

Using Model-to-Text Transformation for Dynamic Web-based Model Navigation

Dimitris Kolovos, Louis Rose, James Williams

Department of Computer Science,
University of York

Motivation

- Modelling is a collaborative process
- Models need to be shared among stakeholders
 - Different backgrounds and levels of technical expertise
- Many of the stakeholders need read-only access to the models
 - To provide feedback
- How do we do this currently?

Use a Common Modelling Tool

- All stakeholders use a common modelling tool and exchange models using email, version control management systems, dropbox etc.
- Pros
 - Straightforward for technical users
- Cons
 - !(Straightforward) for !(technical) users
 - Requires buying/installing the modelling tool
 - ... all sorts of trouble

Generate Documentation

- Automatically generate documentation from models
 - In Word, HTML, LaTeX etc.
- Send the generated documentation around using email, Dropbox, the web etc.
- Pros
 - No need for additional tools
- Cons
 - Documentation always lagging behind
 - Documents can get huge
 - Documents only contain fixed views of the model

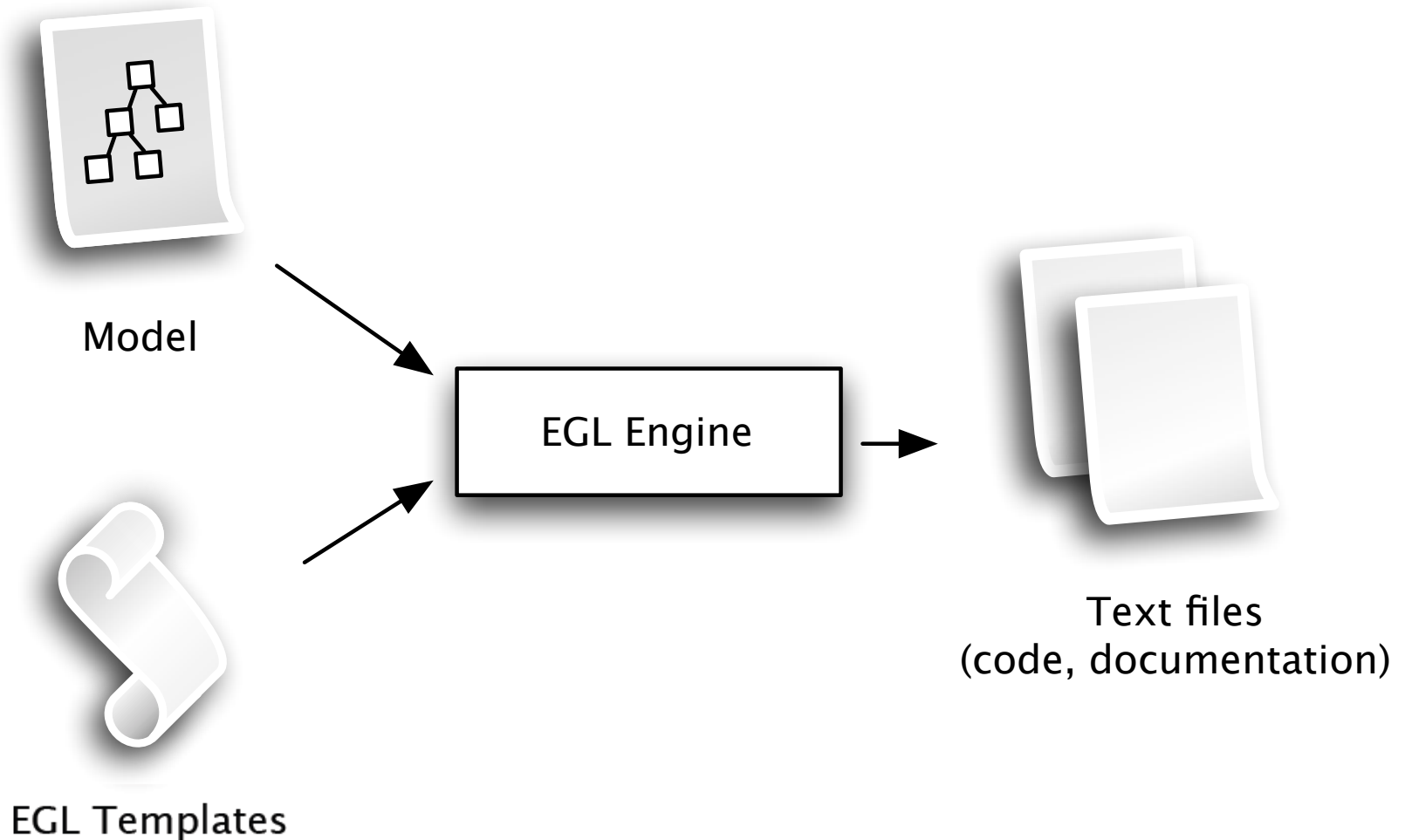
Requirements

- No additional tooling for stakeholders
 - Accessible from their web-browsers
- Stakeholders always have access to the most recent version
- No sending huge files around
- Dynamic views

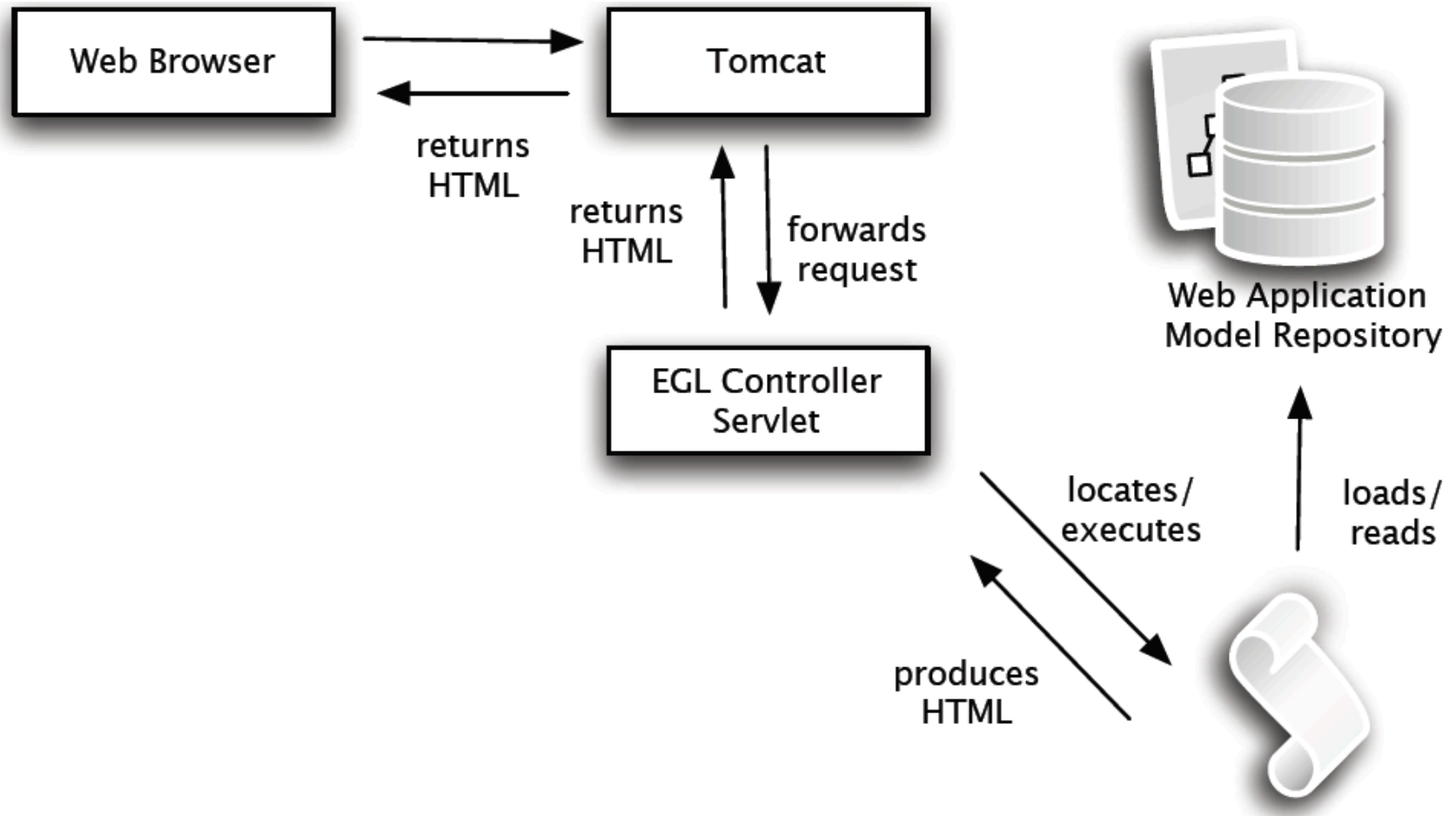
Dynamic Model Navigation

- Build web-based model-browsing applications that draw data directly from the models
- Prototype using
 - EMF: for model representation
 - Tomcat: for web-application hosting
 - EGL: template-based M2T language used as a server-side scripting language

EGL as an M2T Language



EGL as a Server-Side Language



Caching

- Improve performance of web-applications
 - Reduce the need for re-computation
- Page caching
 - Cache whole pages
- Fragment caching
 - Cache specific pieces of response text that are expensive to compute

DEMO

Conclusions

- Alternative option for sharing models
 - No additional tools / training
 - No need to send documentation around
 - Web-based access
 - Documentation always in sync with models
- Prototype: read-only access to models
- Not appropriate for huge models / large numbers of users
 - Memory-resident models not scalable
 - Database-based representations can help with this (e.g. CDO)

Resources

- Installation instructions
 - <http://eclipse.org/gmt/epsilon/doc/articles/egl-server-side/>
- EGL documentation
 - <http://eclipse.org/gmt/epsilon/doc/egl/>

