Industrialization of Research Tools: the ATL Case

Hugo Brunelière, Jordi Cabot, Frédéric Jouault,
Massimo Tisi, Jean Bézivin

AtlanMod
INRIA & EMN
Nantes, France
massimo.tisi@mines-nantes.fr

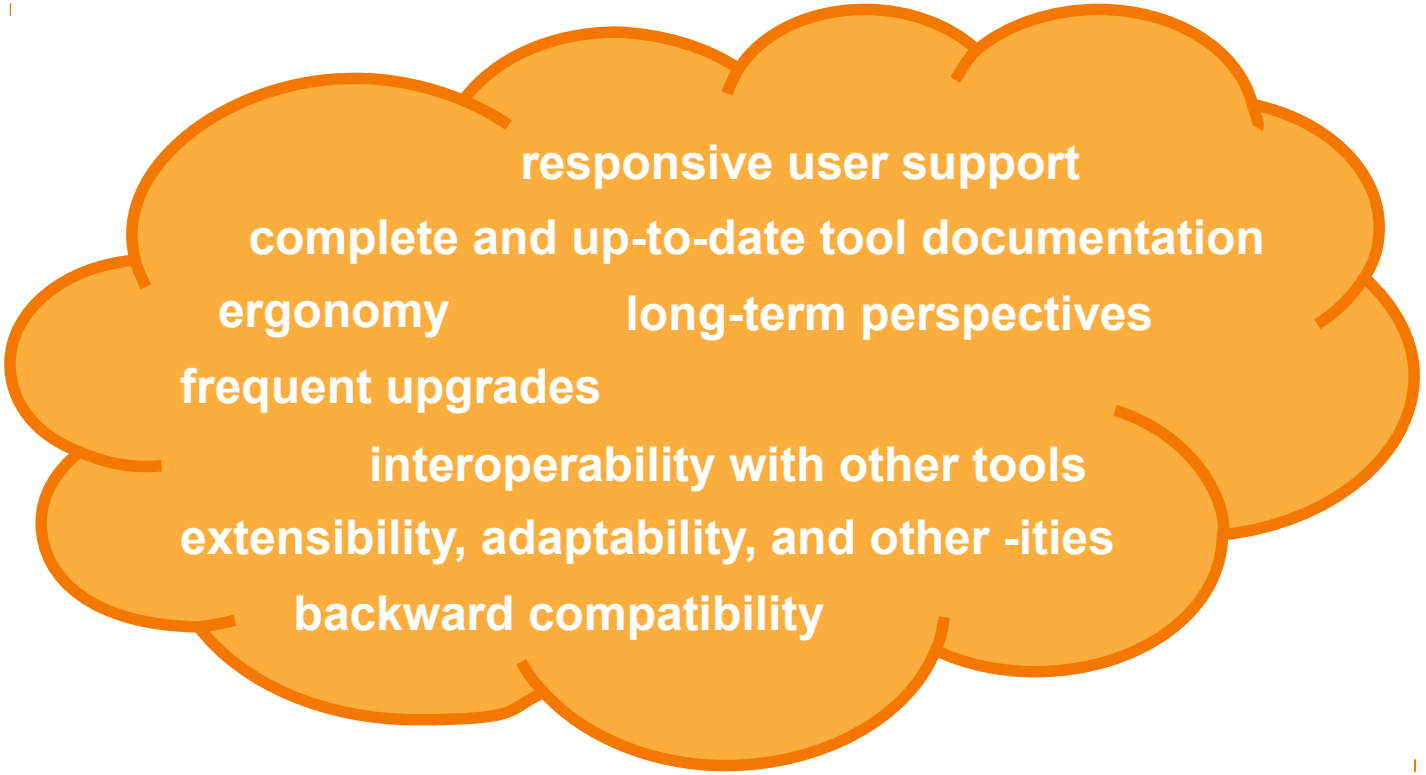
Research tools in SE

- Tools as proofs of concept for research ideas
- A large user base would produce:
  - empirical validation of research
  - feedback
  - visibility
  - collaboration opportunities

Unfortunately...

...resources are limited!

- Research groups focus on core research activities
  - also because of funding criteria
- Companies don't adopt the tool because it lacks:



Open Source is not enough

- An open source license is not enough:

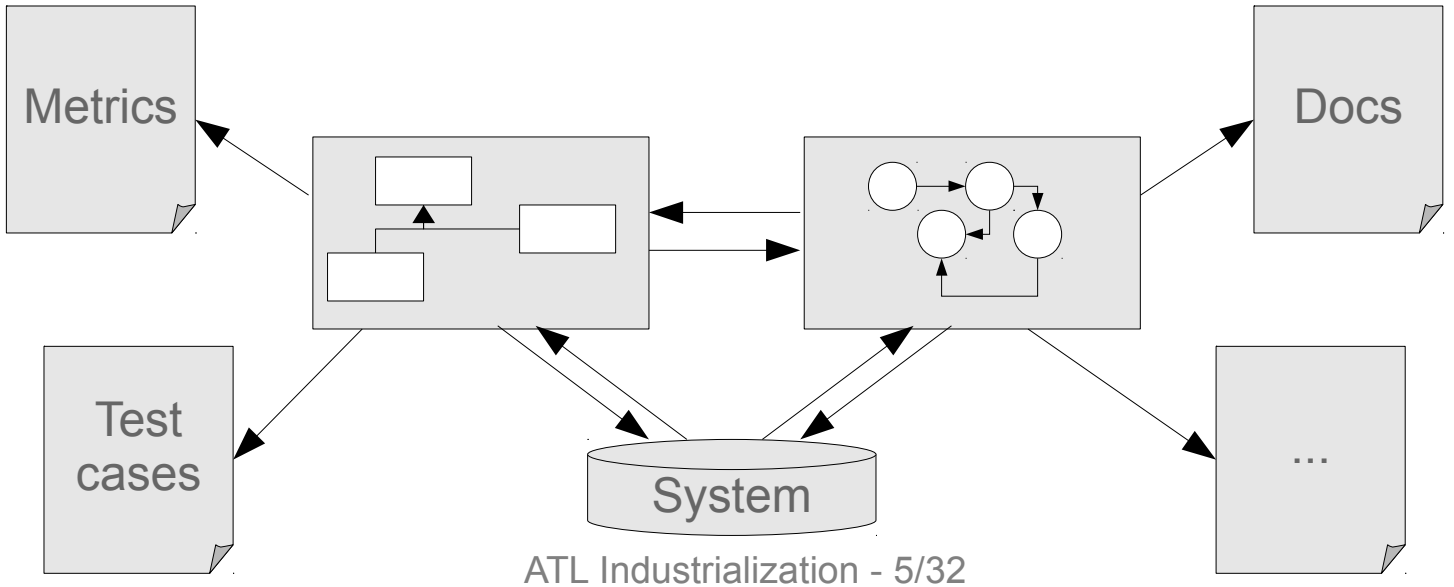
  - only few projects are successfully involving external contributors

[S. Krishnamurthy, Cave or community? an empirical examination of 100 mature open source projects]

  - external users prefer to work on challenging tool features
    - e.g. other research groups

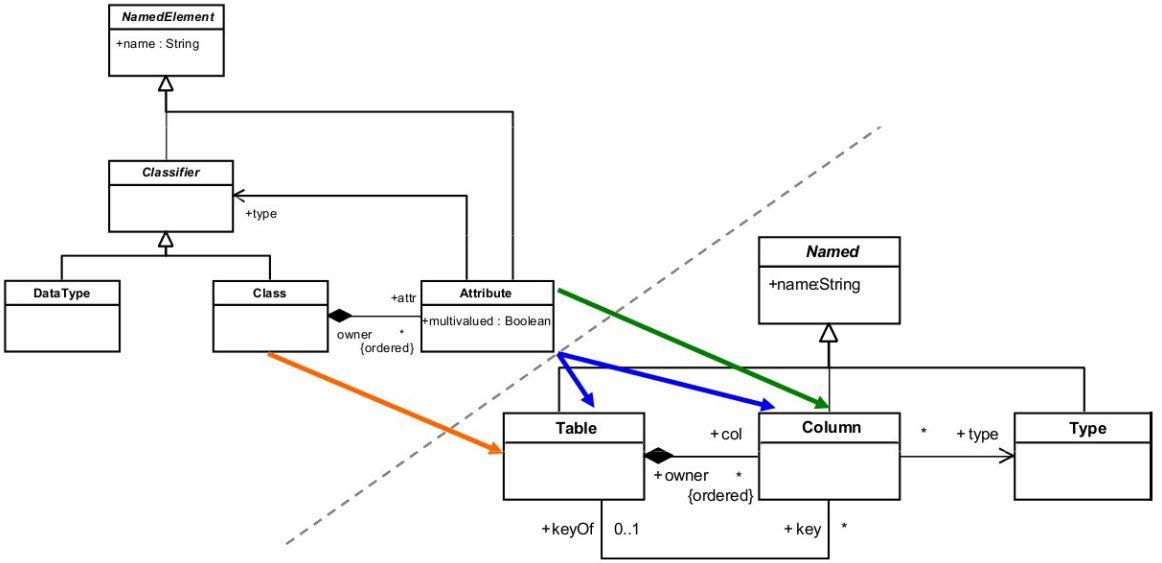
Our context: MDE

- Model Driven Engineering
  - models as primary artifacts in SE
    - growing industrial interest
    - without tools monopolizing the market

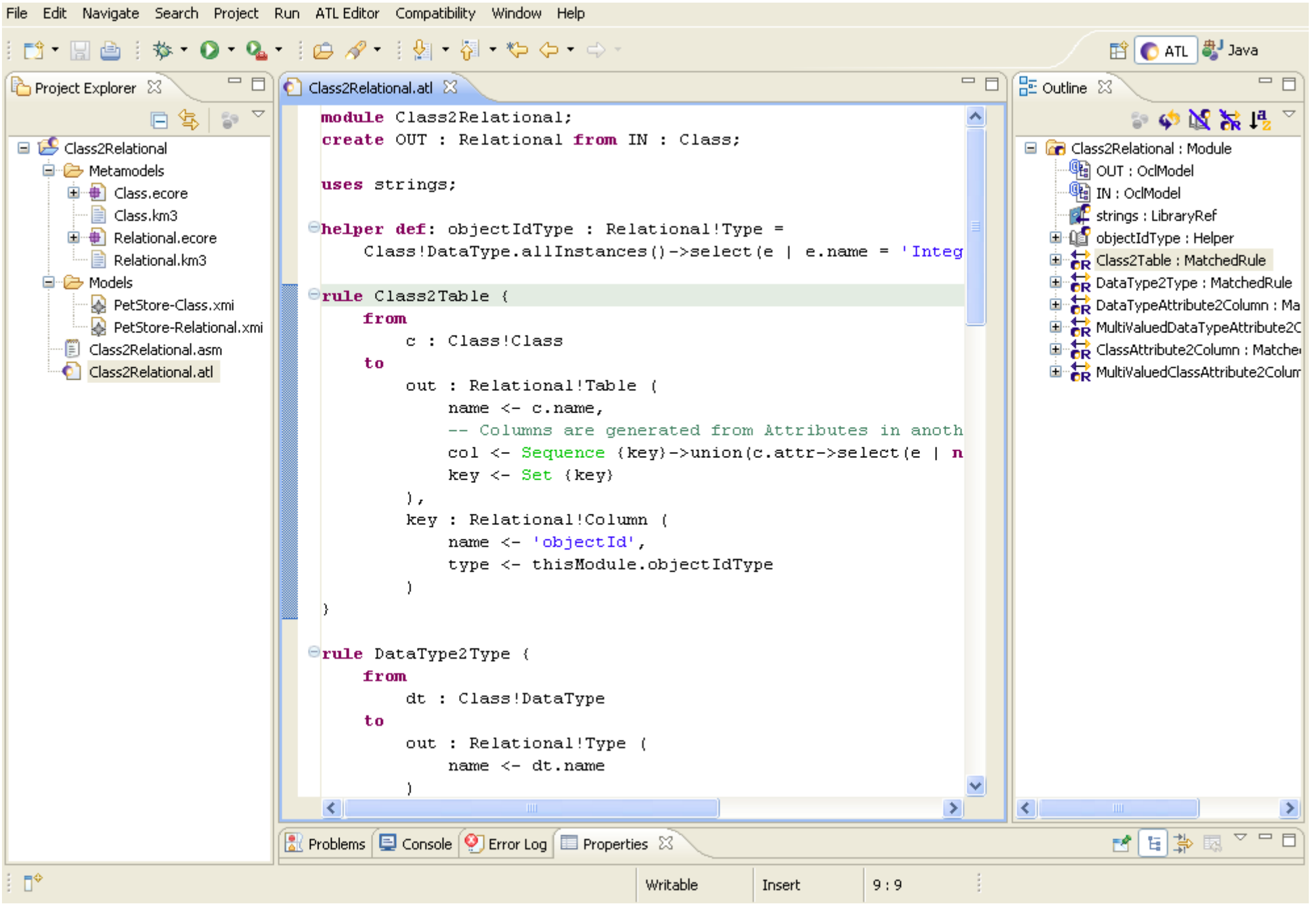


Our research tool: ATL

- Model-to-model (M2M) transformation language
  - production of target models from source models
  - a set of declarative rules that
    - match a subset of the source model
    - create an excerpt of the target model



A look to ATL

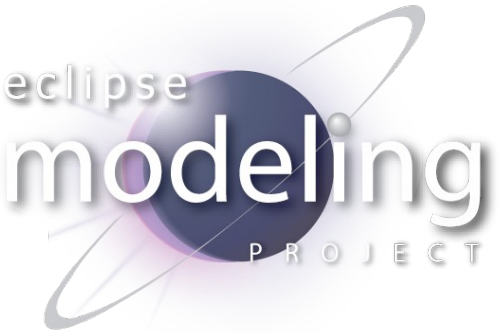

File Edit Navigate Search Project Run ATL Editor Compatibility Window Help
ATL
Java
Project Explorer
Class2Relational
Metamodels
Class.ecore
Class.km3
Relational.ecore
Relational.km3
Models
PetStore-Class.xmi
PetStore-Relational.xmi
Class2Relational.asm
Class2Relational.atl
Class2Relational.atl
module Class2Relational;
create OUT : Relational from IN : Class;
uses strings;
helper def: objectIdType : Relational!Type =
Class!DataType.allInstances()->select(e | e.name = 'Integ
rule Class2Table {
from
c : Class!Class
to
out : Relational!Table (
name <- c.name,
-- Columns are generated from Attributes in anoth
col <- Sequence {key}->union(c.attr->select(e | n
key <- Set {key}
),
key : Relational!Column (
name <- 'objectId',
type <- thisModule.objectIdType
)
}
rule DataType2Type {
from
dt : Class!DataType
to
out : Relational!Type (
name <- dt.name
)
Outline
Class2Relational : Module
OUT : OclModel
IN : OclModel
strings : LibraryRef
objectIdType : Helper
Class2Table : MatchedRule
DataType2Type : MatchedRule
DataTypeAttribute2Column : Ma
MultiValuedDataTypeAttribute2C
ClassAttribute2Column : Matche
MultiValuedClassAttribute2Colum
Problems
Console
Error Log
Properties
Writable
Insert
9 : 9

History of ATL

- **2003**: Phd of Frédéric Jouault
- **2004**: French project CAR-ROLL/MOTOR
- **2004**: ATL in the Eclipse-Modeling GMT project
- **2004-2006**: MODELWARE European integrated project
- **2006**: Eclipse M2M (Model-to-Model Transformation) project
- **2006-2010**: MODELPLEX European integrated project

ATL Today

- **ATL is the most popular model-transformation toolset today**
- Full Eclipse project



- ATL IDE
  - Editor, debugger, compiler, profiler, etc..
- Documentation
  - user guide, developer guide, tutorials, more than one hundred open source transformations, etc...
- Community
  - Eclipse forum, mailing list, bugzilla, wiki, etc..

How did we make it?

1. Tool design

a) Modularity

b) Standard technologies

c) Interoperability

d) Eating your own dog food

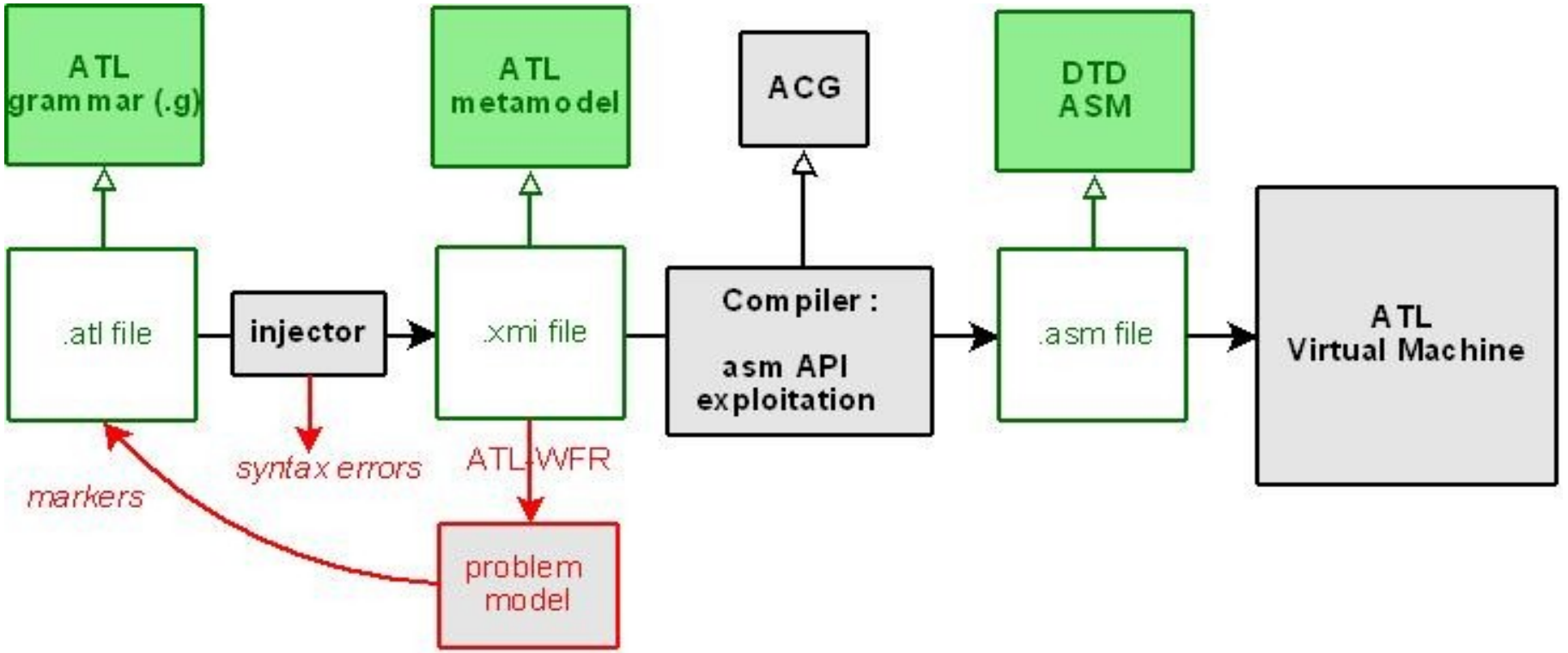
2. Business model

- **ATL Industrialization**

1. Tool Design

a) Modularity

- Modular structure + Eclipse extension mechanism



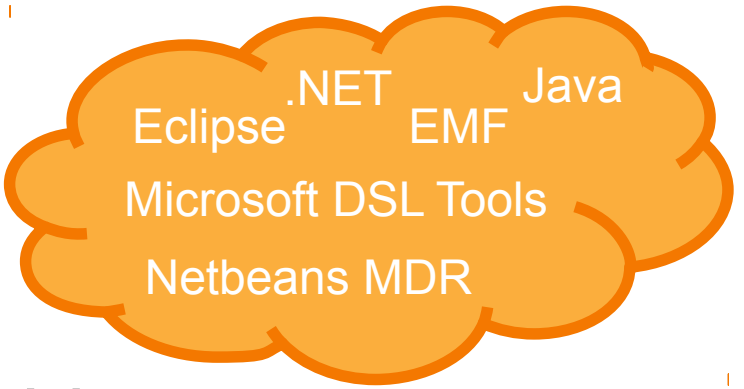
b) Standard technologies

- Eclipse 

  - open development framework
  - solid open-source community, standard platform
  - standard extension mechanism (based on OSGi)
- EMF
  - De facto standard solution for model handling today

c) Interoperability

- Maximum independence from specific technologies
  - → longevity
  - → portability



- Example: the ATL Virtual Machine
  - Specific for model manipulation
    - to support a new model management system
      - → new **Model Handler Abstraction Layer**
    - to execute ATL transformations in other environments
      - → porting the VM

d) Eating your own dog food

- Research groups as the first users of the tool
  - If possible, for building the tool itself (bootstrapping)
- Benefits
  - a non-trivial test for the tool
  - an improvement to a tool component improves all the toolsuite

2. Business Model

ATL Industrialization

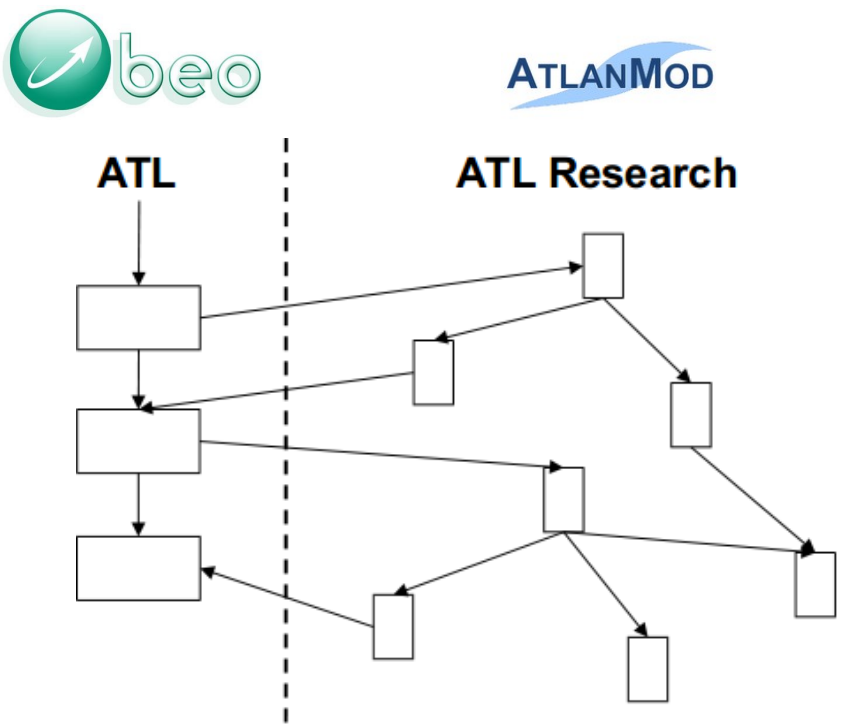
Benefits for AtlanMod	Benefits for Obeo
▪ Focus on new research activities ▪ ATL reaches an industrial quality attracting company users and a stable user base	• Strategic positioning among industries that use Eclipse modeling • Obeo becomes the lead ATL training company • ATL completes the proprietary Obeo offering • Privileged relationship with the creators and lead research contributors of ATL

Obeo responsibilities

1) Quality assurance
   - reactively (bug reporting and correction)
   - proactively (optimization, e.g. scalability)
2) Interoperability
3) Continuity
4) User experience
   - ergonomy, wizards, internationalization, etc..
5) Release management
   - defining milestones, building, testing, packaging and distributing
6) User support
   - documentation, free and subscription-based support and training

ATL Research

- ATL Research
  - specific versions for particular projects
  - if interesting for the general public, they move in ATL
    - by directly importing
    - or re-engineering by Obeo

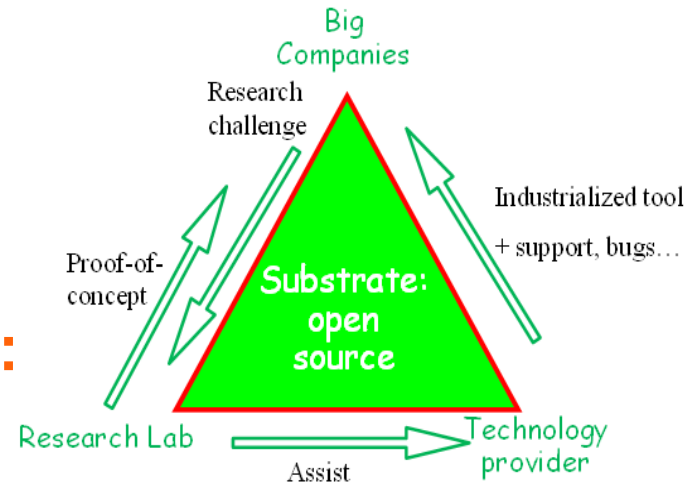


- Both have open-source licence
  - every update can be transferred between branches
  - third parties are guaranteed that any contribution will benefit the whole research community

Replicating the process

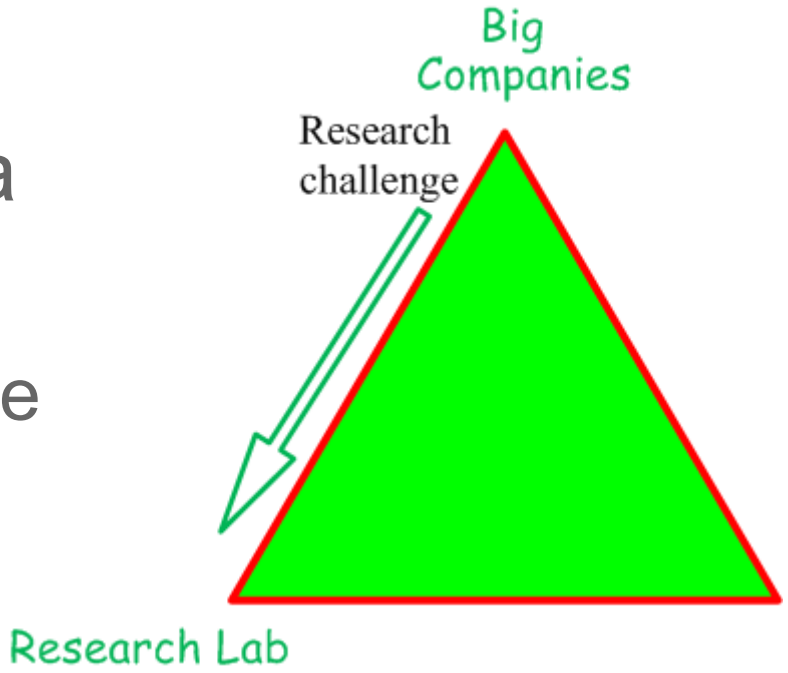
- We are replicating the industrialization model for:
  - **MoDisco** (software modernization tool)
    - with Mia-Software model-driven agility
  - **Megamodeling** (model management approach)
    - with PRODEVELOP Integración de tecnologías

The model can be generalized:



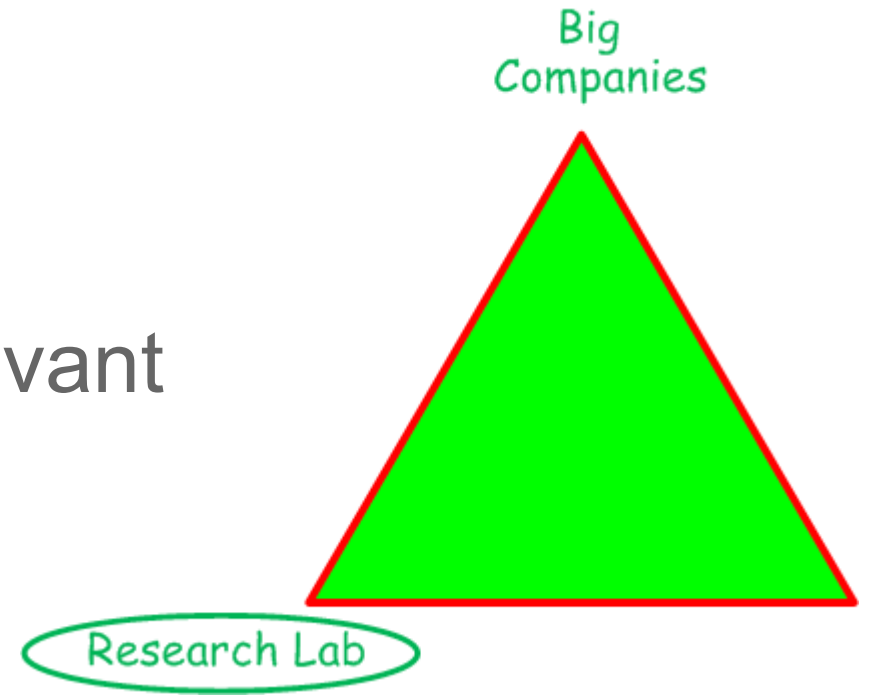
Industrialization Triangle

1. Big users describe a challenging problem
   - e.g. the industrial use cases at the beginnings of ATL



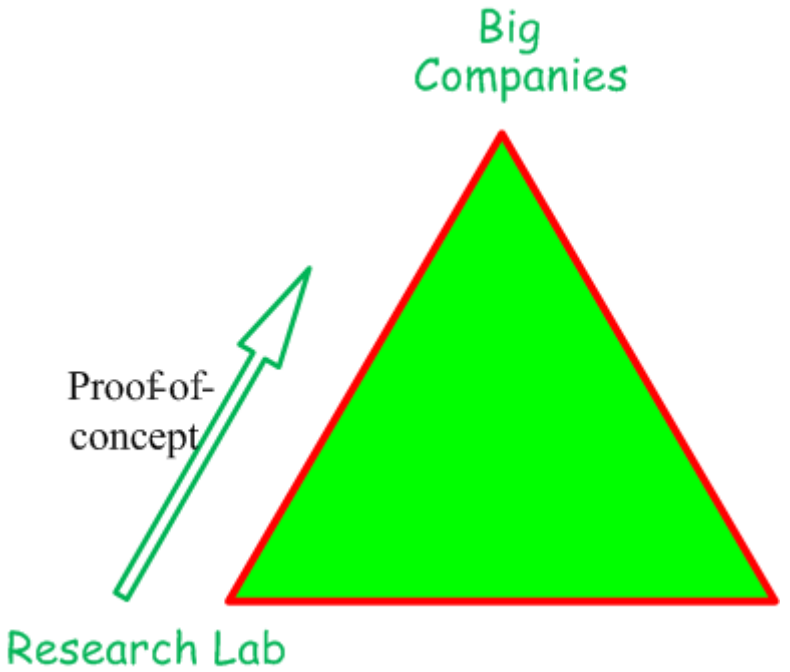
Industrialization Triangle

2. The lab decides whether it is a relevant research problem



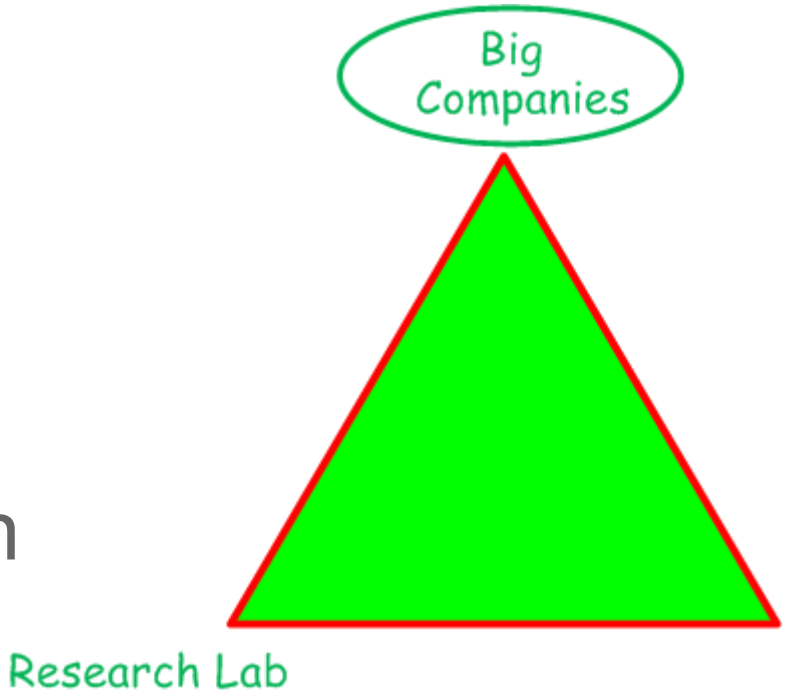
Industrialization Triangle

3. If it is, the lab
   - conducts the research
   - (publishes papers)
   - implements a proof-of-concept



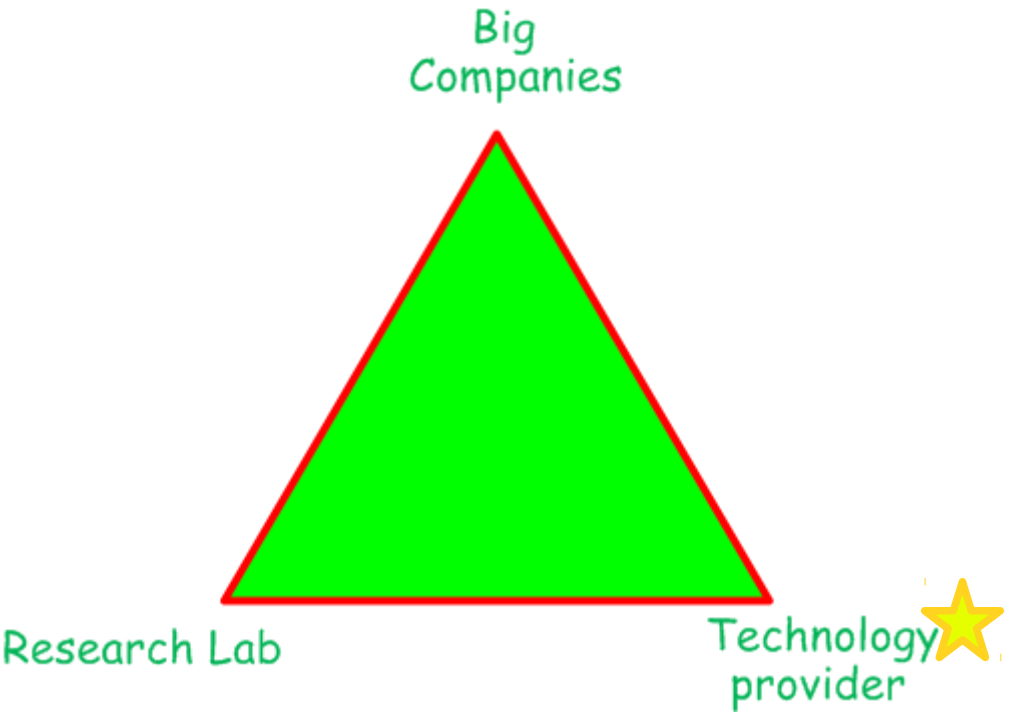
Industrialization Triangle

4. The big company decides whether to request an industrialized version



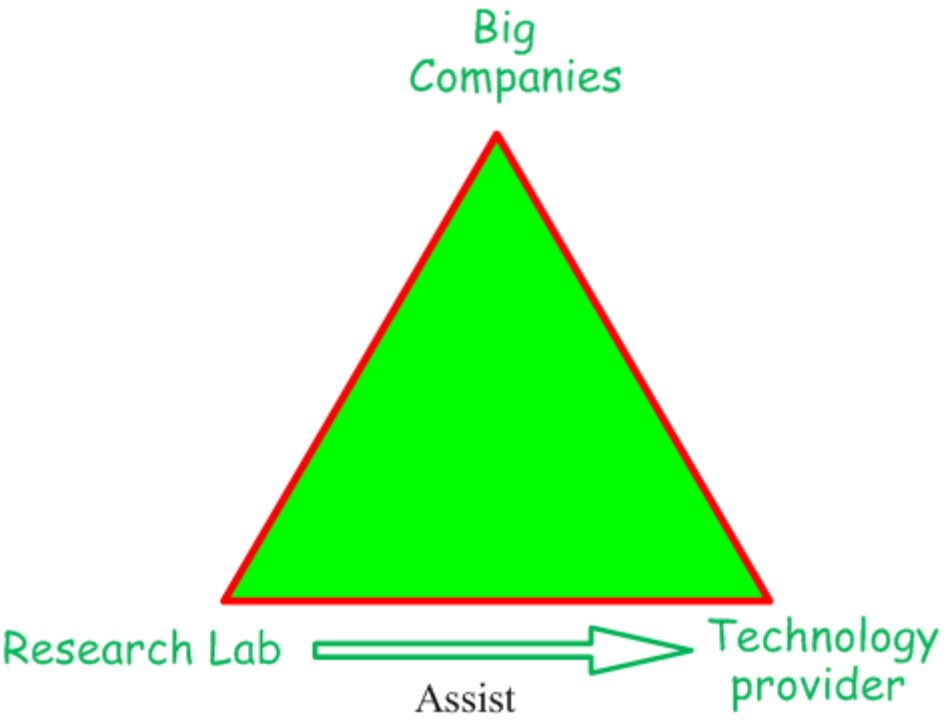
Industrialization Triangle

5. In that case, the big company, with the help of the research lab, selects a technology provider



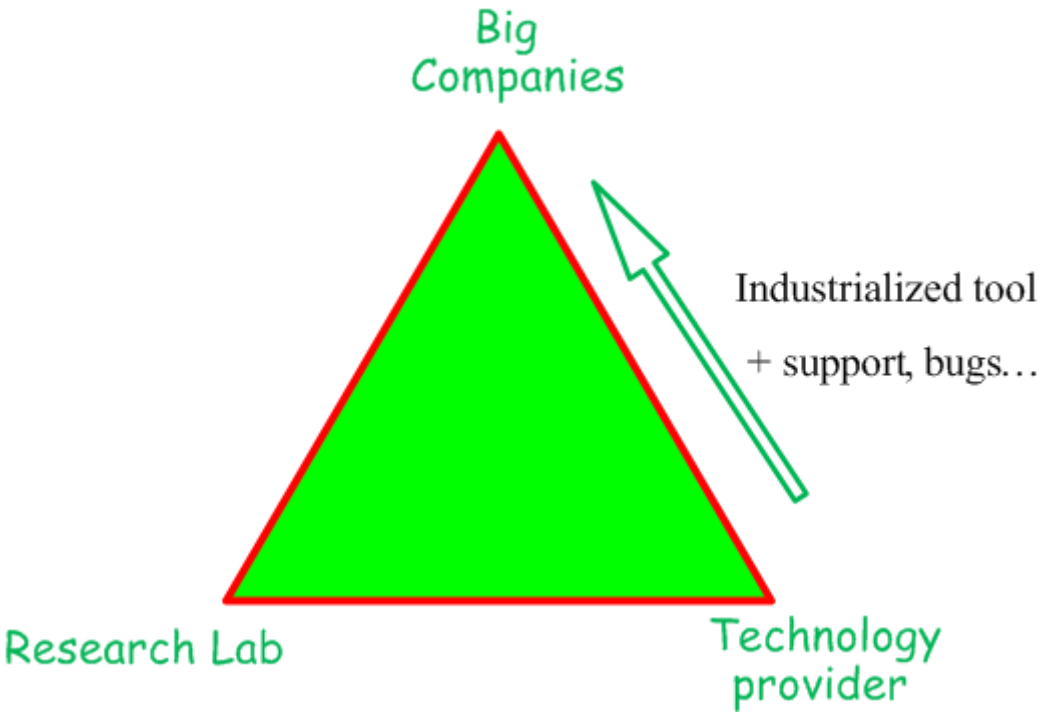
Industrialization Triangle

6. The lab assists the technology provider during the industrialization



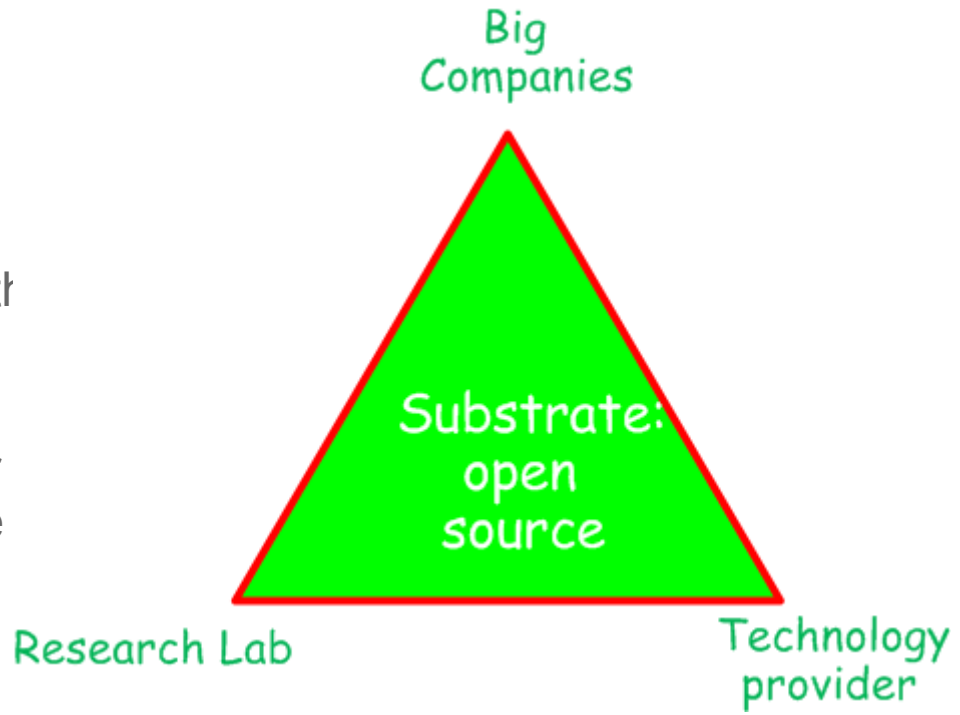
Industrialization Triangle

7. The technology provider releases the tool



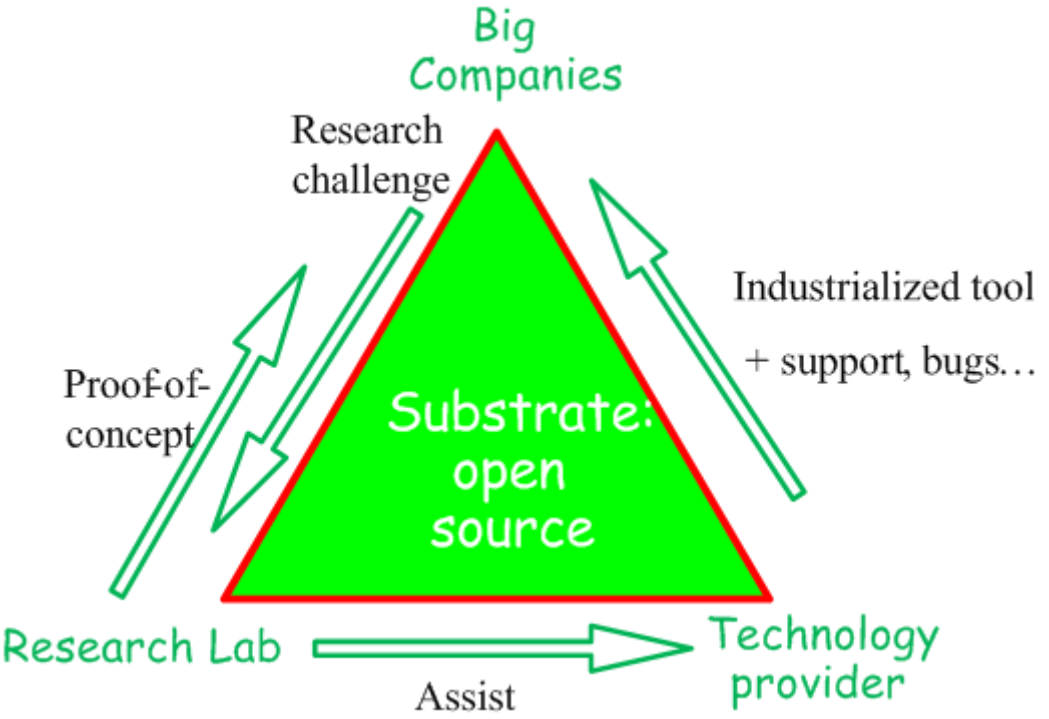
Industrialization Triangle

- Open source is not mandatory but
  - facilitates communication
  - maximizes benefits
    - the research group distributes th code when publishing papers
    - the technology provider commercializes adaptations for other companies with the same problem



All together

- **Application-driven research**
  - ensures return of investment for technology provider



Summarizing

- Our contribution:
  1. **Experience with the industrialization of ATL**
     - Design principles
     - Business model
  2. **Proposed model**
     - Industrialization of the research prototypes thanks to the partnership with a technology provider
     - Validated in ATL and replicated in other projects

Future work

- **Formalized protocol** for tool industrialization
  - investigated in the **OPEES** European project
- For ATL:
  - Fine-grained extension system for the ATL core
  - ATL Labs
    - Web-based collaboration platform for ATL extensions
    - users compose specialized extensions to get a specific ATL “flavor”

Thanks