



## On Model Subtyping

Clément Guy

► **To cite this version:**

Clément Guy. On Model Subtyping. 4emes journées nationales du GDR-GPL 2012, Jun 2012, Rennes, France. <hal-00726399>

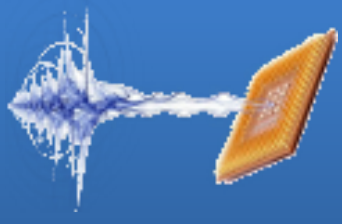
**HAL Id: hal-00726399**

**<https://hal.inria.fr/hal-00726399>**

Submitted on 30 Aug 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

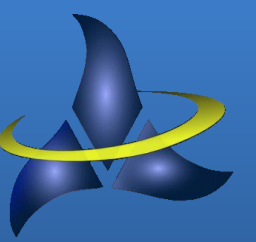
L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# On Model Subtyping

Clément Guy, Triskell and Cairn teams, IRISA-Inria

clement.guy@irisa.fr



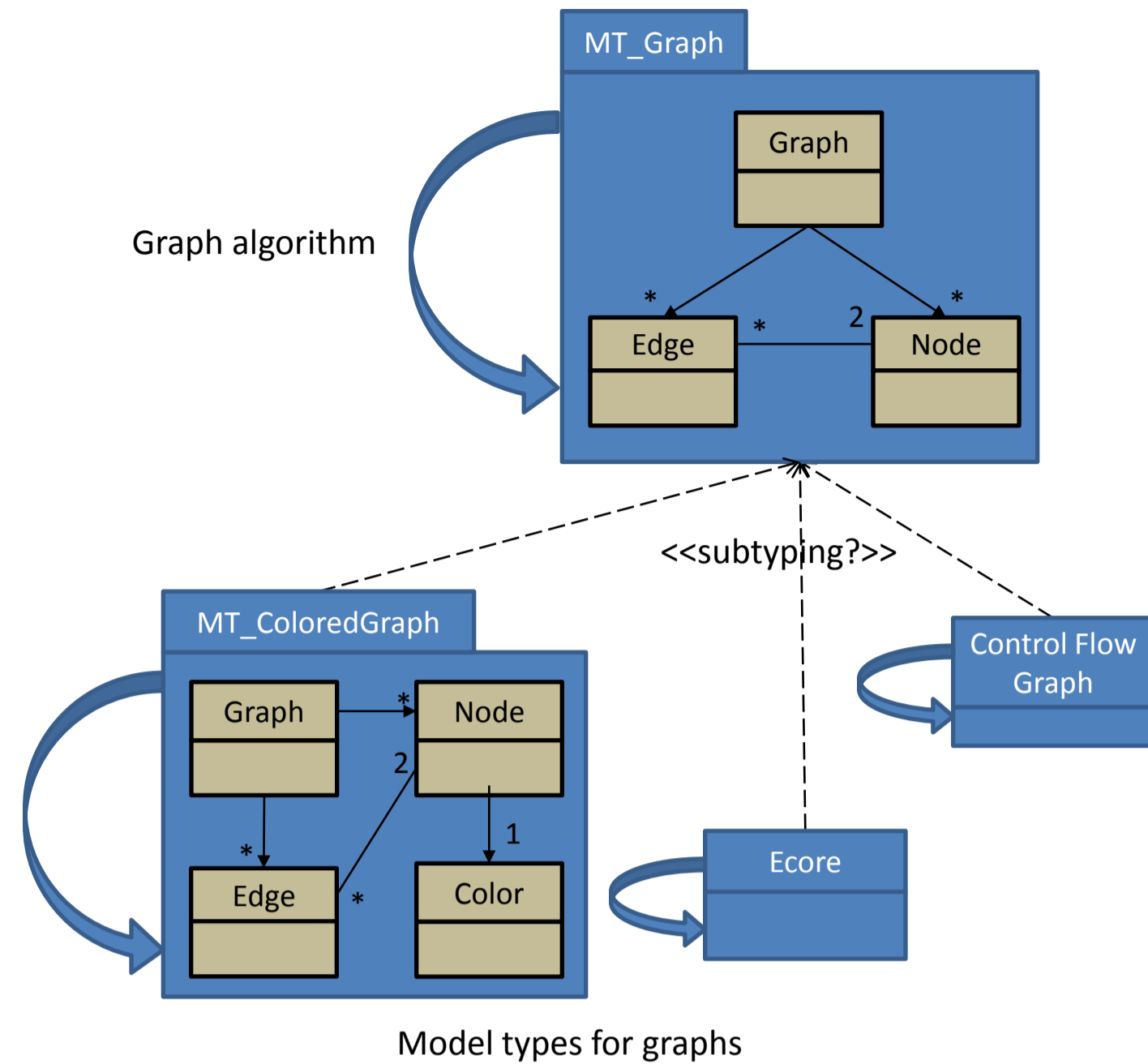
## CONTEXT

- Rapid increase of the number of modeling languages
  - More and more model manipulation operators
- Need for a systematic engineering
  - Providing design methods and **facilities** (reuse of operators and structures, advanced tools...)
- Existing approaches remain disconnected from each other
  - Need for a unified theory

## MODEL-ORIENTED TYPE SYSTEMS

- Model-oriented type systems should provide facilities such as abstraction, reuse, safety, auto-completion...

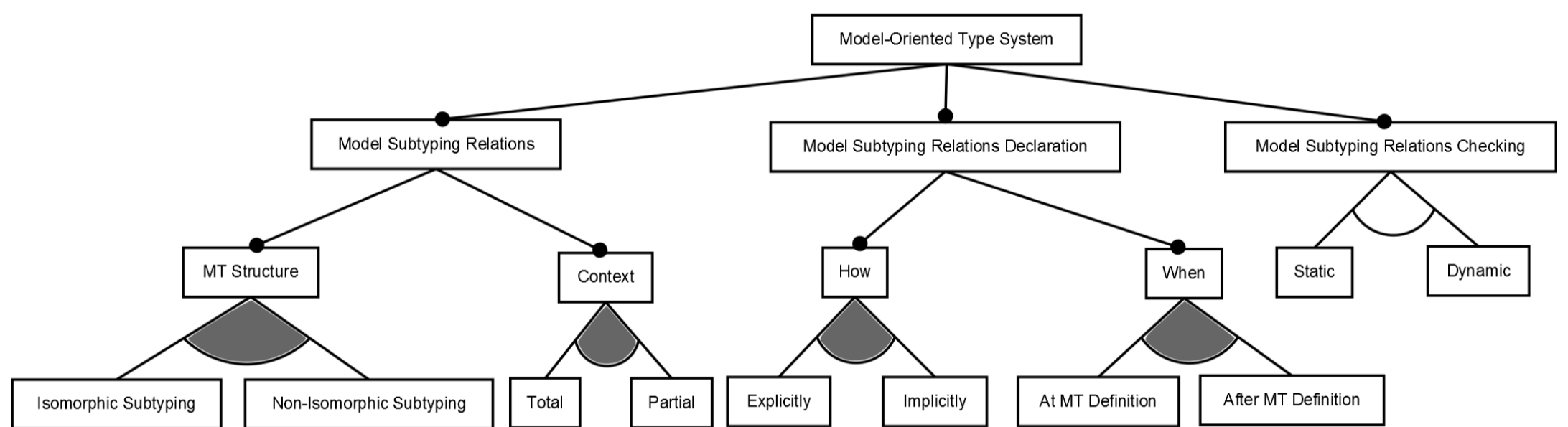
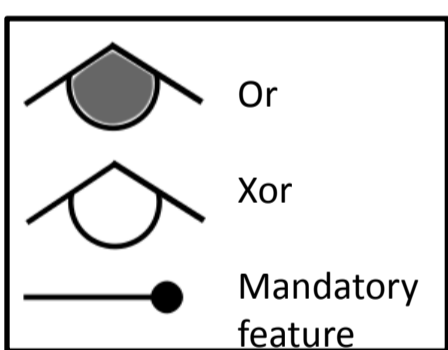
A type of a model is a set of types (MOF classes) of objects which may belong to the model, and their relations.



## MODEL SUBTYPING RELATIONS

- How can we safely use model typed by A where a model typed by B is expected?
- Structure of the model types
  - Isomorphic subtyping*: **Same** names, multiplicities...
  - Non-isomorphic subtyping*: **Adaptation** from the subtype to its supertype

- Context of the subtyping relation
  - Total subtyping*: Models typed by A can be safely used **everywhere** models typed by B are expected
  - Partial subtyping*: Models typed by A can be safely used **in a given context** in which models typed by B are expected (e.g., a given model transformation)



A family of model-oriented type systems

	Total / Partial	Isomorphic / -isomorphic	At / After definition	Explicit / Implicit	Checking	Legacy tool reuse
Varrò <i>et al.</i>	Total	Class renaming	After	Implicit	?	No
Cuccurru <i>et al.</i>	Total	Class renaming	After	Explicit	?	Yes
Steel <i>et al.</i>	Total	Class renaming multiplicities contraction	After	Implicit	At design time, with errors at runtime	Yes
Sanchez Cuadrado <i>et al.</i>	Total	Class renaming multiplicities contraction	After	Explicit	?	Yes
Sen <i>et al.</i>	Partial	Any adaptation	After	Explicit	At design time, with errors at runtime	Yes
De Lara <i>et al.</i>	Total	Class renaming navigation and filtering of properties, n-to-1 bindings	After (Binding) At (Specialization)	Explicit	?	No
Babau <i>et al.</i>	Total	Isomorphic	After	Explicit	?	Yes

Classification of existing approaches for model manipulation reuse

## CLASSIFICATION OF EXISTING APPROACHES

- Underused approaches:
  - Partial subtyping (i.e., subtyping wrt. a specific context)
  - Implicit declaration of a subtyping relation
  - Declaration at the definition of a type
  - Isomorphic subtyping (too restrictive)
- Lack of information on subtyping relations checking

## PERSPECTIVES

- Could we propose a type system providing each one of those features?