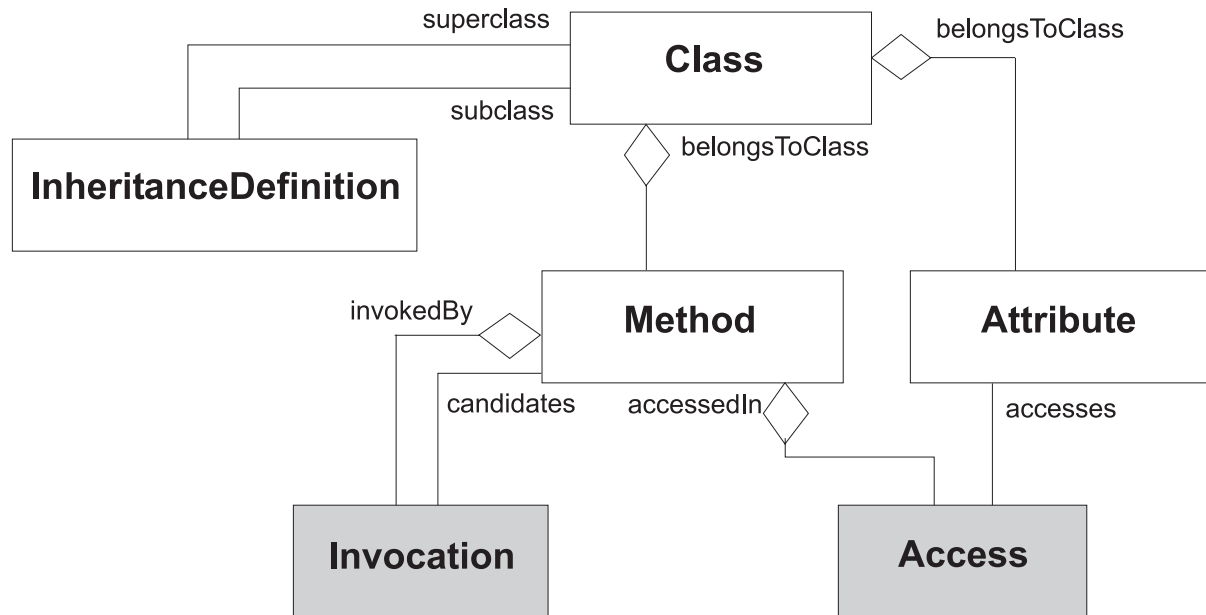


- Devise a meaningful yet flexible DB schema in order to support the extensions to the meta-model decided by the other groups
 - Extensions:
 - New FAMIX classes?
 - New attributes to existing classes?
 - Proposed extensions are...
 - Authors
 - Bug tracker information
 - Log messages
- Choose the appropriate database model
 - Relational database?
 - Document-oriented database?
- Prepare a means to store the data properly and make hypothesis on the metrics to extract.
- What to store?
 - Only metrics
 - Entire model
 - Part of the model (hierarchy) + metrics
 - Changes at FAMIX class level
- How to model the changes?
 - Addition
 - Deletion
 - Change (addition + deletion)
- FAMIX constructs + genesis extensions
- Technologies?

Visualizations & metrics

- BM Bug visualization
 - List of bugs per project
 - Status of each bug
 - Assignee
 - Description
 - History of bug (i.e., to see previous assignees)
- BM Ownership sunburst
 - File owners (i.e., w/ ownership relation)
- EP Developer responsibilities radar
 - Entity owners (i.e., w/ ownership relation)
- EP Bug prediction & coupling
 - Coupling between classes (i.e., how defined?)
 - Prominence?
- LZ Change influence
 - History of code elements (i.e., revision)
 - Inheritance tree
 - Which metrics?
- LZ Generic bubbles
 - Element of interest
 - Code element
 - Developer
 - ...
 - Correlation?

What can be useful to store?



Extensions to the model

- Bug tracker information
- Author information

Abstract classes

- Object
 - sourceAnchor
 - commentsAt
- Entity [ext. Object]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - name
 - uniqueName
- Association [ext. Object]
 - *Inherited*
 - sourceAnchor
 - commentsAt
- Property [Obj \diamond -- has some]
 - name
 - value
- BehaviouralEntity [ext. Entity]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - name
 - uniqueName
 - accessControlQualifier
 - signature
 - isPureAccessor
 - declaredReturnType
 - declaredReturnClass
- StructuralEntity [ext. Entity]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - name
 - uniqueName
 - declaredType
 - declaredClass
- Argument
 - position
 - isReceiver

Concrete classes

- Model [ext. Object]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - exporterName
 - exporterVersion
 - exporterDate
 - exporterTime
 - publisherName
 - parsedSystemName
 - extractionLevel
 - sourceLanguage
 - sourceDialect
- Package [ext. Entity]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - name
 - uniqueName
 - belongsToPackage
- **Class** [ext. Entity]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - name
 - uniqueName
 - isAbstract
 - belongsToPackage
- **Method** [ext. BehaviouralEntity]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - name
 - uniqueName
 - accessControlQualifier
 - signature
 - isPureAccessor
 - declaredReturnType
 - declaredReturnClass
 - belongsToClass
 - hasClassScope

- isAbstract
- isConstructor
- Function [ext. BehaviouralEntity]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - name
 - uniqueName
 - accessControlQualifier
 - signature
 - isPureAccessor
 - declaredReturnType
 - declaredReturnClass
 - belongsToPackage
- **Attribute** [ext. StructuralEntity]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - name
 - uniqueName
 - declaredType
 - declaredClass
 - belongsToClass
 - accessControlQualifier
 - hasClassScope
- GlobalVariable [ext. StructuralEntity]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - name
 - uniqueName
 - declaredType
 - declaredClass
 - belongsToPackage
- ImplicitVariable [ext. StructuralEntity]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - name
 - uniqueName
 - declaredType
 - declaredClass

- belongsToContext
- LocalVariable [ext. StructuralEntity]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - name
 - uniqueName
 - declaredType
 - declaredClass
 - belongsToBehaviour
- FormalParameter [ext. StructuralEntity]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - name
 - uniqueName
 - declaredType
 - declaredClass
 - belongsToBehaviour
 - position
- **InheritanceDefinition** [ext. Association]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - subclass
 - superclass
 - accessControlQualifier
 - index
- **Access** [ext. Association]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - access
 - accessedIn
 - isAccessLValue
- **Invocation** [ext. Association]
 - *Inherited*
 - sourceAnchor
 - commentsAt
 - invokedBy
 - invokes

- base
- candidatesAt
- ExpressionArgument [ext. Argument]
 - *Inherited*
 - position
 - isReceiver
- AccessArgument [ext. Argument]
 - *Inherited*
 - position
 - isReceiver

Concrete classes (pruned)

- **Model** [ext. Object]
 - parsedSystemName
 - (extractionLevel)
 - sourceLanguage
 - sourceDialect
- **Package** [ext. Entity]
 - **owner**
 - name
 - uniqueName
 - belongsToPackage
- **Class** [ext. Entity]
 - **owner**
 - name
 - uniqueName
 - (isAbstract)
 - belongsToPackage
- **Method** [ext. BehaviouralEntity]
 - **owner**
 - name
 - uniqueName
 - accessControlQualifier
 - signature
 - declaredReturnType
 - declaredReturnClass
 - belongsToClass
 - (hasClassScope)
 - (isAbstract)
 - (isConstructor)
- **Function** [ext. BehaviouralEntity]
 - **owner**
 - name
 - uniqueName
 - accessControlQualifier
 - signature
 - declaredReturnType
 - declaredReturnClass
 - belongsToPackage
- **Attribute** [ext. StructuralEntity]
 - name

- uniqueName
 - declaredType
 - declaredClass
 - belongsToClass
 - accessControlQualifier
 - (hasClassScope)
- GlobalVariable [ext. StructuralEntity]
 - commentsAt
 - name
 - uniqueName
 - declaredType
 - declaredClass
 - belongsToPackage
- ImplicitVariable [ext. StructuralEntity]
 - name
 - uniqueName
 - declaredType
 - declaredClass
 - belongsToContext
- LocalVariable [ext. StructuralEntity]
 - name
 - uniqueName
 - declaredType
 - declaredClass
 - belongsToBehaviour
- FormalParameter [ext. StructuralEntity]
 - name
 - uniqueName
 - declaredType
 - declaredClass
 - belongsToBehaviour
- **InheritanceDefinition** [ext. Association]
 - subclass
 - superclass
 - accessControlQualifier
 - (index)
- **Access** [ext. Association]
 - access
 - accessedIn
 - (isAccessLValue)

- **Invocation** [ext. Association]
 - invokedBy
 - invokes
 - (base)
 - candidatesAt
- ExpressionArgument [ext. Argument]
 - position
 - isReceiver
- AccessArgument [ext. Argument]
 - position
 - isReceiver

Extensions to the meta-model

- Bug-tracker information
 - belongsToProject
 - name
 - description
 - status
 - assignee
- Bug-tracker history
 - uniqueName
 - previousAssignees
- Developer
 - name
 - worksOnProject

Concerning the metrics...

We categorized metrics into...

- (1) Metrics that must be stored (cannot be recovered from what is stored into DB)
 - Revision metrics
 -
 - Method metrics
 - MLOC
 - Class metrics
 - LOC (or computed summing up MLOC)
 - Package metrics?
- (2) Metrics that are computationally intensive to compute (i.e., probably it's better pre-compute them and cache them)
 - Coupling?
- (3) Metrics obtainable by querying the DB (not stored at all!)
 - NOM
 - Weighted Methods per Class
 - NOA (NOF)
 - NOC
 - Cohesion