- **Density** is the ratio between the total number of edges in a business process model and the theoretical maximum number of possible edges regarding the number of nodes. It affects the understandability and modifiability negatively, i.e., lower density values lead to business process models with a lower level of intricacy.
- **Separability** represents the ratio between the number of cut-vertices in a business process model (i.e. nodes that serve as bridges between otherwise strongly-connected components) and the total number of nodes. Separability affects the modifiability negatively, since higher separability implies hard and error-prone modifications of business process models.

The case under study is XCare information system. XCare is a mobile application of 9.9 thousands of lines of code. This application is intended for diabetes patients, which analyzes blood (through an external device) and suggests diet plans. Hence, the *independent variables* of this case study are each business process model obtained from XCare through reverse engineering.

The case study procedure consists of a set of semiautomatic steps that are executed in a computer with a 2.66 GHz dual processor and 4.0 GB RAM. The steps are as follows:

1. First of all, the extraction of business process model from XCare is performed by using MARBLE [25]. MARBLE is a tool used to recover business process models from existing Java code. This tool was selected because is released as an Eclipse plug-in and it therefore can be easily integrated with the IBUPROFEN tool.
2. Fig. 3 gives an example of a business process model obtained by MARBLE from XCare. This business process model contains 255 nodes and 512 edges, being the largest mined from XCare. The smallest model obtained has around 7 nodes and 6 edges. The sample can be visualized entirely and perfectly online [26]. Thus, a sample of 25 business process models is obtained from the source code from XCare.
3. The whole set of IBUPROFEN refactoring algorithms, that was mentioned in Section 3.2, are applied in each business process model retrieved in the above step. Refactoring algorithms are applied in isolation.
4. The dependent variables (size, density and separability) are recorded before and after applying each refactoring algorithm in order to be analyzed later.

Table 1 collects the value of the size, density and separability mean after applying each refactoring algorithm, as well as the gain obtained with respect to the original value. The gain is defined as the ratio between the difference of measure values and the original measure value. Thus, a positive gain means that the refactoring affects the measure positively while a negative gain means that the refactoring affects the measure negatively. A zero gain means that the value for a certain measure did not change after refactoring.

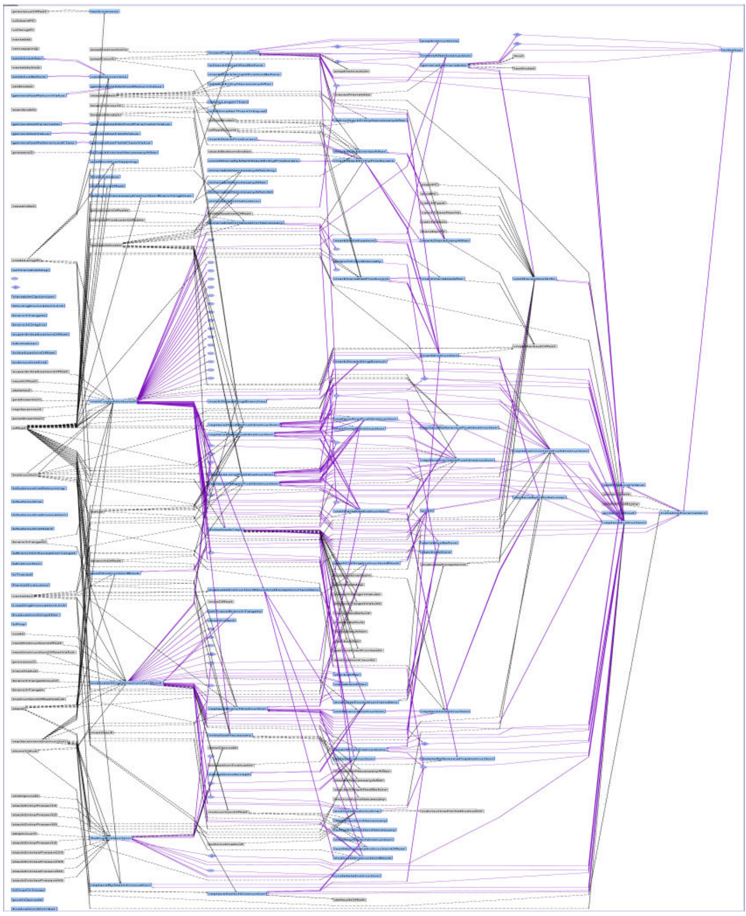

Fig. 3. Example of business process model managed by IBUPROFEN

Table 1. Effect of each refactoring algorithms on the size, density and separability

	Size		Density		Separability	
	Mean	Gain	Mean	Gain	Mean	Gain
Original	70.760	0.000	0.086	0.000	47.88	0.000
R1	46.440	0.366	0.150	-5.903	23.56	0.450
R2	67.120	0.030	0.087	-0.061	44.24	0.040
R3	70.760	0.000	0.086	0.000	47.88	0.000
R4	70.760	0.000	0.086	0.007	47.88	0.000
R5	70.440	0.003	0.086	-0.002	47.88	0.000
R6	62.560	0.114	0.067	0.093	48.76	-0.015
R7	63.040	0.098	0.093	-0.068	41.96	0.120
R8	74.360	-0.201	0.114	-0.511	51.48	-0.249
R9	90.160	-0.229	0.064	0.111	48.08	-0.002
R10	70.760	0.000	0.086	0.000	47.88	0.000

Table 1 reveals that removing isolated nodes decreases the size and separability while the density is increased. Despite the density is higher after R1, the relevance of the model has been increased since non-relevant elements have been removed. Similarly, R2 causes an increase of density when the size is decreased. Separability is decreased slightly. However, R3 has not impact on these measures due to business process models under study do not have nesting gateways. Removing inconsistencies (R4) maintains the same size and separability while the density is decreased because the number of edges is lower. Unnecessary gateways are removed (R5) and therefore the size is decreased while the density is increased owing to the number of nodes is lower. Separability after R5 is exacerbated slightly. R6 creates compound tasks in several business process models. This fact makes the size and density decrease in the most of cases. The same happens with R7, the number of nodes is lower but the number of edges is lower to and therefore, the density may increase while separability increases. R8 adds new missing elements in the model as start and end event as well as complex gateways. This makes that the size, separability and density are higher. In the same way, adding gateways in incoming and outgoing branches causes higher size. Nevertheless, the density after R9 tends to decrease due to there is more nodes in the model. Separability increases slightly since elements are more connected. In contrast, R10 does not have affect in any measures but the refinement of names implies an enhancement of the understandability.

5 CONCLUSIONS AND FUTURE WORK

Refactoring techniques has proved to be a good choice for improving business process models in terms, for example, of their understandability and modifiability levels. While graph-based algorithms have been successfully employed in different contexts, most business process model refactoring techniques often use alternative data structures [10-12] which leads to inefficient results. For this reason, this paper presents IBUPROFEN as a technique for refactoring business process models following a graph-based approach. Thus, the business process model is managed by means of a graph, changing its internal structure while its semantic is preserved. IBUPROFEN proposes ten refactoring algorithm divided into three groups in order to address the common problems that organizations have to deal when they retrieved such business process models by reverse engineering.

In order to demonstrate the feasibility of this approach, IBUPROFEN has been firstly implemented as an open source tool, and secondly, a case study with industrial business process models has been conducted. The case study reveals that the application the proposed graph-based refactoring algorithms improve the size, separability and density of business process models in the most of cases by removing non-relevant and fine-grained elements as well as by completing the models. The main limitation of this study is that results show size and density have an inverse relationship, i.e., when one increase the other decrease.

The second limitation lies in the empirical study analyzes the application of each refactoring algorithm in isolation. However, studies reveals that the order of application of various refactoring algorithms in sequence could have an effect on the obtained results [18].

In line with the mentioned limitations, the future work will focus on the replication of the case study by analyzing alternative measures as well as the effect of different application orders. Furthermore, an algorithm improvement endeavor will be made to conciliate the size and density gain at the same time.

Acknowledgments

This work was supported by the FPU Spanish Program and the R&D projects MAGO /PEGASO (Ministerio de Ciencia e Innovación [TIN2009-13718-C02-01]) and GEODAS-BC (Ministerio de Economía y Competitividad & Fondos FEDER [TIN2012-37493-C03-01]).

6 REFERENCES

1. Weske, M., *Business Process Management: Concepts, Languages, Architectures*2007, Leipzig, Germany: Springer-Verlag Berlin Heidelberg. 368.
2. OMG. *Business Process Modeling Notation Specification 2.0*. 2011; Available from: http://www.omg.org/spec/BPMN/2.0/PDF/.
3. Jeston, J., J. Nelis, and T. Davenport, *Business Process Management: Practical Guidelines to Successful Implementations*. 2nd ed2008, NV, USA: Butterworth-Heinemann (Elsevier Ltd.). 469.
4. Davenport, T.H., *Need radical innovation and continuous improvement? Integrate process reengineering and TQM.* Strategy & Leadership Journal, 1993. **21**(3): p. 6-12.
5. Heuvel, W.-J.v.d., *Aligning Modern Business Processes and Legacy Systems: A Component-Based Perspective (Cooperative Information Systems)*2006: The MIT Press.
6. van der Aalst, W., *Process Mining: Overview and Opportunities.* ACM Transactions on Management Information Systems (TMIS), 2012. **3**(2): p. 7.
7. Pérez-Castillo, R., I. García-Rodríguez de Guzmán, and M. Piattini, *Business Process Archeology using MARBLE.* Information and Software Technology, 2011.
8. Fahland, D. and W.M.P.v.d. Aalst, *Repairing Process Models to Reflect Reality.* 2012.
9. Fernández-Ropero, M., R. Pérez-Castillo, and M. Piattini, *Refactoring Business Process Models: A Systematic Review*, in *7th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)*, J. Filipe and L. Maciaszek, Editors. 2012, SciTePress: Wrocław, Poland. p. 140-145.
10. Dijkman, R., et al., *Identifying refactoring opportunities in process model repositories.* Inf. Softw. Technol., 2011. **53**(9): p. 937-948.
11. Weber, B., et al., *Refactoring large process model repositories.* Computers in Industry, 2011. **62**(5): p. 467-486.
12. La Rosa, M., et al., *Managing process model complexity via abstract syntax modifications.* Industrial Informatics, IEEE Transactions on, 2011. **7**(4): p. 614-629.
13. Pham, M.-D., P. Boncz, and O. Erling, *S3G2: A Scalable Structure-Correlated Social Graph Generator*, in *Selected Topics in Performance Evaluation and Benchmarking*, R. Nambiar and M. Poess, Editors. 2013, Springer Berlin Heidelberg. p. 156-172.

14. Gubichev, A. and T. Neumann, *Fast approximation of steiner trees in large graphs*, in *Proceedings of the 21st ACM international conference on Information and knowledge management*2012, ACM: Maui, Hawaii, USA. p. 1497-1501.
15. Mens, T., G. Taentzer, and O. Runge, *Analysing refactoring dependencies using graph transformation*. Software & Systems Modeling, 2007. **6**(3): p. 269-285.
16. Bobrik, R., M. Reichert, and T. Bauer, *View-Based Process Visualization*, in *Business Process Management*, G. Alonso, P. Dadam, and M. Rosemann, Editors. 2007, Springer Berlin Heidelberg. p. 88-95.
17. Hauser, R. and J. Koehler, *Compiling Process Graphs into Executable Code*, in *Generative Programming and Component Engineering*, G. Karsai and E. Visser, Editors. 2004, Springer Berlin Heidelberg. p. 317-336.
18. Fernández-Ropero, M., et al., *Assessing the Best-Order for Business Process Model Refactoring*, in *28th Symposium On Applied Computing (SAC)*2013: Coimbra, Portugal. p. 1400-1406.
19. Fernández-Ropero, M., R. Pérez-Castillo, and M. Piattini. *IBUPROFEN*. 2012; Available from: http://marketplace.eclipse.org/content/IBUPROFEN.
20. Naveh, B. and Contributors. *JGraphT*. 2011; Available from: http://jgrapht.org/.
21. Hunter, J., *JDOM*, 2009.
22. Pérez-Castillo, R., et al., *Generating Event Logs from Non-Process-Aware Systems Enabling Business Process Mining*. Enterprise Information System Journal, 2011. **5**(3): p. 301–335.
23. Zou, Y. and M. Hung, *An Approach for Extracting Workflows from E-Commerce Applications*, in *Proceedings of the Fourteenth International Conference on Program Comprehension*2006, IEEE Computer Society. p. 127-136.
24. Mendling, J., H.A. Reijers, and W.M.P. van der Aalst, *Seven process modeling guidelines (7PMG)*. Information and Software Technology, 2010. **52**(2): p. 127-136.
25. Pérez-Castillo, R., et al., *MARBLE. A Business Process Archeology Tool*, in *27th IEEE International Conference on Software Maintenance (ICSM'11)*2011, IEEE Computer Society: Williamsburg, Virginia, USA. p. 578-581.
26. Fernández-Ropero, M., R. Pérez-Castillo, and M. Piattini. *Extra material of Graph-Based Business Process Model Refactoring* 2013; Available from: http://alarcos.esi.uclm.es/per/mfernandez/material3.html.