Author: Fernandez, Saez A.M.
Title: Studying the benefits of using UML on software maintenance : an evidence-based approach
Issue Date: 2018-11-15

APPENDIX A. LIST OF PRIMARY STUDIES (related to Chapter 2)

The papers considered as primary studies in the systematic mapping study presented in this paper and that have been treated as primary studies are presented below.

P1. Abrahao, S., Insfran, E., Gravino, C., and Scanniello, G. (2009). On the effectiveness of dynamic modeling in UML: Results from an external replication. In the Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM'09), pp. 468-472.

P2. Gemino, A., and Parker, D. (2009). Use case diagrams in support of use case modeling: Deriving understanding from the picture. Journal of Database Management, 20(1), 1-24.

P3. Manso, M. E., Genero, M., and Piattini, M. (2003). No-redundant metrics for UML class diagram structural complexity. In the Proceedings of the 15th International Conference on Advanced Information Systems Engineering (CAiSE'08) (LNCS 2681 pp. 127-142).

P4. Otero, M. C., and Dolado, J. J. (2004). Evaluation of the comprehension of the dynamic modeling in UML. Information and Software Technology, 46(1), 35-53.

P5. Arisholm, E., Briand, L. C., Hove, S. E., and Labiche, Y. (2006). The impact of UML documentation on software maintenance: An experimental evaluation. IEEE Transactions on Software Engineering, 32(6), 365-381.

P6. Burd, E., Overy, D., and Wheetman, A. (2002). Evaluating using animation to improve understanding of sequence diagrams. In the Proceedings of the 10th International Workshop on Program Comprehension (IWPC'02), pp. 107-107.

P7. Cruz-Lemus, J. A., Genero, M., and Piattini, M. (2008). Using controlled experiments for validating UML statechart diagrams measures. In the Proceedings of the International Workshop in Software Measurement and International Conference on Software Process and Product Measurement (IWSM/Mensura'07) (LNCS 4895 pp. 129-138).

P8. Dzidek, W. J., Arisholm, E., and Briand, L. C. (2008). A realistic empirical evaluation of the costs and benefits of UML in software maintenance. IEEE Transactions on Software Engineering, 34(3), 407-432.

P9. Eichelberger, H., & Schmid, K. (2009). Guidelines on the aesthetic quality of UML class diagrams. Information and Software Technology, 51(12), 1686-1698.

P10. Genero, M., Cruz-Lemus, J. A., Caivano, D., Abrahão, S., Insfran, E., and Carsí, J. A. (2008). Assessing the influence of stereotypes on the comprehension of UML sequence diagrams: A controlled experiment. In the

Proceedings of the 11th international conference on Model Driven Engineering Languages and Systems (MoDELS'09) (LNCS 5301 pp. 280-294).

P11. Genero, M., Moody, D. L., and Piattini, M. (2005). Assessing the capability of internal metrics as early indicators of maintenance effort through experimentation. Journal of Software Maintenance and Evolution: Research and Practice, 17(3), 225-246.

P12. Genero, M., Piattini, M., and Calero, C. (2002). Empirical validation of class diagram metrics. In the Proceedings of the 2002 International Symposium on Empirical Software Engineering (ISESE'02), pp. 195-203.

P13. Genero, M., Piattini, M., Manso, E., and Cantone, G. (2003). Building UML class diagram maintainability prediction models based on early metrics. In the Proceedings of the 9th International Symposium on Software Metrics (METRICS'03), pp. 263-263.

P14. Glezer, C., Last, M., Nachmany, E., and Shoval, P. (2005). Quality and comprehension of UML interaction diagrams-an experimental comparison. Information and Software Technology, 47(10), 675-692.

P15. Irani, P., and Ware, C. (2004). The effect of a perceptual syntax on the learnability of novel concepts. In the Proceedings of The Eighth International Conference on Information Visualisation (IV'04), pp. 308-314.

P16. Lange, C. F., and Chaudron, M. R. V. (2006). Effects of defects in UML models: An experimental investigation. In the Proceedings of The 28th International Conference on Software Engineering (ICSE'06), pp. 401-411.

P17. Lange, C. F., Wijns, M. A., and Chaudron, M. R. V. (2007). A visualization framework for task-oriented modeling using UML. In the Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS'07), pp. 289a-289a.

P18. Lange, C. F. J., Wijns, M. A. M., and Chaudron, M. R. V. (2007). Supporting task-oriented modeling using interactive UML views. Journal of Visual Languages and Computing, 18(4), 399-419.

P19. Manso, M. E., Cruz-Lemus, J. A., Genero, M., and Piattini, M. (2009). Empirical validation of measures for UML class diagrams: A meta-analysis study. In the Proceedings of the International Conference on Model Driven Engineering Languages and Systems (MODELS'09) (LNCS 5421 pp. 303-313).

P20. Nugroho, A. (2009). Level of detail in UML models and its impact on model comprehension: A controlled experiment. Information and Software Technology, 51(12), 1670-1685.

P21. Otero, M. C., and Dolado, J. J. (2002). An initial experimental assessment of the dynamic modelling in UML. Empirical Software Engineering, 7(1), 27-47.

P22. Otero, M. C., and Dolado, J. J. (2005). An empirical comparison of the dynamic modeling in OML and UML. Journal of Systems and Software, 77(2), 91-102.

P23. Razali, R., Snook, C. F., and Poppleton, M. R. (2007). Comprehensibility of UML-based formal model: a series of controlled experiments. In the Proceedings of the 1st ACM International Workshop on Empirical Assessment of Software Engineering Languages and Technologies (WEASEL'07), pp. 25-30.

P24. Reinhartz-Berger, I., and Dori, D. (2005). OPM vs. UML - Experimenting with comprehension and construction of web application models. Empirical Software Engineering, 10(1), 57-79.

P25. Ricca, F., Di Penta, M., Torchiano, M., Tonella, P., and Ceccato, M. (2006). An empirical study on the usefulness of Conallen's stereotypes in Web application comprehension. In the Proceedings of the Eighth IEEE International Symposium on Web Site Evolution (WSE'06), pp. 58-68.

P26. Ricca, F., Penta, M. D., Torchiano, M., Tonella, P., and Ceccato, M. (2010). How developers' experience and ability influence web application comprehension tasks supported by UML stereotypes: A series of four experiments. IEEE Transactions on Software Engineering, 36(1), 96-118.

P27. Riva, C., Selonen, P., Systa, T., and Xu, J. (2004). UML-based reverse engineering and model analysis approaches for software architecture maintenance. In the Proceedings of the 20th IEEE International Conference on Software Maintenance (ICSM'04), pp. 50-59.

P28. Sheldon, F. T., and Chung, H. (2006). Measuring the complexity of class diagrams in reverse engineering: Research articles. Journal of Software Maintenance and Evolution: Research and Practice, 18(5), 333-350.

P29. Staron, M., Kuzniarz, L., and Wohlin, C. (2006). Empirical assessment of using stereotypes to improve comprehension of UML models: A set of experiments. Journal of Systems and Software, 79(5), 727-742.

P30. Tilley, S., and Huang, S. (2003). A qualitative assessment of the efficacy of UML diagrams as a form of graphical documentation in aiding program understanding. In the Proceedings of the 21st Annual International Conference on Documentation (SIGDOC'03), pp. 184-191.

P31. Xie, S., Kraemer, E., and Stirewalt, R. E. K. (2007). Empirical evaluation of a UML sequence diagram with adornments to support understanding of thread interactions. In the Proceedings of the 15th IEEE International Conference on Program Comprehension (ICPC'07), pp. 123-134.

P32. Genero, M., Piattini, M., Abrahao, S., Insfran, E., Carsi, J. A., and Ramos, I. (2007). A controlled experiment for selecting transformations based on quality attributes in the context of MDA. In the Proceedings of the First International Symposium Empirical Software Engineering and Measurement (ESEM'07), pp. 498-498.

P33. Gross, A., and Doerr, J. (2009). EPC vs. UML activity diagram - Two experiments examining their usefulness for requirements engineering. In the Proceedings of the 2009 17th IEEE International Requirements Engineering Conference (RE'09), pp. 47-56.

P34. Purchase, H. C., Colpoys, L., McGill, M., and Carrington, D. (2002). UML collaboration diagram syntax: An empirical study of comprehension. In the Proceedings of the First International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT'02), pp. 13-22.

P35. Settimi, R., Cleland-Huang, J., Khadra, O. B., Mody, J., Lukasik, W., and DePalma, C. (2004). Supporting software evolution through dynamically retrieving traces to UML artifacts. In the Proceedings of the 7th International Workshop on Principles of Software Evolution (IWPSE'04), pp. 268-272.

P36. Sharif, B., and Maletic, J. I. (2009a). The effect of layout on the comprehension of UML class diagrams: A controlled experiment. In the Proceedings of the 5th IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT'09), pp. 11-18.

P37. Sharif, B., and Maletic, J. I. (2009b). An empirical study on the comprehension of stereotyped UML class diagram layouts. In the Proceedings of the IEEE International Conference on Program Comprehension (ICPC'09), pp. 268-272.

P38. Swan, J., Barker, T., Britton, C., and Kutar, M. (2005). An empirical study of factors that affect user performance when using UML interaction diagrams. In the Proceedings of the International Symposium on Empirical Software Engineering (ISESE'05), pp. 10-10.

APPENDIX B. DEFINITIONS OF MEASURES (related to Chapter 2)

The definition of the measures for the dependent variables used in the empirical studies covered in this systematic mapping study is presented below. In the definition of the measures we use the word *question* to simplify the definition, but we can also refer to a *task*. Note that in Table 7, all the studies that use measures which measure the same concept are grouped together (even though they were originally presented with different names), using the names according to the classification set out below.

Correctness

Definition: The percentage of questions that are answered correctly.

Formula: Number of correct answers/Number of questions.

Papers which use this measure: [P9], [P7], [P8], [P9], [P18], [P20], [P10]

Accuracy

Definition: the number of correct answers.

Papers which use this measure with this name: [P23], [P34], [P36], [P37]

Papers which use this measure with different names:

- Total score: [P4], [P21], [P24], [P29]
- Correct interpretation: [P6]
- Number of responses: [P22]
- Comprehension: [P2]
- Without a specific name: [P14], [P17], [P31]

Effectiveness

Definition: The percentage of questions answered which are correct.

Formula: Number of correct answers/Number of answers

Papers which use this measure with this name: [P32]

Papers which use this measure with a different name:

- Correctness: [P7], [P13], [P19]

F-Measure

Definition: It is an aggregate measure which is a standard combination of the *recall* and *precision*, defined as their harmonic mean.

Formula: $F - Measure = \frac{2 \cdot precision_{s,i} \cdot recall_{s,i}}{precision_{s,i} + recall_{s,i}}$

Recall

Definition: It measures the fraction of expected items that are in the answer.

Formula: $\frac{|A_{s,i} \cap C_i|}{|C_i|}$

$A_{s,i}$: Set of elements mentioned in the answer to question i by subject s.

C_i: The correct set of elements expected for question i.

Precision

Definition: It measures the fraction of items in the answer that are correct.

Formula: $\frac{|A_{s,i} \cap C_i|}{|A_{s,i}|}$

$A_{s,i}$: Set of elements mentioned in the answer to question i by subject s.

C_i: The correct set of elements expected for question i.

Papers which use this measure: [P1], [P25], [P26]

Efficiency

Definition: The number of correct answers per time units.

Formula: Number of correct answers/Time

Papers which use this measure with this name: [P7], [P32], [P10]

Papers which use this measure with a different name:

- Efficacy: [P17]

Relative time (for a correct answer)

Definition: It measures the time that a subject took to obtain a correct answer.

Formula: Time/Number of correct answers

Papers which use this measure: [P29], [P7]

Perceived comprehensibility

This is a subjective measure obtained as a ranking of the subject´s perceived understandability of a certain diagram. Measured using a 1-5 Likert ordinal scale, where the score of 1 indicated that the diagram was absolutely incomprehensible.

Papers which use this measure: [P3], [P12]

Perceived ease of construction

This is a subjective measure obtained as a ranking of the subject´s perceived ease of construction of a certain diagram. It is measured using a 1-5 Likert ordinal scale, where the score of 1 indicates that the diagram is very difficult.

Papers which use this measure: [P14]

Time

This is a measure which is used to calculate the number of units of time used to perform a task.

Papers which use this measure: [P3], [P4], [P5], [P7], [P8], [P9], [P11], [P12], [P13], [P14], [P17], [P18], [P19], [P21], [P22], [P23], [P28], [P29], [P32], [P32], [P36], [P37], [P38]

Errors

This is a measure which counts the number of mistakes made in solving a specific task.

Papers which use this measure: [P9], [P8], [P9], [P15], [P16], [P27], [P28], [P33]

APPENDIX C. THE SEARCH STRINGS (related to Chapter 2)

The definition of the search strings used in each search engine is presented as follows. As commented on in Table 2.1 we had three major terms, and we also considered alternative spellings and synonyms of, or terms related to, the major terms. The original search string was:

(UML OR (Unified Modelling Language))

AND

(Maintenance OR Maintainability OR Modularity OR Reusability OR Analyzability OR Changeability OR Evolution OR Evolvability OR Modification OR Stability OR Testability OR Comprehensibility OR Comprehension OR Understandability OR Understanding OR Misinterpretation)

AND

(Empirical OR Experiment OR Survey OR Case study OR Action research)

Owing to the limitation of the search engines, we observed that such a long string could not be used directly in all the search engines. It was therefore necessary to tailor the search string to each digital library by splitting the original search string and then combining the results manually. The search strings used for each digital source are presented below.

ACM and IEEE search string

(Abstract:UML OR (Abstract:Unified AND Abstract:Modeling AND Abstract:Language)) AND (Abstract:Maintenance OR Abstract:maintainability OR Abstract:modularity OR Abstract:reusability OR Abstract:analyzability OR Abstract:changeability OR Abstract:evolution OR Abstract:evolvability OR(Abstract:modification AND Abstract:stability) OR Abstract:testability OR Abstract:comprehensibility OR Abstract:comprehension OR Abstract:understandability OR Abstract:understanding) AND (Abstract:empirical OR Abstract:experiment OR Abstract:survey OR (Abstract:case AND Abstract:study) OR (Abstract:action AND Abstract:research))

Science Direct and SCOPUS search string

TITLE-ABSTR-KEY((UML OR (Unified AND Modeling AND Language)) AND (Maintenance OR maintainability OR modularity OR reusability OR analyzability OR changeability OR evolution OR evolvability OR(modification AND stability) OR testability OR comprehensibility OR comprehension OR understandability OR understanding) AND (empirical OR experiment OR survey OR (case AND study) OR (action AND research)))

Springerlink search string

The search string was divided into 28 search strings because this string only allows 10 terms to be placed in the search string textbox. After the searches had been carried out, we combined their results using the SLR-Tool, which automatically detects duplicate papers.

String 1: ab:(UML and maintenance and(empirical or experiment or survey or(case and study)or(action and research)))

String 2: ab:(UML and maintainability and(empirical or experiment or survey or(case and study)or(action and research)))

String 3: ab:(UML and modularity and(empirical or experiment or survey or(case and study)or(action and research)))

String 4: ab:(UML and reusability and(empirical or experiment or survey or(case and study)or(action and research)))

String 5: ab:(UML and analyzability and(empirical or experiment or survey or(case and study)or(action and research)))

String 6: ab:(UML and changeability and(empirical or experiment or survey or(case and study)or(action and research)))

String 7: ab:(UML and evolution and(empirical or experiment or survey or(case and study)or(action and research)))

String 8: ab:(UML and evolvability and(empirical or experiment or survey or(case and study)or(action and research)))

String 9: ab:(UML and(modification and stability)and(empirical or experiment or survey or(case and study)or(action and research)))

String 10: ab:(UML and testability and(empirical or experiment or survey or(case and study)or(action and research)))

String 11: ab:(UML and comprehensibility and(empirical or experiment or survey or(case and study)or(action and research)))

String 12: ab:(UML and comprehension and(empirical or experiment or survey or(case and study)or(action and research)))

String 13: ab:(UML and understandability and(empirical or experiment or survey or(case and study)or(action and research)))

String 14: ab:(UML and understanding and(empirical or experiment or survey or(case and study)or(action and research)))

String 15: ab:("Unified Modeling Language" and Maintenance and(empirical or experiment or survey or(case and study)or(action and research)))

String 16: ab:("Unified Modeling Language" and maintainability and(empirical or experiment or survey or(case and study)or(action and research)))

String 17: ab:("Unified Modeling Language" and modularity and(empirical or experiment or survey or(case and study)or(action and research)))

String 18: ab:("Unified Modeling Language" and reusability and(empirical or experiment or survey or(case and study)or(action and research)))

String 19: ab:("Unified Modeling Language" and analyzability and(empirical or experiment or survey or(case and study)or(action and research)))

String 20: ab:("Unified Modeling Language" and changeability and(empirical or experiment or survey or(case and study)or(action and research)))

String 21: ab:("Unified Modeling Language" and evolution and(empirical or experiment or survey or(case and study)or(action and research)))

String 22: ab:("Unified Modeling Language" and evolvability and(empirical or experiment or survey or(case and study)or(action and research)))

String 23: ab:("Unified Modeling Language" and(modification and stability)and(empirical or experiment or survey or(case and study)or(action and research)))

String 24: ab:("Unified Modeling Language" and testability and(empirical or experiment or survey or(case and study)or(action and research)))

String 25: ab:("Unified Modeling Language" and comprehensibility and(empirical or experiment or survey or(case and study)or(action and research)))

String 26: ab:("Unified Modeling Language" and comprehension and(empirical or experiment or survey or(case and study)or(action and research)))

String 27: ab:("Unified Modeling Language" and understandability and(empirical or experiment or survey or(case and study)or(action and research)))

String 28: ab:("Unified Modeling Language" and understanding and(empirical or experiment or survey or(case and study)or(action and research)))

Wiley Inter Science search string

We used the advanced search in which it is possible to use three (or more) textboxes to enter complex strings (see Figure 1). We used a search string divided into three parts, which were linked by AND connectors. Different textboxes were used to introduce each part of the search string:

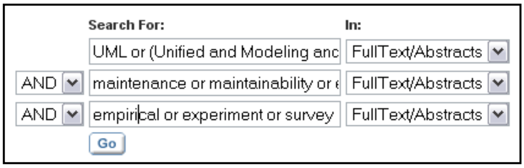

Figure 1. Wiley advanced search.

UML or (Unified and Modeling and Language)

AND

Maintenance OR maintainability OR modularity OR reusability OR analyzability OR changeability OR evolution OR evolvability OR (modification AND stability) OR testability OR comprehensibility OR comprehension OR understandability OR understanding

AND

empirical OR experiment OR survey OR (case AND study) OR (action AND research)

APPENDIX D. EXAMPLES OF LOW AND HIGH LOD DIAGRAMS (related to Chapter 3)

This appendix shows examples of a low LoD class diagram (Figure D.1) and a high LoD class diagram (Figure D.2).

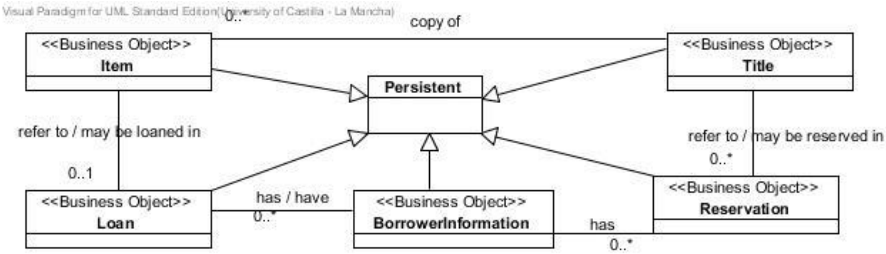

Figure D.1. Example of a low LoD class diagram.

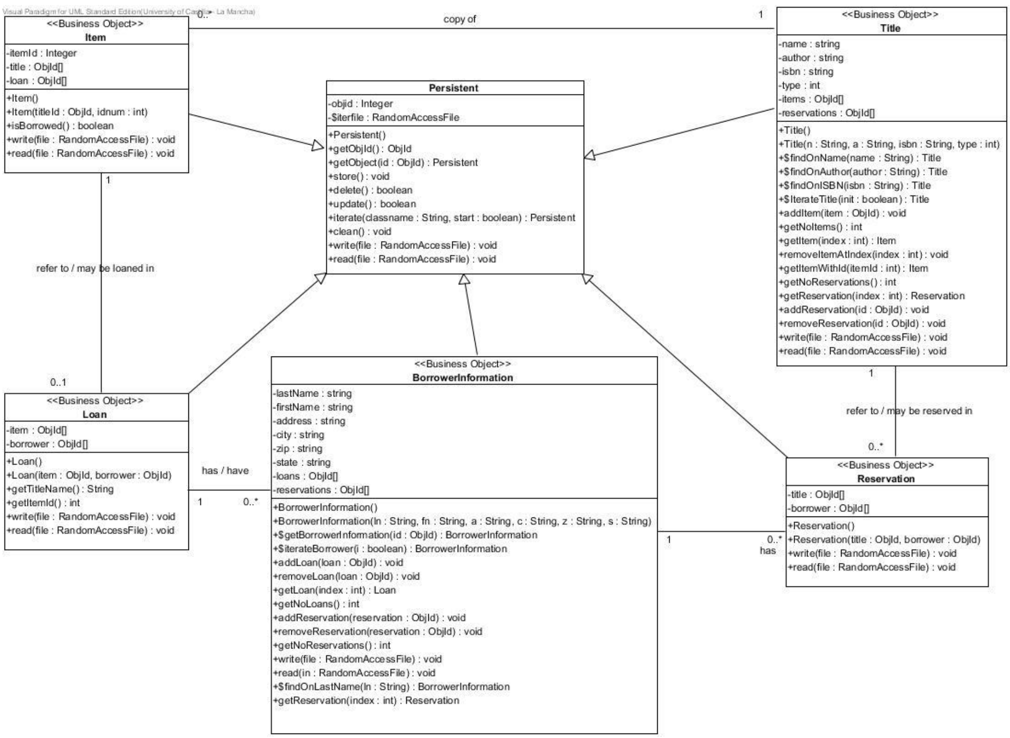

Figure D.2. Example of a high LoD class diagram.

APPENDIX E. INTERVIEW QUESTIONNAIRE (related to Chapter 6)

The following lines present the questionnaire used to carry out the interviews. The questionnaire is divided into 3 blocks:

Common questions for all the interviewees

1. What is your background and your experience?
2. What is your role, and what are your responsibilities within the project?
3. Which kind of documentation do you use to perform maintenance tasks: diagrams, code, textual information, etc.?
4. How do you use documentation/diagrams?
5. How often do you use the documentation?

Block of questions for those interviewees who use UML diagrams

6. Why do you use UML diagrams? (Give reasons) / For what purpose is UML modelling used?
7. For maintenance, do you manage (look up/ create/ modify) diagrams in a modelling tool (i.e. Enterprise Architect, Visio, etc.)? Or do you look them up in the documentation (i.e. word documents, pictures, etc.)? Did you receive any training about the tool?
8. Which diagrams do you consider to be most frequently used to perform the maintenance tasks? Which diagrams do you consider to be the most useful for performing the maintenance tasks?
9. Do diagrams help in solving defects?

   IF the answer is YES

   9.1. How do they do so?
10. When you maintain the code, do you also maintain the diagrams?

    IF the answer is YES

    10.1. How much time does it take?

    10.2. Who maintains the diagrams? (The same person who maintains the code or a different one?)

    IF the answer is NO

    10.3. Why do you not maintain the diagrams? Are the diagrams correct but not the code? Or is there another reason?
11. Do you like UML?
12. Do you think using UML has an impact on the time of the project? Do you think using UML has an impact on the quality of the final product? How?
13. What cost factors are related to using UML modelling in your work (training, tooling, etc.)?
14. Do you think there is another way in which to improve your work other than UML (i.e. another kind of diagram, etc.)?

15. Did you receive any training about UML at the Company? And before coming to the Company?
16. Do you think that the use of modelling allows errors to become incorporated?
17. Where does the diagram originate from and go to? (chain of use)
18. Do you reuse documentation from previous projects?

Block of questions for those interviewees who do not use UML diagrams

19. Do you use any kind of diagram to maintain the system and to communicate between team members?
20. Would you like UML diagrams to be available?

If the answer is YES

20.1. How do you think UML would help you to maintain the system?
20.2. What benefits do you think UML diagrams could contribute to your work?
20.3. Do you think UML helps to improve the quality of the final product? How?
20.4. What cost factors are related to using UML modelling in a project?
20.5. Do you think the size of the system influences the way in which UML is used (or not used) on a project?
20.6. Do you think the size of the team influences the way in which UML is used (or not used) on a project?

APPENDIX F. BACKGROUND INFORMATION RELATED TO EACH INTERVIEWEE (related to Chapter 6)

Table F.1 summarizes the main background information related to each interviewee of the case study.

Interviewee	ICT experience	Context	Educational field	Educational level	Gender	Role
[Int1]	very high	common project	n.a.	school	male	project architect
[Int2]	medium	n.a.	computer sciences	master's degree	male	project manager
[Int3]	very high	n.a.	electronics and mathematics	bachelor's degree	male	project architect
[Int4]	n.a.	n.a.	computer sciences	bachelor's degree	male	project architect
[Int5]	medium	n.a.	computer sciences	bachelor's degree	male	information analyst
[Int6]	low	n.a.	n.a.	n.a.	male	technical lead
[Int8]	very high	n.a.	navy	n.a.	male	test engineer
[Int9]	high	outsourcing	n.a.	n.a.	male	delivery lead
[Int10]	very high	Embedded real-time programming	n.a.	n.a.	male	programmer / application developer
[Int11]	very high	n.a.	computer sciences	bachelor's degree	female	programmer / application developer
[Int12]	low	migration	art	high school	male	test coordinator
[Int13]	very high	n.a.	n.a.	school	male	technical lead
[Int14]	high	n.a.	computer sciences	n.a.	male	information analyst
[Int16]	n.a.	web/mobile projects (SCRUM)	electronics	n.a.	male	SCRUM master
[Int18]	very high	n.a.	chemistry and physics	bachelor's degree	male	system analyst
[Int19]	high	common project	computer sciences	master's degree	female	programmer / application developer

Table F.1. Background information of interviewees of the case study.

Interviewee	ICT experience	Context	Educational field	Educational level	Gender	Role
[Int20]	very high	n.a.	Business and Finances	bachelor's degree	male	programmer / application developer
[Int21]	n.a.	n.a.	computer sciences	master's degree	male	analyst developer
[Int23]	very high	n.a.	n.a.	high school	female	analyst developer
[Int24]	very high	web/mobile projects (SCRUM)	n.a.	n.a.	male	project architect
[Int25]	very high	n.a.	n.a.	bachelor's degree	male	programmer / application developer
[Int26]	very high	common project	computer sciences	master's degree	male	project architect
[Int27]	very high	mainframe	n.a.	HBO	male	programmer / application developer
[Int28]	very high	old legacy system	psychology	HBO	male	programmer / application developer
[Int29]	very high	n.a.	n.a.	n.a.	male	team leader
[Int31]	high	common project	computer sciences	bachelor's degree	male	deployer
[Int32]	very high	common project	computer sciences	HBO	male	programmer / application developer
[Int33]	very high	web and mobile projects (SCRUM)	n.a.	n.a.	male	programmer / application developer
[Int35]	very high	change from mainframe to agile	n.a.	n.a.	male	information analyst
[Int36]	very high	old legacy system	computer sciences	n.a.	male	program analyst
[Int37]	high	outsourcing	n.a.	n.a.	male	project manager

Table F.1. Background information of interviewees of the case study.