Kermeta Day'09 – Rennes 2009/04/02

Breathe life into your metamodels

Kermeta in prodution mode

INRIA

IRISA

Agenda

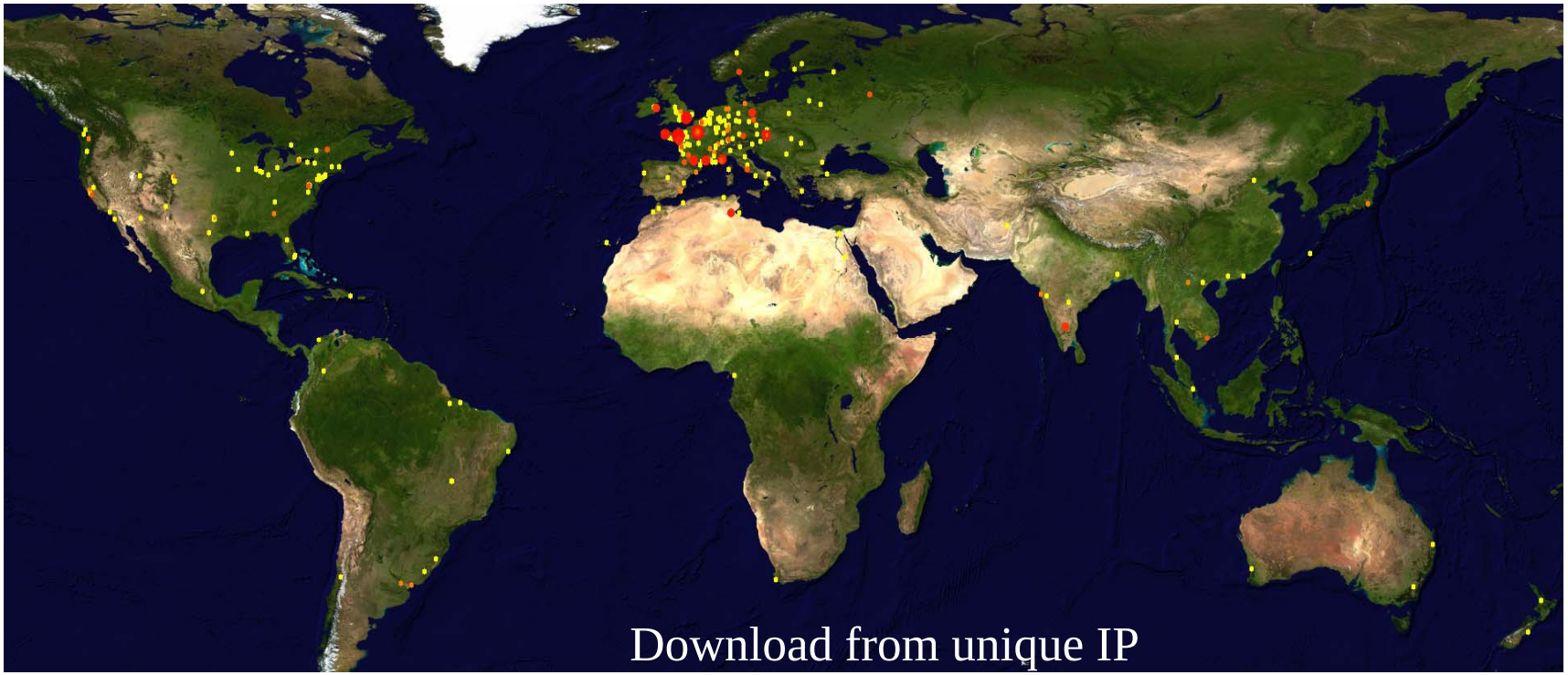
- Current status
- Future

Status

- Development started January 2005
- Current version (1.3.0)

Statistics 1/2

- 9000 download since 2006
- More than 300 articles using or citing Kermeta (cf. google.scholar)



Projects Using Kermeta

- Speeds, FAROS, DOMINO, S-Cube, MOVIDA; MOPCOM-I, ...
- OpenEmbedd
- Fiacre2AADL
- SCA To Java (Adam)
- SpoonEMF
- Teaching
  - Toulouse, Montpellier, Nice, Rennes, Nantes, ...

Language features reminder

- Object oriented
  - Support for modular programming and Design Patterns
- Model associations
  - ie. References and attributes
- Class re-opening (~lightweight aspects)
  - Reuse existing metamodels; Simplify your design
- Design by contract
  - Ie. inv, pre, post
- Function types
  - Ie. Write your own each, select, ...
- Model type
  - Ie. Apply the same code to various metamodels
- Generics
  - Stronger typechecking

Aspects and Kermeta (Open classes)

- Tends to be more and more helpful
  - Many design patterns are improved by using it
  - Even improve performances (ex: traceability, opposite on association, …)

- **MultiJava: modular open classes and symmetric multiple dispatch for Java** (OOPSLA 2000)

Kermeta development process

- Nightly build and tests
  - For kermeta and the major MDKs
  - With regression checks
- Feature request and bug tracker
  - For managing evolutions

Improved language features

- Some improvement in the framework
  - Refactoring : KM model manipulation moved to a dedicated MDK
  - Better support for invariants
    - allows analysis of the invariant for feedback to the end-user.
  - EPackage registration from kermeta
  - andThen and orElse (partial)
  - ...
- Some of the new extensions
  - Singleton
  - Date support

IDE features reminder

- Parser, type checker, interpreter
- Text Editor, Browser, Launcher
- EMF Ecore meta-model Import / Export
- EMF models Import / Export
- Debugger (currently not fully reliable)
- Graphical editor
- Constraint support; Support for OCL syntax
- Support of the concept of model : Jim's model type (need a small refactoring)
- Build process support (KPM)
- Easy installation and samples

Other improvements

- Ket
  - Now can generate kermeta aspects
- Kompose

IDE Improved features

- Improved feature
  - Support of Eclipse 3.4
  - Double click in outline
  - checkInvariant also check multiplicities
- New features
  - Kermeta doc view
    - also shows where the aspect comes from
  - Java compiler
    - Speed increase by x35 compared to interpreter

Kermeta compiler :

- New deployment = new possibilities
  - Deploy the resulting java in your user environment without Kermeta runtime itself
  - Many new usages might be investigated
    - Build a simulator
      - Solution 1 : Connect a graphical editor directly on top of your extended MM
      - Solution 2 : use EMF notifications to connect your extended MM with the one linked to the graphical editor
    - Offering an interpreter for your DSL to your user might be efficient enough
      - depends on your DSL speed requirements

Kermeta compiler

- Kermeta compiled programs can run without eclipse
- Kermeta compiled code dependencies (3,5 Mo)
  - org.eclipse.emf.common_2.4.0.v200808251517.jar
  - org.eclipse.emf.ecore.xmi_2.4.1.v200808251517.jar
  - org.eclipse.emf.ecore_2.4.1.v200808251517.jar"
  - org.eclipse.core.resources_3.4.0.v20080604-1400.jar"
  - org.eclipse.core.runtime_3.4.0.v20080512.jar"
  - org.eclipse.emf.codegen_2.4.0.v200808251517.jar"
  - org.eclipse.emf.ecoretools.registration_1.0.0.jar"
  - org.eclipse.equinox.registry_3.4.0.v20080516-0950.jar"
  - fr.irisa.triskell.eclipse.util_1.2.0.jar"
  - org.eclipse.equinox.common_3.4.0.v20080421-2006.jar"
  - org.eclipse.core.jobs_3.4.0.v20080512.jar"
  - fr.irisa.triskell.kermeta.model_1.2.0.jar"
  - org.eclipse.osgi_3.4.0.v20080605-1900.jar
- Some can be removed (<2Mo)
  - org.eclipse.core.resources_3.4.0.v20080604-1400.jar"
  - org.eclipse.core.runtime_3.4.0.v20080512.jar"
  - org.eclipse.osgi_3.4.0.v20080605-1900.jar
  - org.eclipse.core.jobs_3.4.0.v20080512.jar"

Kermeta compiler

- A compiler for integrating several languages
  - A compiler for OCL,
  - Ecore,
  - some parts of Alloy

New feature under development / test for the compiler

- Integration with Legacy
  - AspectJ generation
- POJO generation

New feature under development / test

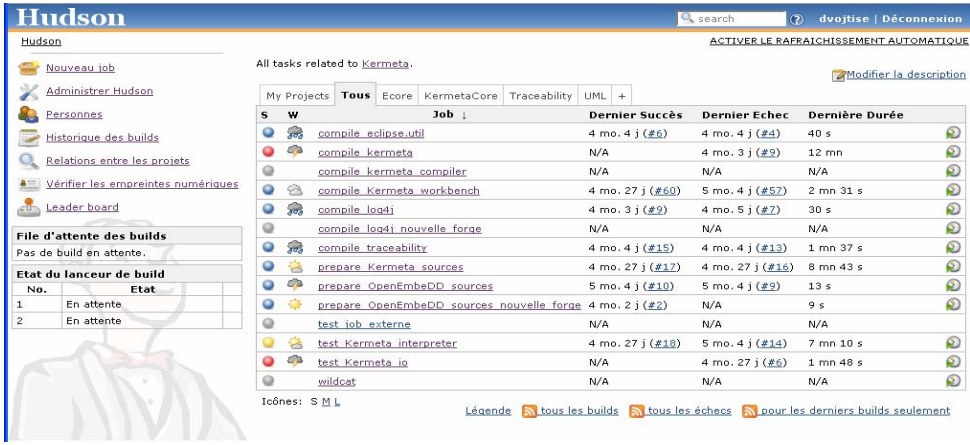
- Incremental model checking
- Generalize the support of ecore generated from xml schema (opens a lot of interesting use cases)
- Model Type consolidation (rename)
- Metamodel weaving of .ocl files (currently still need to be converted first by the user)
- More documentation (Manual, tutorial and cookbook)
- Consolidate existing MDKs !
  - Integration of more MDKs in the nightly build and tests
  - Systematically use the team know how for integration and deployment (samples, documentation, ...)
- Model prototyper/simulator (in connexion with editors)

Future of Kermeta

- Improve implementation and design issues of Kermeta itself (thanks to aspects !)
- Visible part for the end user :
  - Split Kermeta (and associated tools) into reusable DSL fragments in order to build new DSL
    - Do not need to embed all Kermeta into your MM just to reuse a part of Kermeta behavior !

Improve community collaborative efforts

- Offer simpler integration process of all MDKs built with Kermeta
  - Thanks to Hudson



- Vulgarization article in PROgrammez! Le magazine du développement

Available MDKs

- Available (will be deployed shortly)
  - Kermeta
  - Kompose
  - Ecore
  - Java5
  - Traceability
  - UML2
    - Splitted for better readability / separation of concerns
  - Ket (templates)
  - RDL (requirement langage)
- Under dev
  - Docbook, Pattern Matching, KTR (transformation rules), Trek (test management), …

Conclusion

- Help us to give priorities to the development
  - Any suggestion is welcome
  - Please feel free to add feature requirements
- Download the V1.3.0 bundle or install it in your current eclipse from :
  - **http://www.kermeta.org**