Investigating techniques from the 2000's for class model extraction

Marianne Huchard, Ines Ammar, Ahmad Bedja-Boana, Jessie Carbonnel, Theo Chartier, Franz Fallavier, Julie Ly, Daniel Alias Nguyen Vu-Hao, Florian Pinier, Ralf Saenen and Sébastien Villon

Université Montpellier 2 - LIRMM

July 9, 2014

Industrial context

Request of a major (anonymous) IT service company

Design Low-cost migration of a legacy software suite composed of:

- man-machine interfaces (HTML, VBScript/ASP, Javascript)
- several databases, SQL procedures (SQL Server 2000)
- procedural source code (VB6)

Low-cost (money is invested in new developments)

- less effort than fully manual migration
- automatize as far as possible
- open-source, free, tools

Teaching context

Research and development project in Master course

- each student: 1 man/month
- distributed during 5 months (other classes and projects in parallel).
- read research papers (at least one per student)
- project managements activities: Gantt diagram, role/task distribution, meeting management
- reproduce solutions of papers

10 students

- 3 groups
- one common meeting every week (half of the meetings with IT service company partner), and other meetings inside the groups

Project organization

Main tasks

- Reducing migration to class model extraction and to 2 software systems from the suite
- Designing a migration chain
- Choosing relevant research papers about class model extraction
- Implement the found extraction heuristics
- Apply to the software systems

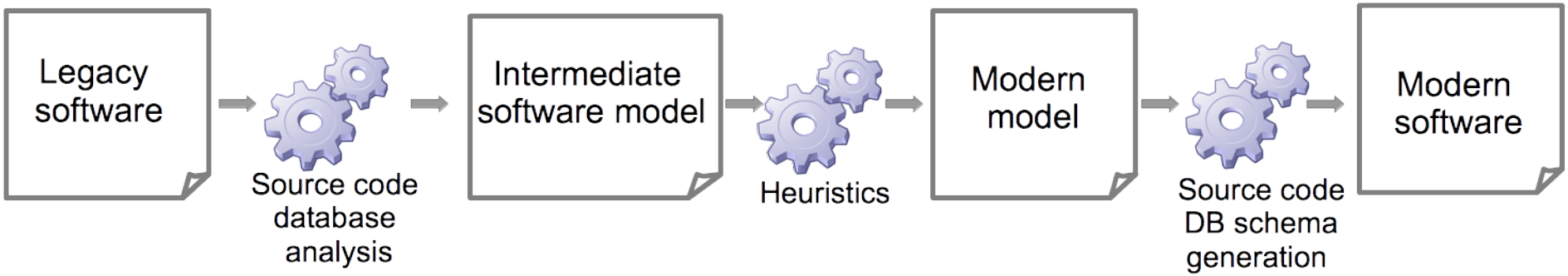
Group organization

- ACL Group (3): project management + 1 extraction heuristic
- CPS Group (3): analyze MMI code + 1 extraction heuristic
- Moretz Group (4): analyze SQL and VB code + 1 extraction heuristic

The proposed papers

- [Sahraoui et al., 1999]
- [Canfora et al., 1999]: minimization of coupling
- [Cimitile et al., 1999]: manual part, metrics+routine assig. algo.
- [van Deursen and Kuipers, 1999]
- [Lucca et al., 1997]: metrics+routine assig. algo.
- [Bhatti et al., 2008]: FCA on bad object design
- [Glavas and Fertalj, 2011]
- [Maletic and Marcus, 2001]: LSI + semantic clustering
- [Zou and Kontogiannis, 2003]: ad hoc alg. for amalgamating class properties

The generic process



Data extraction and encoding

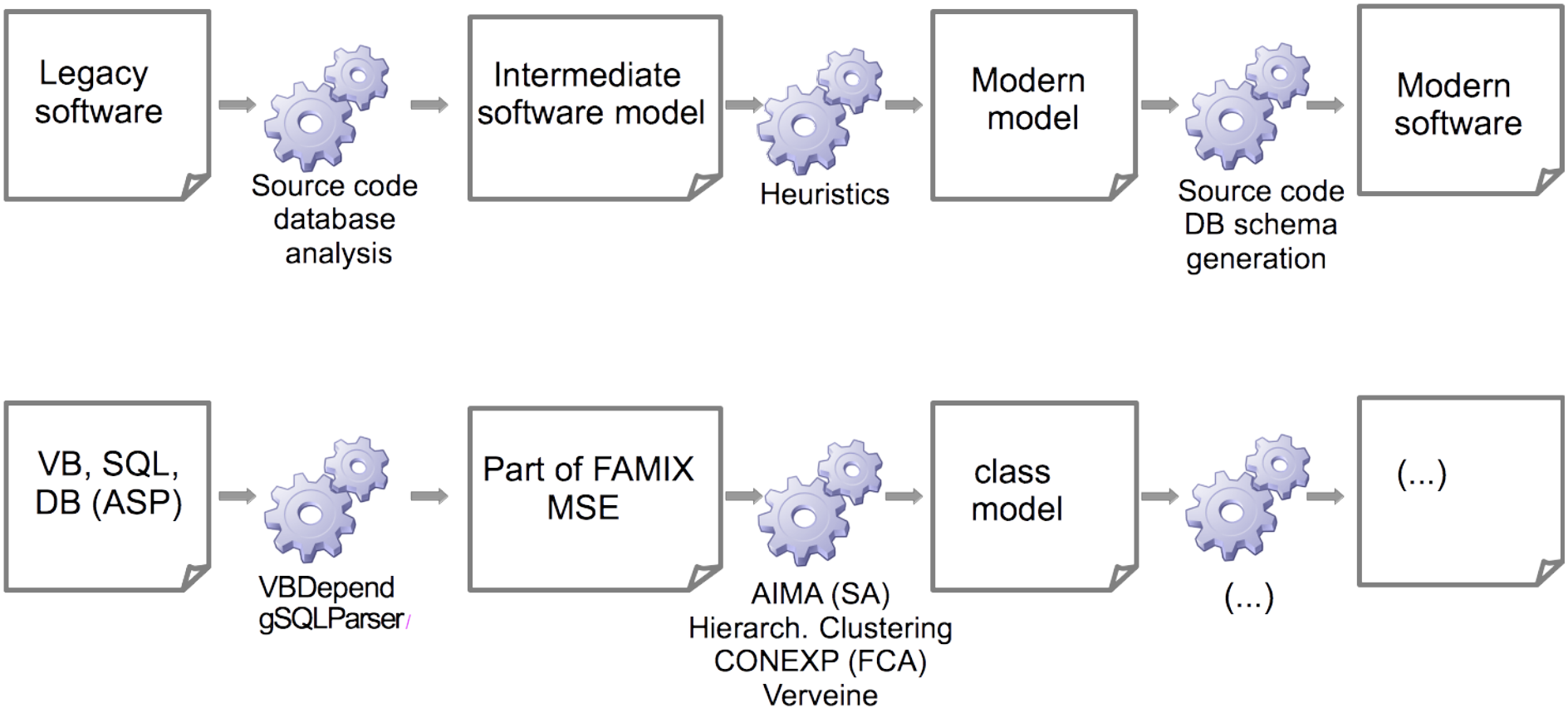
Expected input data: tables, columns, functions, access, invocation
Tools:

- FAMIX / MSE format, Verveine (http://www.moosetechnology.org/docs/famix)
- VBdepend (http://www.vbdepend.com)
- GSP (http://www.sqlparser.com)

Missing:

- VBdepend and GSP not free (trial versions were used)
- database representation in FAMIX
- analyzing VB functions where parameters are the SQL function and its parameters
- merge VB analysis result and SQL analysis result

The instantiated process



[Sahraoui et al., 1999] FCA++

- FCA: data *is accessed by* routine
- select concepts by decreasing routine number and increasing data number
- classes are given by data part of the concepts
- merge concepts that have more in common than not in common
- assign functions to classes when they refer or modify them

In current project:

- data are columns of the database tables
- routines are functions that directly have access to columns

Tools:

- Concept Explorer (http://conexp.sourceforge.net)
- specific code for creating Formal Context and exploit the concept lattice

[van Deursen and Kuipers, 1999] Dendogram

Hierarchical clustering on data similarly accessed by functions

- Create a CRUD matrix: data $\times$ functions
- calculate a distance matrix between data based on CRUD matrix
- build a dendogram based on distance and a chosen cut point
- assign functions to classes when they refer or modify only one class

Tools:

- Entirely implemented

[Glavas and Fertalj, 2011]

Meta-heuristics

- focus in the project: Simulated annealing
- solution: a set of candidate classes composed of data and functions
- fitness functions: software metrics (coupling, cohesion)

Tools:

- AIMA framework (implements *Peter Norvig And Stuart Russell's "Artificial Intelligence - A Modern Approach 3rd Edition."*) (http://code.google.com/p/aima-java/)
- specific Java code to connect to MSE files

Application size

Software size (the smallest)

- two databases: 45 tables
- SQL+ VB code:
  - smallest software: 346 functions, 26042 LOC

Results on TR software (smallest - 45 tables)

FCA++

	attributes			methods		
#class	#min	#max	#av	#min	#max	#av
74	2	165	10	0	30	8

Dendogram-11

	attributes			methods		
#class	#min	#max	#av	#max	#min	#av
20	1	36	12.8	0	12	2.5

Analysis

FCA++

- many classes
- post-treatment creates many duplications
- attributes poorly distributed
- merging method is too strict
+ all methods are assigned

Dendogram

+ reasonable class number (correspond to connected tables)
- few assigned methods

Simulated annealing

- difficulty to understand weighting in metrics
- impossible to reproduce results of the paper on the included example

Conclusion

- not so easy to reproduce paper results
- no good results of FCA++ approach due to post-treatment
- limited results of dendogram approach for method assigment

→ Dendogram results have been chosen by the company for detailed study

Perspectives

- Change FCA++ post-treatment
- Add better method assigment to Dendogram
- Finalize Simulated Annealing
- Apply identifier analysis to tables/variables/columns names
- Use database schema
- Use MMI and interactions
- what about associations?

Thank you!

Bhatti, M. U., Ducasse, S., and Huchard, M. (2008).
Reconsidering classes in procedural object-oriented code.
In *International Conference on Reverse Engineering (WCRE)*.

Canfora, G., Cimitile, A., Lucia, A. D., and Lucca, G. A. D. (1999).
A case study of applying an eclectic approach to identify objects in code.
In *IWPC*, pages 136–143. IEEE Computer Society.

Cimitile, A., Lucia, A. D., Lucca, G. A. D., and Fasolino, A. R. (1999).
Identifying objects in legacy systems using design metrics.
Journal of Systems and Software, 44(3):199–211.

Glavas, G. and Fertalj, K. (2011).
Solving the class responsibility assignment problem using metaheuristic approach.
CIT, 19(4):275–283.

Lucca, G. A. D., Fasolino, A. R., Guerra, P., and Petruzzelli, S. (1997).
Migrating legacy systems towards object-oriented platforms.
In *ICSM*, pages 122–129. IEEE Computer Society.

Maletic, J. I. and Marcus, A. (2001).
Supporting program comprehension using semantic and structural information.

In Müller, H. A., Harrold, M. J., and Schäfer, W., editors, *ICSE*, pages 103–112. IEEE Computer Society.

Sahraoui, H. A., Lounis, H., Melo, W. L., and Mili, H. (1999).
A concept formation based approach to object identification in procedural code.
Autom. Softw. Eng., 6(4):387–410.

van Deursen, A. and Kuipers, T. (1999).
Identifying objects using cluster and concept analysis.
In Boehm, B. W., Garlan, D., and Kramer, J., editors, *ICSE*, pages 246–255. ACM.

Zou, Y. and Kontogiannis, K. (2003).
Incremental transformation of procedural systems to object oriented platforms.
In *COMPSAC*, pages 290–295. IEEE Computer Society.