Automatic Synthesis of Graphical User Interfaces for Health Information Systems

Síntese Automática de Interfaces Gráficas de Usuário para Sistemas de Informação em Saúde

Iuri M. Teixeira, Regina M. Braga, Antônio Tadeu A. Gomes

Universidade Federal de Juiz de Fora (UFJF), Juiz de Fora, Brasil
Laboratório Nacional de Computação Científica (LNCC), Petrópolis, Brasil

September 23, 2012

Introduction

- Model-Driven Development (MDD) for Health Information Systems (HIS) is still a challenge.
  - Transcribing the clinical's knowledge to computer systems is not trivial.
  - One way of mitigating such problem is the use of **clinical data models** (ex.: *open*EHR).
- *open*EHR models help separate the knowledge of the **clinical** domain from the **technological** domain.
  - However, they still demand too much effort to develop HIS implementations.
  - Such models are not sufficient for the construction of HIS (e.g. lack of **architectural information**).

Introduction

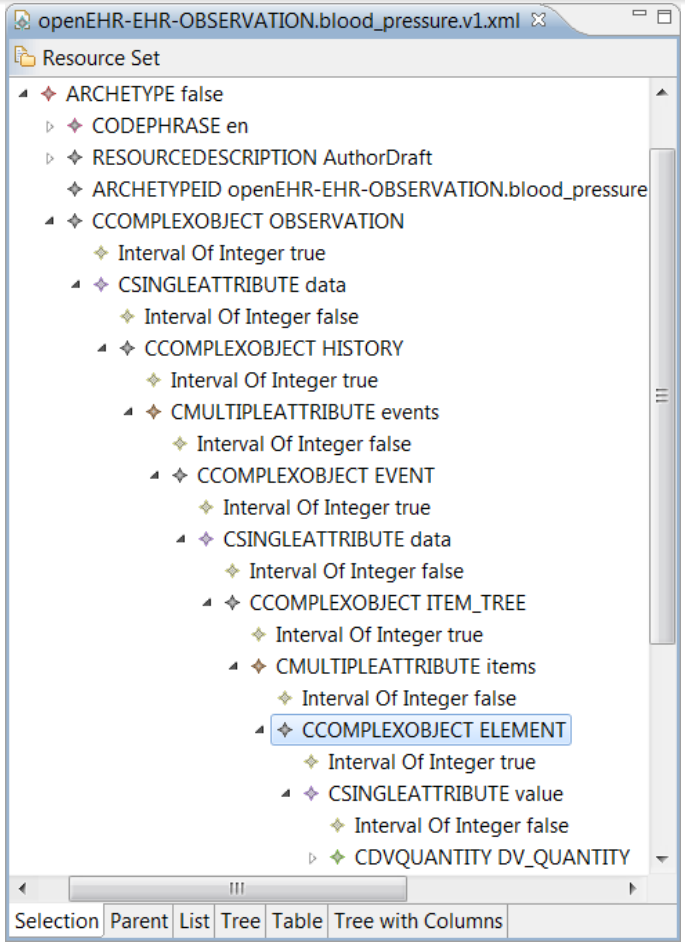
Objective

- Provide an architecture for the automated code generation of GUI for HIS from clinical data models.
  - MDD techniques based on a repository of **reusable rules** for model transformation.
- With this architecture, not only the implementation of HIS but also the specification of transformation rules can be simplified.

Basic principles

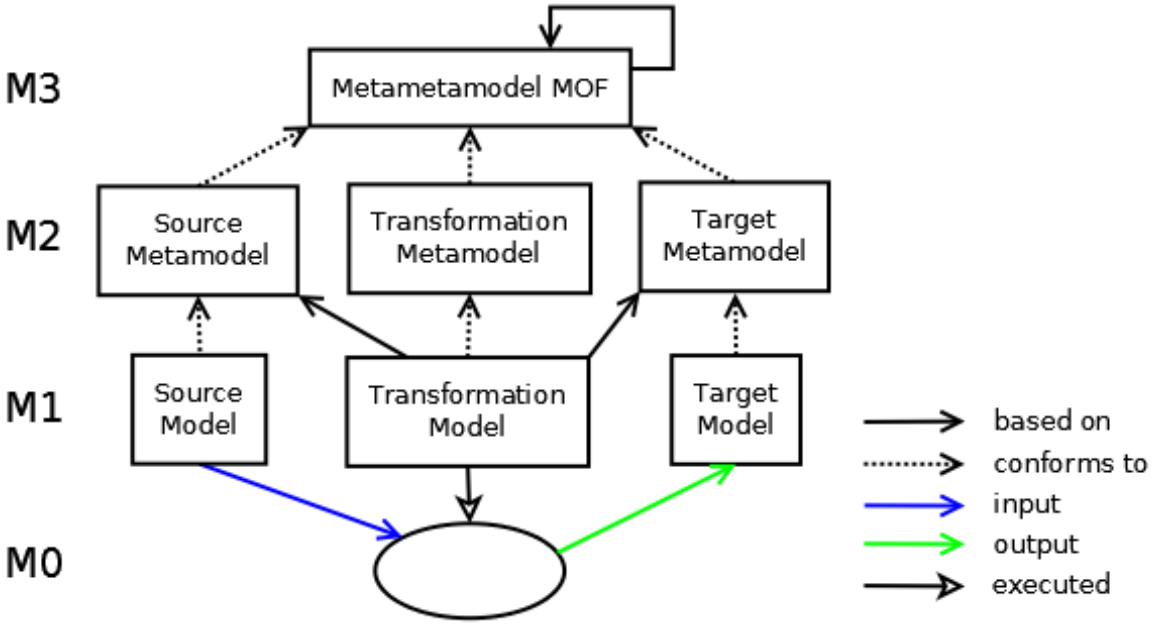
Clinical data models – *open*EHR

- *open*EHR is an open and collaborative specification for interoperable clinical data models.
- Metamodels: overall characteristics of the components of the health record.
- Models (archetypes): formal definitions of prescribed combinations of metamodel elements that describe clinical domain concepts.
  - Reuse of domain concepts in healthcare.
  - Reduced dependence among physicians and technologists in the development process.



Example of an archetype in the Eclipse platform.

Basic principles
Model Transformation



Model Transformation

Related work

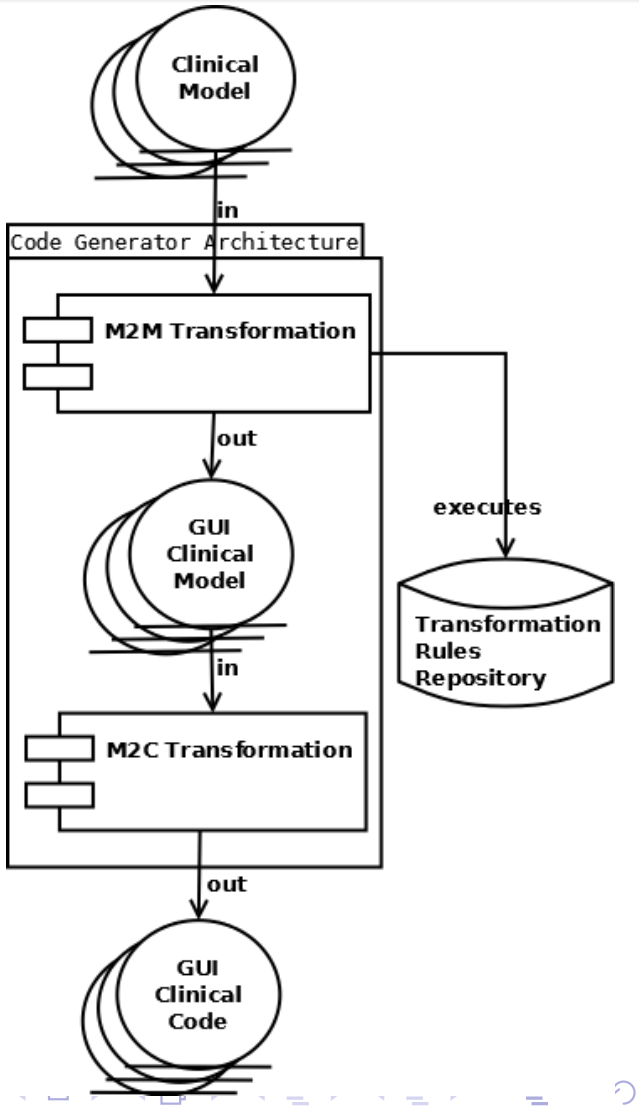
- GUI code generation for HIS from clinical data models [Schuler et al, 2006, Nardon et al, 2007, Van Der Linden, 2009, Atalag e Yang, 2010].
  - Do not employ metamodeling and transformation rules.
  - Compromises design reuse in MDD.
- GUI code generation with metamodeling and model transformations [Costa, 2011].
  - Not restricted to HIS and does not use a common metametamodel.
  - Compromises the reuse of clinical concepts and, therefore, the HIS interoperability.
- The work [Menárguez-Tortosa et al, 2011] explores the aforementioned concepts, however, does not consider architectural information.

Proposed approach

- Until the submission of this article, two different strategies for defining model transformation rules that combine clinical data models and architectural information were evaluated:
  1. Transformation without explicit architectural information.
  2. Transformation based on architectural annotation models.
- A third strategy is being investigated right now:
  3. Transformation based on Acme models.

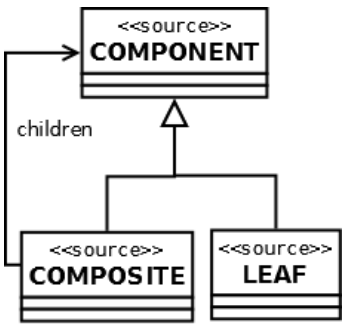
Architecture for code generation of GUI for HIS

- First step
  - M2M transformation (clinical data models to GUI models)
  - This process is executed through a repository of reusable rules.
- Second step
  - M2C transformation (GUI models to GUI code).
  - This process is executed through tools (RichUbi [Cirilo et al, 2010]) that generate code from models.



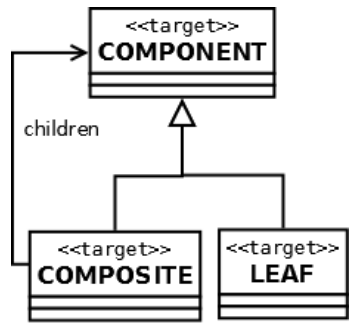
Rules repository

- Rules are structured according to the *open*EHR metadata hierarchy to generate different GUI elements.
  - Allows to capture the structure from archetypes information.
  - Also allows to apply **design patterns** on the transformations rules.



SOURCE

```
1  create target: TARGET from  source : SOURCE;
2
3  rule Component2Component{
4      from source: SOURCE!Component
5      to target: TARGET!Component
6  }
7
8  rule Composite2Composite extends Component2Component{
9      from source: SOURCE!Composite
10     to target: TARGET!Composite(
11         children <- source.children
12     )
13 }
14
15 rule Leaf2Leaf extends Component2Component{
16     from source: SOURCE!Leaf
17     to target: TARGET!Leaf
18 }
```



TARGET

First strategy

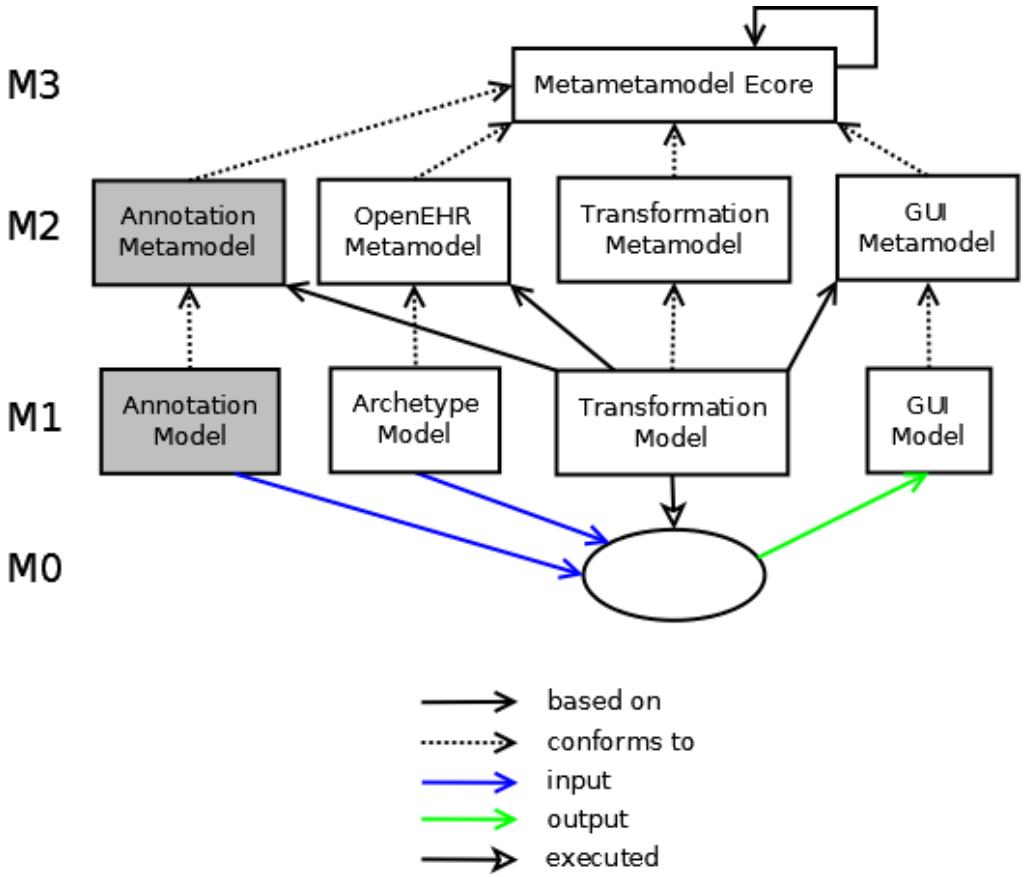
Transformation without explicit architectural information.

- Our hypothesis for this strategy is that it leads to a combinatorial explosion of rules (different rules for different combinations among different *families* of HIS).
- Without further architectural information, the developer will have a cognitive overload in defining applicable rules for different families of HIS.

Second strategy

Transformation based on architectural annotation models

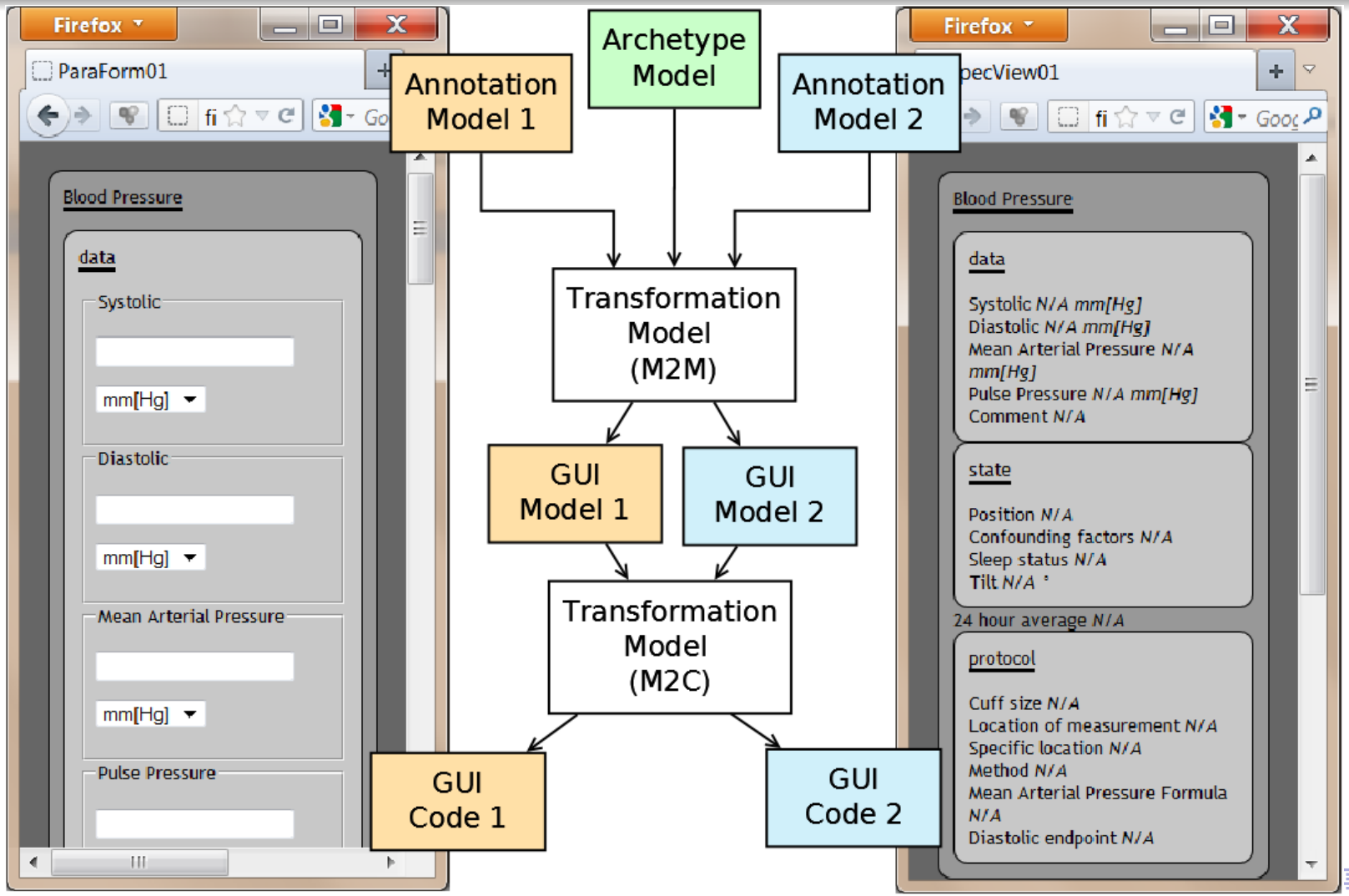
- Inclusion of models for architectural annotations.
- Metamodels for annotations allows to create models using *property-value* pairs.



Relationship between annotation models and the model transformation proposed.

Second strategy

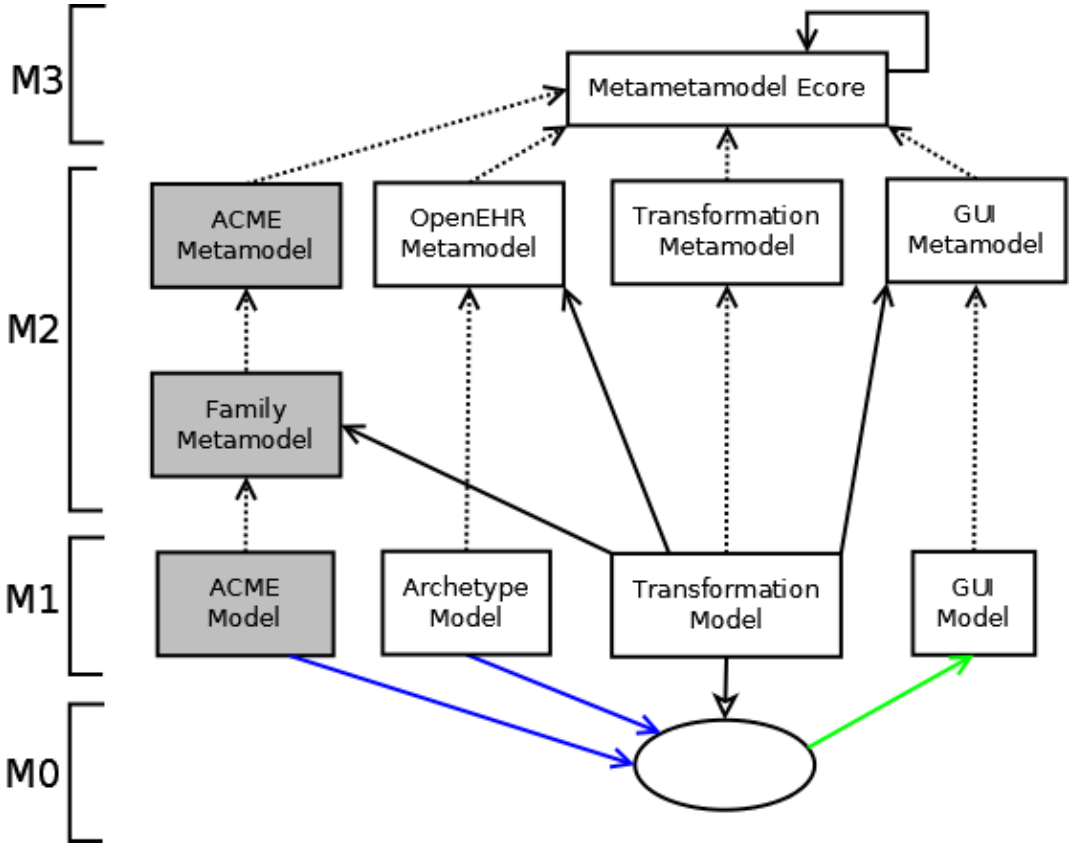
Transformation based on architectural annotation models



Current state of work – Third strategy

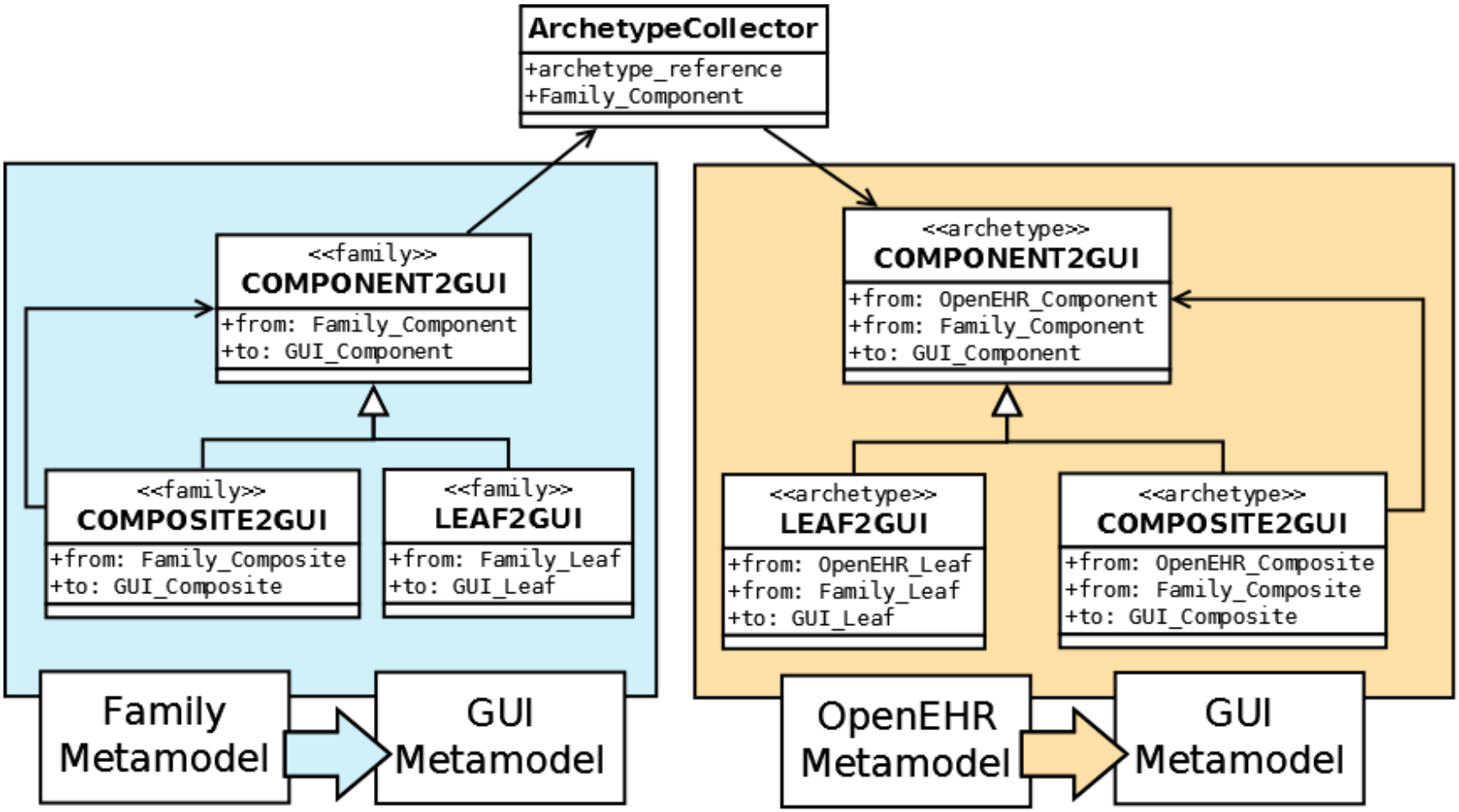
Transformation based on Acme models

- Acme is an Architecture Description Language (ADL)
- Acme has its metamodel specifications mapped onto the Ecore metametamodel.
- The current state of the work consists in exploring families of HIS described using Acme's Ecore metamodel.



Current state of work – Third strategy

Transformation based on Acme models



Conclusions

- Transformation rules are formally specified from the clinical data models.
  - Code generation of GUI for HIS with reuse of domain concepts.
- Employment of Ecore metametamodel allows greater interoperability of software.
- First experiments:
  - Exploration of two families: epidemiological surveillance and prehospital emergency healthcare
    - Conducted under the SPLiCE project (Software-Product Lines in HealthCarE), a software product line for HIS being developed on the National Institute of Science and Technology on Medicine Assisted by Scientific Computing (INCT-MACC).

References I

[Atalag e Yang, 2010] Atalag, K.; Yang, H. Y.
From openEHR Domain Models to Advanced User Interfaces: A Case Study in Endoscopy.
Health Informatics New Zealand Conference, 2010.

[Cirilo et al, 2010] Cirilo, C. E.; Prado, A. F.; Souza, W. L. D. ; Zaina, L. A. M.
Model driven richubi: a model driven process for building rich interfaces of context-sensitive ubiquitous applications.
Proceedings of the 28th, p. 207–214, 2010.

[Costa, 2011] da Costa, S. L.
Uma Abordagem Baseada em Modelos para Construção Automática de Interfaces de Usuário para Sistemas de Informação.
Dissertação de Mestrado. Goiânia, UFG, 2011.

[Menárguez-Tortosa et al, 2011] Menárguez-Tortosa, M.; Martínez-Costa, C. ; Fernández-Breis, J. T.
A generative tool for building health applications driven by iso 13606 archetypes.
J Med Syst, 2011.

[Nardon et al, 2007] Nardon, F. B.; França, T. ; Naves, H.
Construção de aplicações em saúde baseadas em arquétipos.
Acta Scientiarum Biological Sciences, v.28, n.4, 2007.

[Schuler et al, 2006] Schuler, T.; Garde, S.; Heard, S. ; Beale, T.
Towards automatically generating graphical user interfaces from openehr archetypes.
In: Hasman, A.; Haux, R.; van der Lei, J.; Clercq, E. D. ; France, F. H. R., editors, MIE, volume 124 de **Studies in Health Technology and Informatics**, p. 221–226. IOS Press, 2006.

[Van Der Linden, 2009] Van Der Linden, H.; Austin, T. ; Talmon, J.
Generic screen representations for future-proof systems, is it possible? there is more to a gui than meets the eye.
Computer Methods and Programs in Biomedicine, v.95, n.3, p. 213–226, 2009.

Automatic Synthesis of Graphical User Interfaces for Health Information Systems

Síntese Automática de Interfaces Gráficas de Usuário para Sistemas de Informação em Saúde

Iuri M. Teixeira, Regina M. Braga, Antônio Tadeu A. Gomes

Universidade Federal de Juiz de Fora (UFJF), Juiz de Fora, Brasil
Laboratório Nacional de Computação Científica (LNCC), Petrópolis, Brasil

September 23, 2012