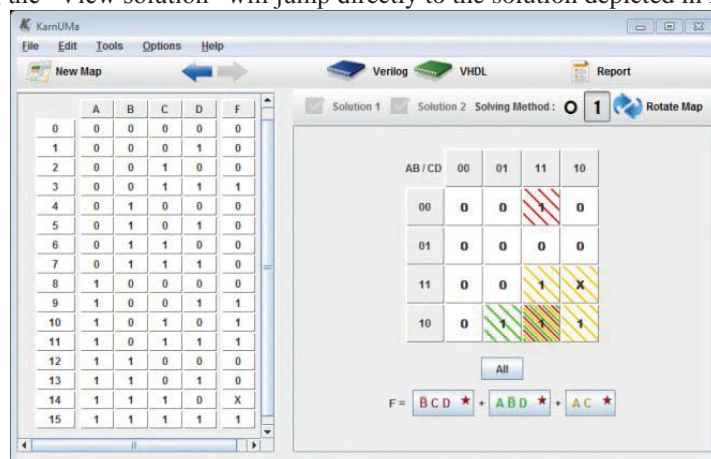Selecting the "View solution" will jump directly to the solution depicted in figure 7.
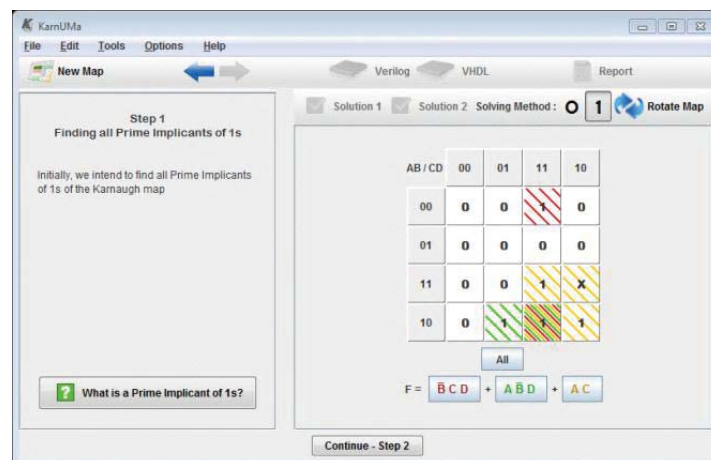

Fig. 7. – KarnUMa - View solution.

Fig. 8. – KarnUMa - View solution in detail.

If "View solution in detail" option is chosen the user is guided through the solution using several steps: step 1 is find all the Implicants; step 2 is identify the Essential Prime Implicants; step 3 is the final solution. Throughout the steps help is available as can be seen in figure 8.

In the learning stage the user can choose to solve the map or solve the map in detail.

The first option allows the user to solve the map in a single step, by selecting the Implicants until the minimal solution is found. The final solution can be validated and the application informs the user if there are errors. To help the user get the minimal solution there's a feedback bar at the bottom of the screen that indicates how many Prime and Essential Prime Implicants are needed to achieve the minimal solution. The bar shows the progress and unlocks the "Check Solution" option.

In the "Solve map in detail" the user is guided in detail through the process of selecting all the Prime Implicants first, then choosing the essential ones and finally find a minimal solution.

These two options are for different stages of learning. The latter is for beginners and steps are more guided while the former is for users who are more familiar with the process.

3.2 Pocket KarnUMa

The Pocket KarnUMa is built on the same base as KarnUMa, but adapted to a mobile platform. Pocket KarnUMa uses the Java MicroEdition (ME). This technology was, when the software was developed, on its 3.0.5 version [8-9]. The choice fell on this technology because "around 80% of all mobile handsets worldwide support the mobile Java standard (Java ME formerly known as J2ME)" [10]. Pocket KarnUMa was also adapted to Android due to the high growth this technology has sustained.

Since Pocket KarnUma is an adaptation of KarnUMa to mobile terminals, it has the features of KarnUMa that could be implemented and that are relevant in this platform since there are visual limitations. Considering that the features are similar to Karnuma, they will be described in less detail.

Figure 9 shows examples of the screens from Pocket KarnUMa and, as can be seen in the main features are similar. The number of variables is limited to 3 or 4 since the screen dimensions do not allow a good visualization of more than four variables in the majority of handset terminals.

Fig. 9. – Pocket KarnUMa -New Map (Left) and Solve map (Right).

Figure 10 shows the Pocket KarnUMa screens for filling the Karnaugh map content. As can be seen, the screens had to be reorganized considering the size available but the application still allows using the truth tables and Karnaugh map to choose the function to be implemented.

The major educational component is not available in Pocket KarnUMa due to viewing size and memory restrictions and only the options of view the solution and solve the map are available.

The View Solution option is very similar to KarnUMa, but the get HDL source code and Karnaugh map's report are not available. These features were not included since they were not important in this version of the application.

As far as the Solve Map option goes, the main difference is that the feedback bar is now placed above the Karnaugh map and due to its small size each bar element's description can only be seen in the Help menu.

Fig.10. – Filling the map's content

In addition to the features not included in this release, the most important difference lies in the different organization of the screen due to its small size. The differences in the organization focused mainly on diverting some options to the menu and also removing some descriptions that now can only be viewed in the Help menu.

Both applications are freely available for download at: http://www.cee.uma.pt/morgado/aplicacoes_sd/index.php/en/

4 Conclusion

This paper presents three applications that can help freshman students to learn digital systems.

The first application is ParTec, designed for learning the partition technique for removing redundant states in finite state machines. This is usually a tedious and repetitive task that the students do not like to perform. As far as the authors could verify, no other tool has been proposed in the scientific community for this purpose.

Regarding KarnUMa applications (PC and Pocket version) unlike another existing applications a strong emphasis is placed in the Essential Prime Implicants identifying them unequivocally. These Implicants are essential to use a deterministic approach to solving Karnaugh maps.

As a teaching tool, KarnUMa provides all the steps leading to a minimal solution of the Karnaugh Map, giving the user the possibility to solve the map or see the solution.

KarnUMa can find up to two map's solutions and verify any alternative solution, by checking if the alternative solution is identical to the two solutions found by the application or equivalent.

KarnUMa, when compared to other applications addressing the same issue, is more complete, is focused in the learning process of the student and as a feedback bar, which allows the user to become aware of his performance in the solving process.

Acknowledgements

References

1. Quine, W. V. (1952). "The Problem of Simplifying Truth Functions". The American Mathematical Monthly 59 (8): 521–531.
2. Quine, W. V. (1955). "A Way to Simplify Truth Functions". The American Mathematical Monthly 62 (9): 627–631.
3. McCluskey, E.J., Jr. (1956). "Minimization of Boolean Functions". Bell System Technical Journal 35 (6): 1417–1444.
4. LiteratePrograms. (2010). Quine-McCluskey algorithm (Java). Retrieved February 22, 2012, from http://en.literateprograms.org/Quine-McCluskey_algorithm_(Java)
5. Freitas, V. and Morgado-Dias, F. (2012^), KarnUMa an Educational Tool for Digital Systems, Controlo'2012, Funchal, Portugal.
6. Reis, A. I., et Ribas, R. P. (2011). Karma, 3.61. (Logic Circuit Synthesis Labs, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Brazil) Retrieved October 2011, from
http://www.inf.ufrgs.br/logics/index.php?option=com_docman&task=doc_download&gid=202&Itemid=58
7. Sena, A., et Torres, M. (2009). EasyKarnaugh 3.0 - Uma ferramenta computacional para o auxílio no ensino de Mapas de Karnaugh em Lógica Digital. Universidade Estadual de Santa Cruz, Departamento de Ciências Exatas e Tecnologias, Brazil.
8. ORACLE. (2012). Java. Retrieved February 16, 2012, from http://www.java.com/pt_BR/about/
9. ORACLE. (2012). JAVA FOR MOBILE DEVICES. Retrieved February 16, 2012, from http://www.oracle.com/technetwork/java/javame/javamobile/overview/getstarted/index.html
10. Balaz, A., et Kromp, J. (2011). Don't Panic Mobile Developer's Guide to the Galaxy (9th ed.). Sogestrass, Breman, Germany: Enough Software.