http://www.isis.vanderbilt.edu/projects/gme

Model-Integrated Computing in the Large

Distributed/Multiuser Access and Version Control in GME

Akos Ledeczi

Senior Research Scientist

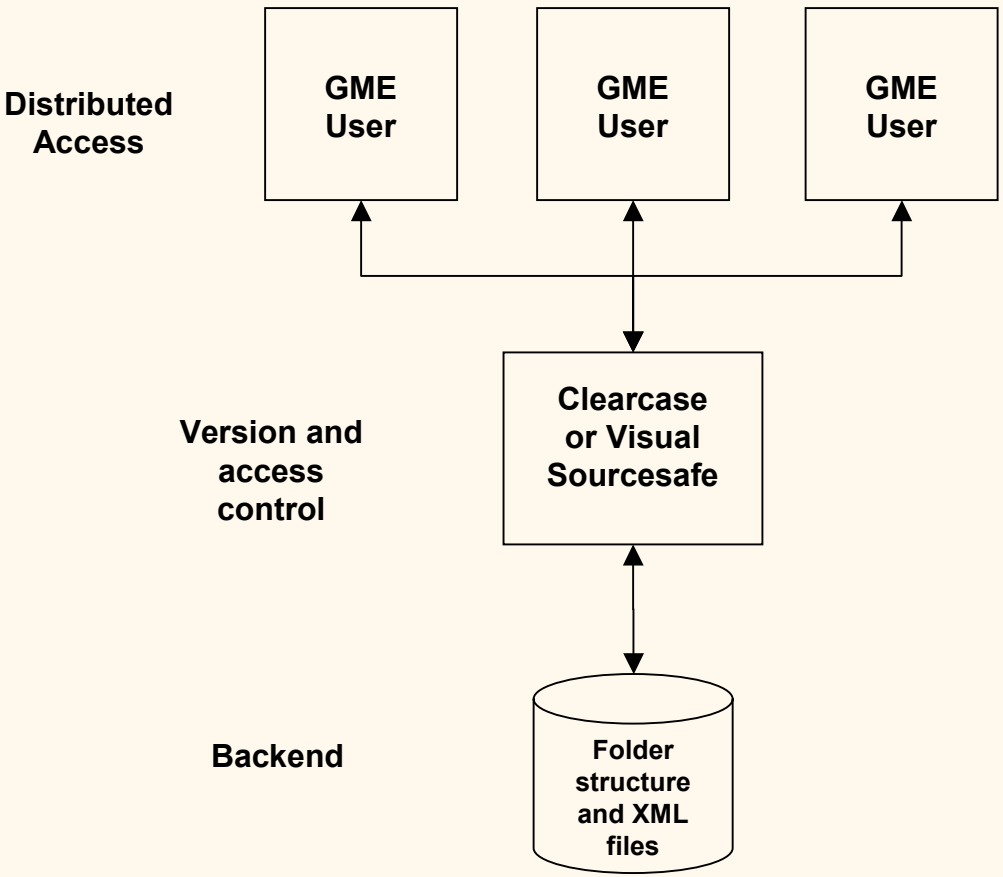
Institute for Software Integrated Systems

Vanderbilt University

Overview

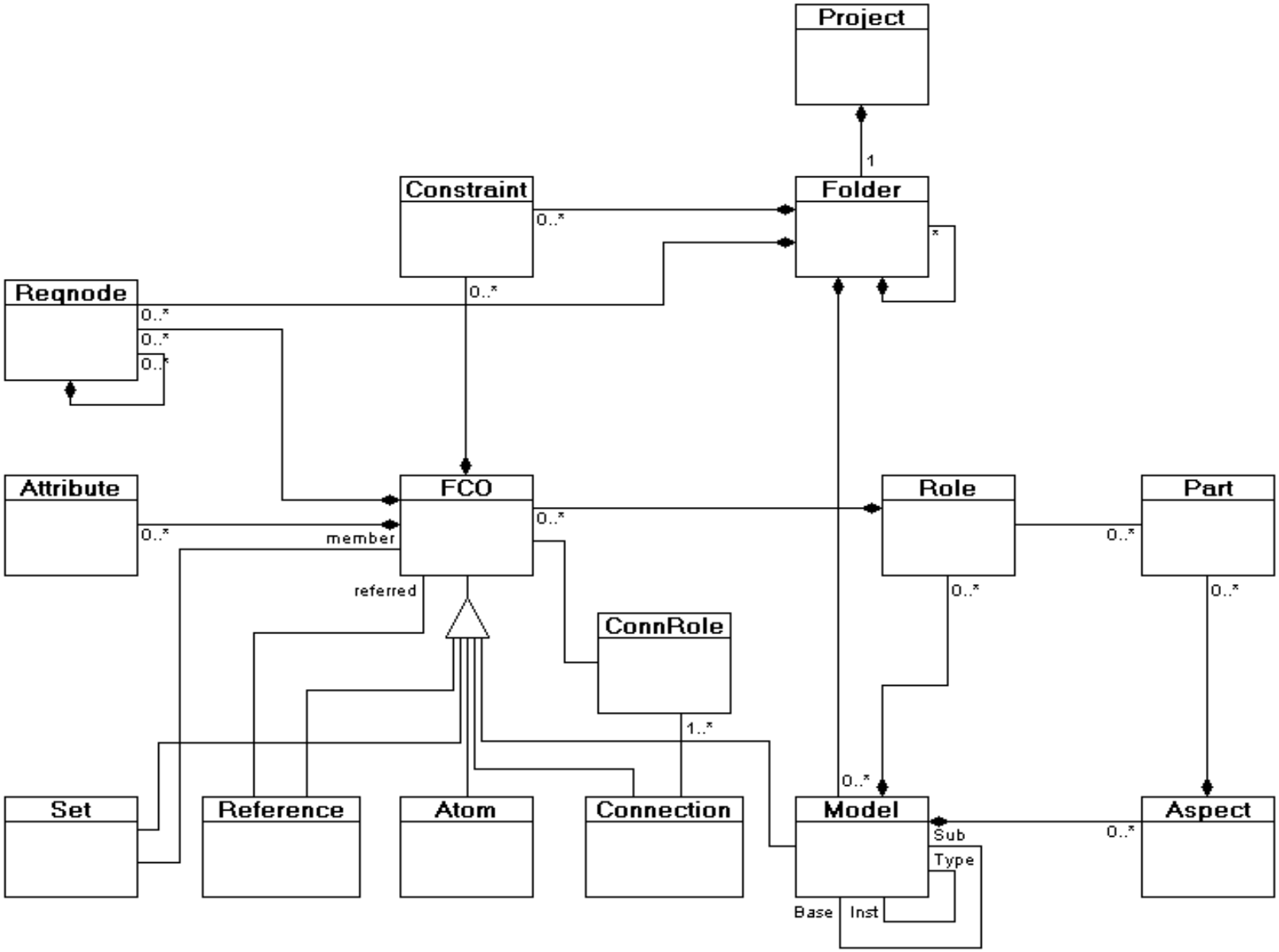
- Model-based vs. traditional development
- Multi-person teams, large code base and model base
- Needs tool support:
  - Distributed access to the models by all developers
  - Model versioning and control
- Solution: utilize existing source code control tools
- Issue: models have much richer interdependencies than source code that need to be kept in a consistent state

Multiuser access and model versioning



- Fine grain xml backend enabling:
  - Scalability (only required models need to be read in memory)
  - Tool interoperability
  - Readability
- Version control tool enabling:
  - Versioning
  - Locking for distributed access

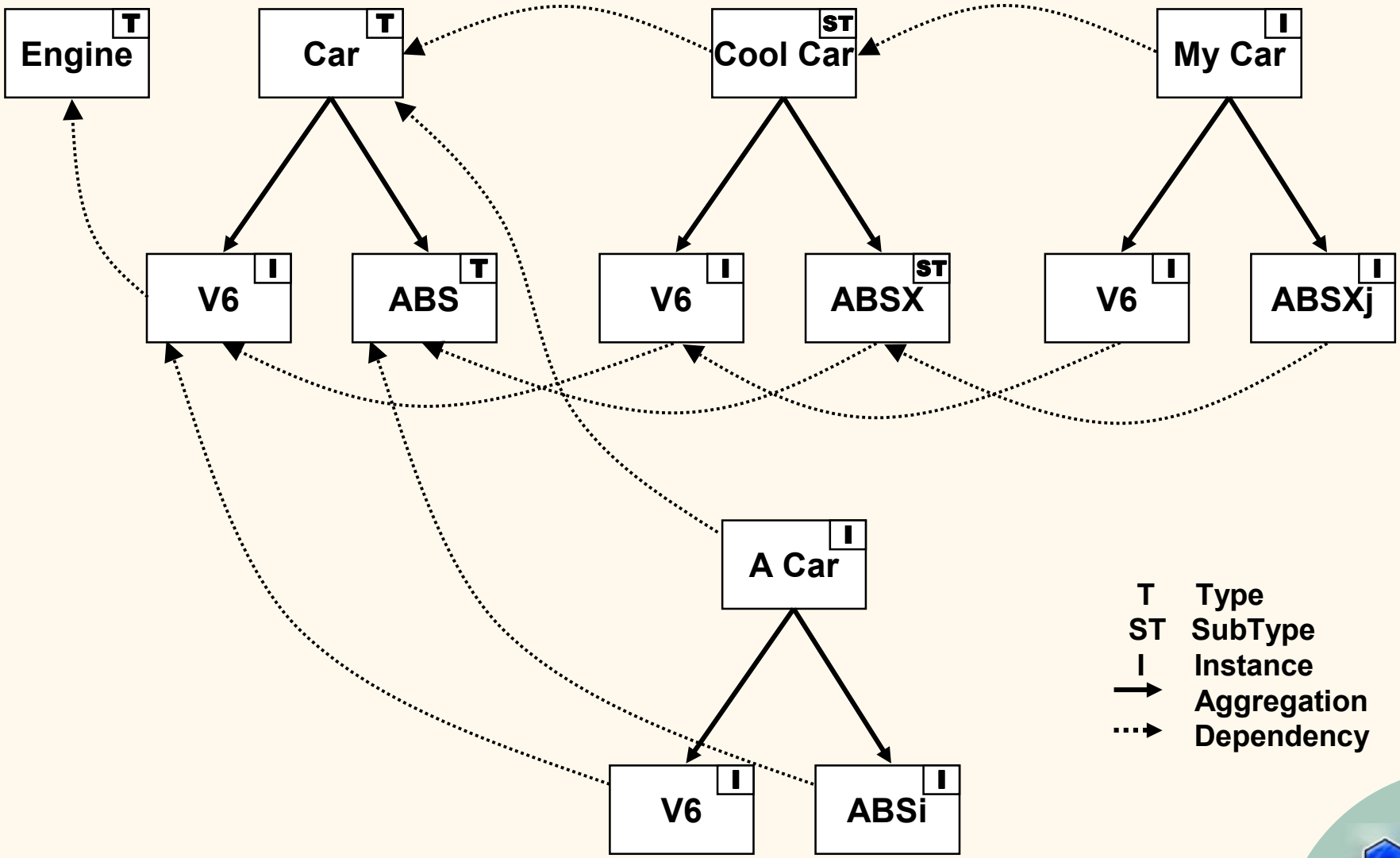
GME Concepts



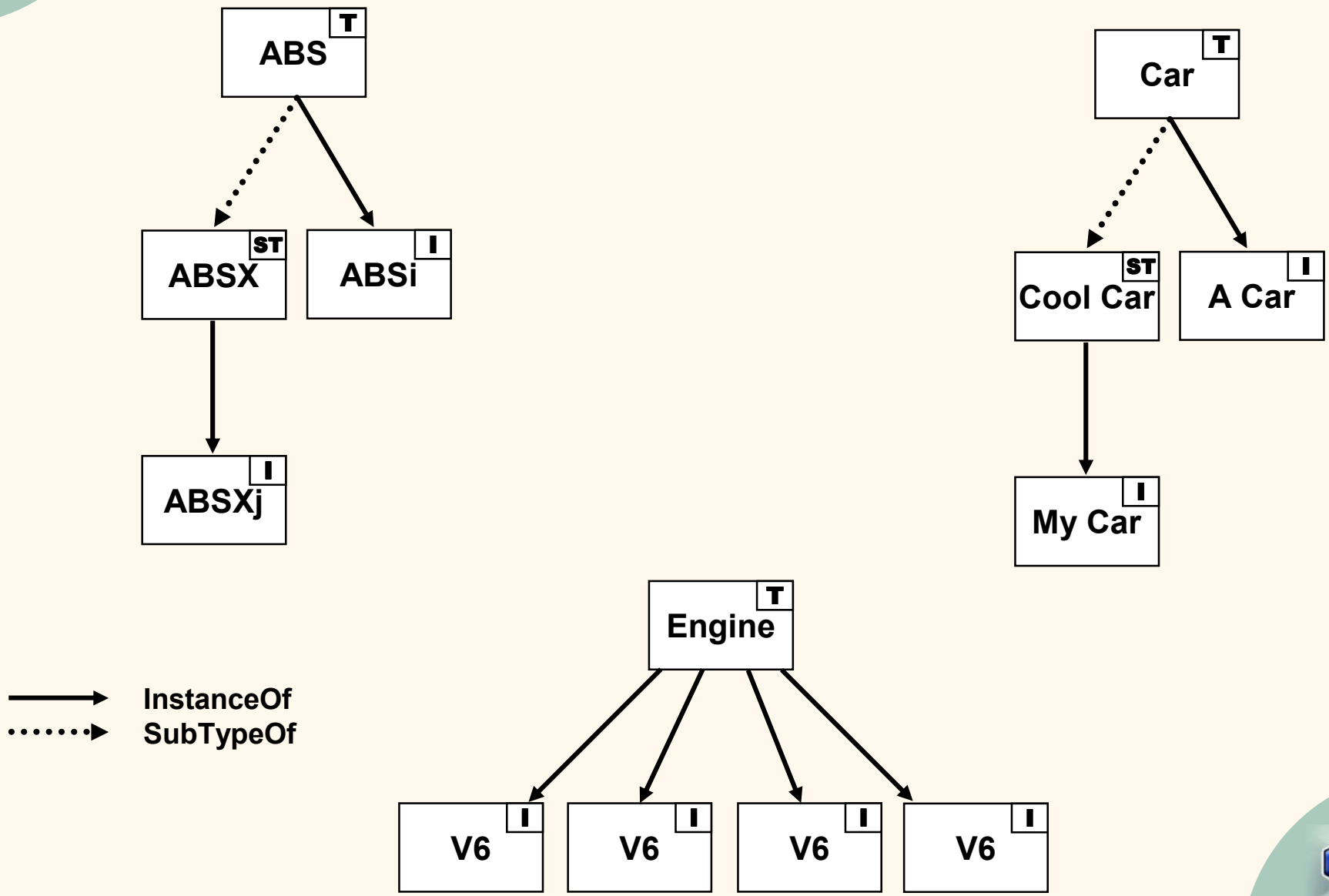
Type Inheritance

- Type, Subtype, Instance
- A model created from scratch is a type
- Explicit and implicit derivation/instantiation
- A deep copy preserving the linkage
- In subtypes:
  - New objects can be added
  - Attributes can be modified
  - Objects cannot be deleted
- In instances:
  - Attributes can be modified
- Composition and type inheritance raises interesting questions: Only root models can be explicitly derived or instantiated to avoid multiple dependency chains
- Similar to prototype/clone technique
- Enables model reuse, maintainability
- Utilized in libraries

Type Inheritance (dependency chain)



Type Inheritance (type/instance hierarchy)



Operation

- Each folder and model has its own xml file on the server
- All files have local copies
- Cache file storing all relationships locally: makes it possible not to lock “parent” object of relationships
- Objects are not locked until user attempts to modify them, i.e. browsing is transparent
- Upon a modification attempt the model is checked out from the server.
- The following relationships are followed recursively to check out the target objects also:
  - All children (and consequently all descendants) in the containment hierarchy
  - All references are followed in the reference chain all the way to the final target
  - All derived types and instances are followed in the inheritance hierarchy (e.g. if you edit a basetype, you cannot modify its dependent objects)
- If checkout fails, modification is not allowed
- Objects remain checked out until user explicitly checks them in (by saving the project)

Consistency

- Checked in models are consistent at all times
- All available previous versions are consistent also
- Getting portions of the database from different versions is not supported
- Strict locking policy may block access to a significant portion of the model database (paradigm-, and project-dependent and also greatly influenced by which model is being modified)
- Usability remains to be seen:
  - How many users can work simultaneously in a productive manner
  - To add a new file to a Clearcase database seems to take one second: if you copy a whole subtree of models this can be very slow (albeit only at checkin time, hence, only saving is a slow operation). Sourcesafe does not have this issue.
- Alternative approach: allow inconsistencies and have a compiler-like tool resolve them periodically as requested by users. This would also enable mixing/matching portions of the models from different versions/dates.

Closure and Smart Copy

- Another new feature in the works helping in the development of large/complex projects
- Simple copy/paste only gets the objects explicitly selected. None of the relationships are followed, connections, references, inheritance etc. are all broken.
- Closure: follow user specified relationships in a recursive manner.
- Smart copy: pastes objects while trying to identify identical objects in the target project and resolve references/connections etc. to existing objects instead of copying the new identical object.
- Very useful when
  - Source and target use the same library
  - Copying portions of a project to create a subset
  - Etc.