

# sparrow-api ontology data model, mediator, and management

Preliminary Draft, June 2004

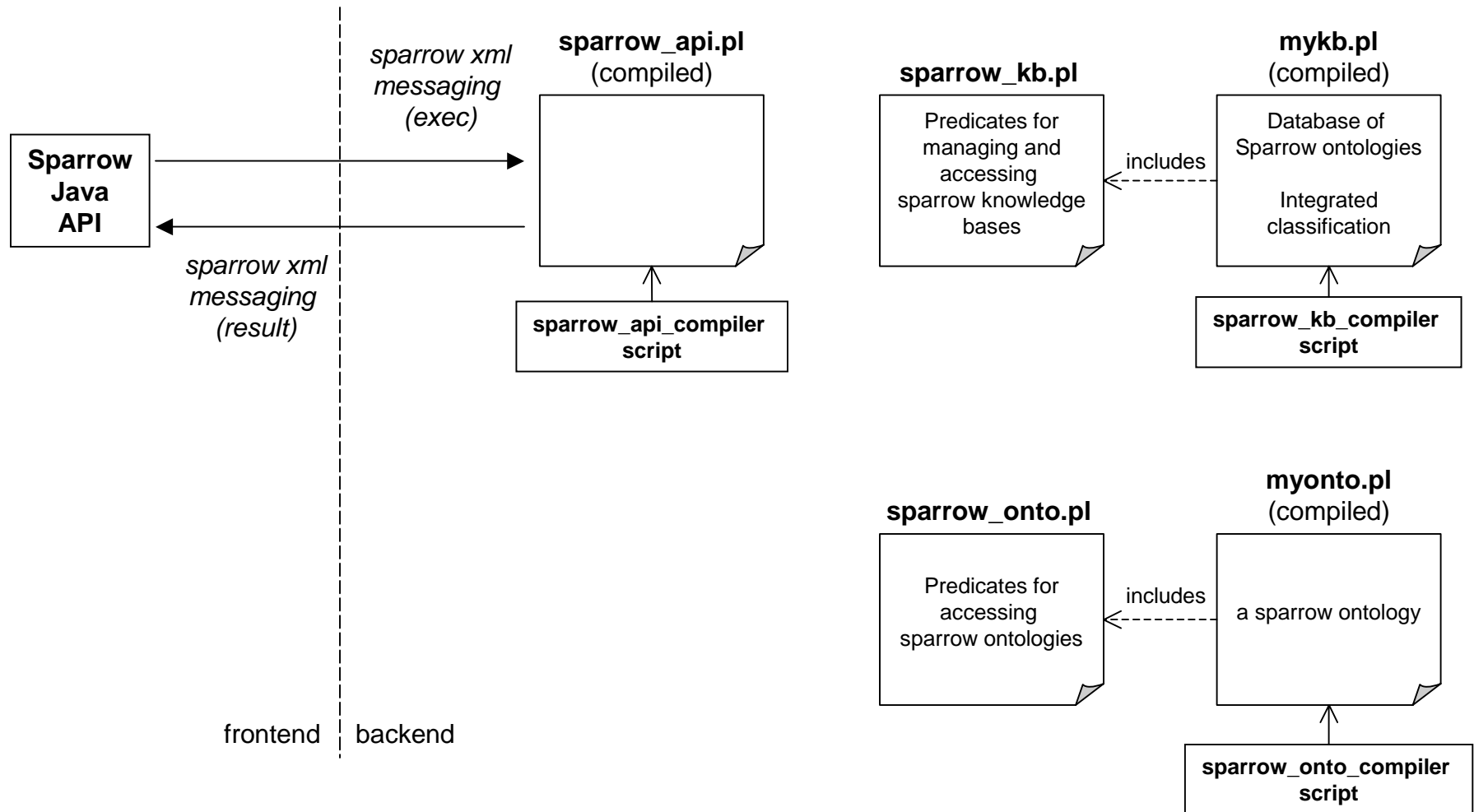
by Shawn Bowers  
(bowers at sdsc.edu)

# sparrow java api [preliminary, minimal draft]

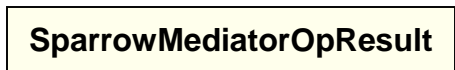
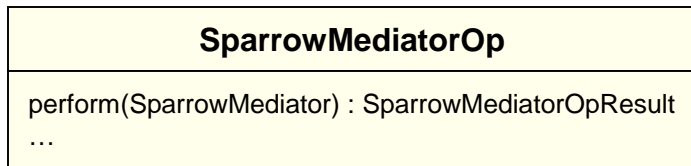
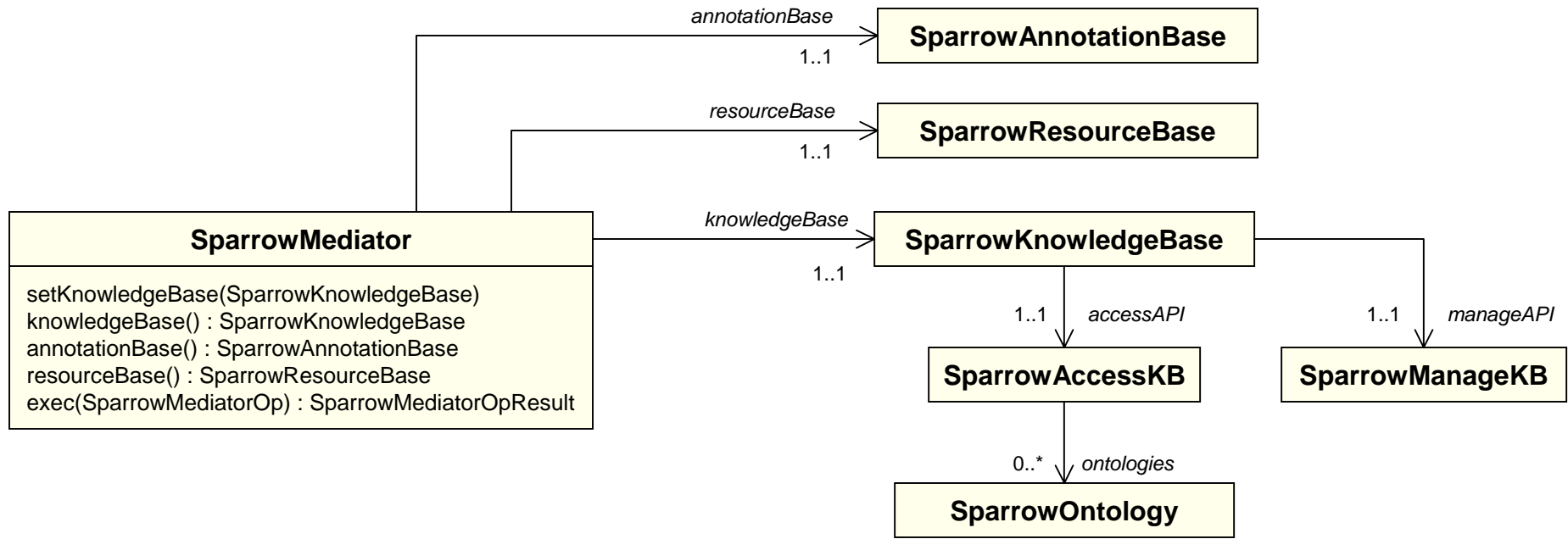
## Package naming

- org.ecoinformatics.seek.sparrow
  - datamodel
  - mediator
  - knowledgebase
  - annotationbase
  - resourcebase
  - util

# sparrow prolog interface



# sparrow java api [preliminary, minimal draft]



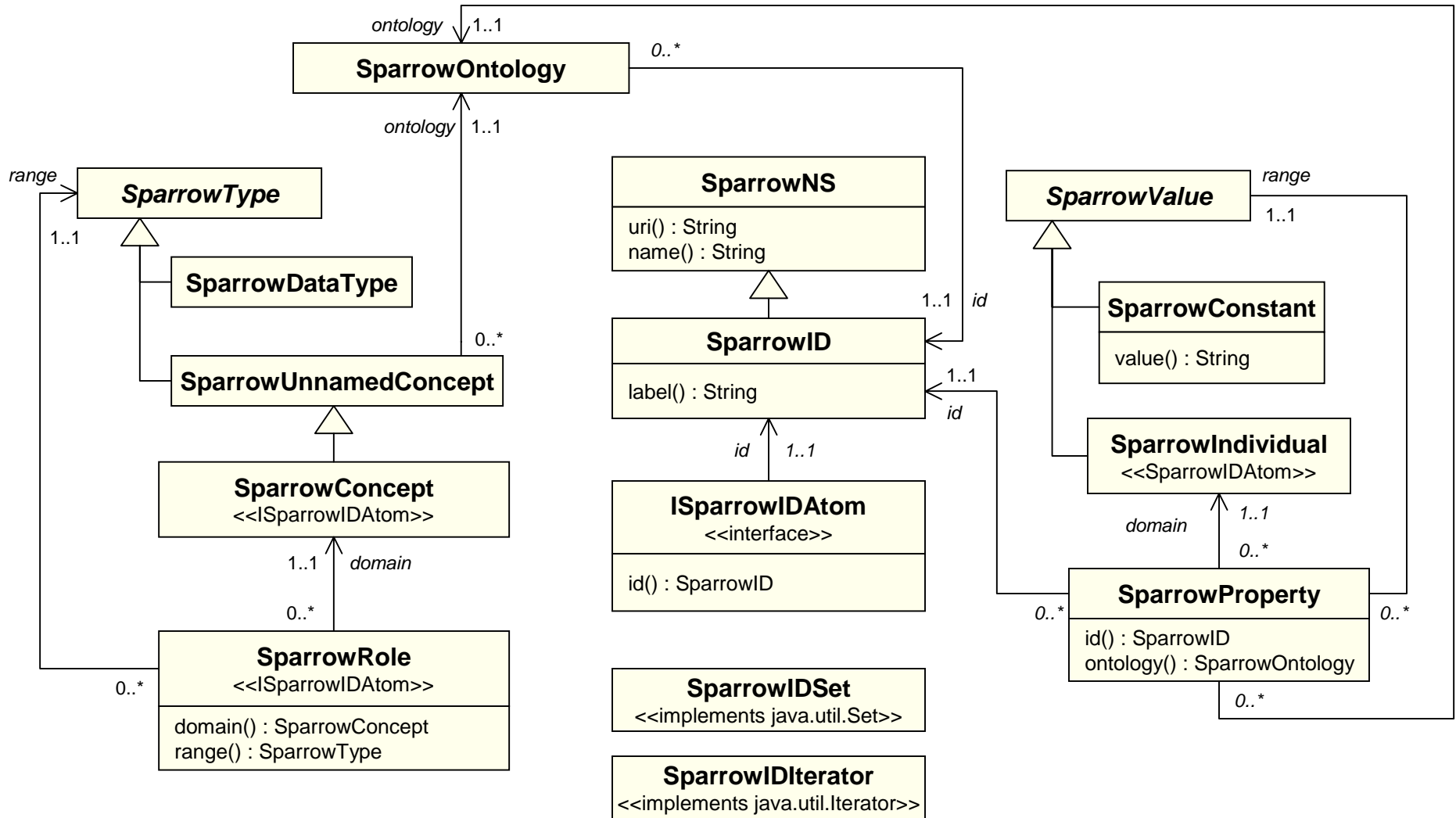
Operations pass control of execution to the mediator, and allow for easy addition of new operations. The perform method available at the package level only

The resource base should be an index structure, mapping dataset and service resource identifiers to annotations (semantic descriptions)

Note that we should separate the resource base into data and service resources.

Resources should have SparrowIDs, as should annotations

# sparrow java api [preliminary, minimal draft]



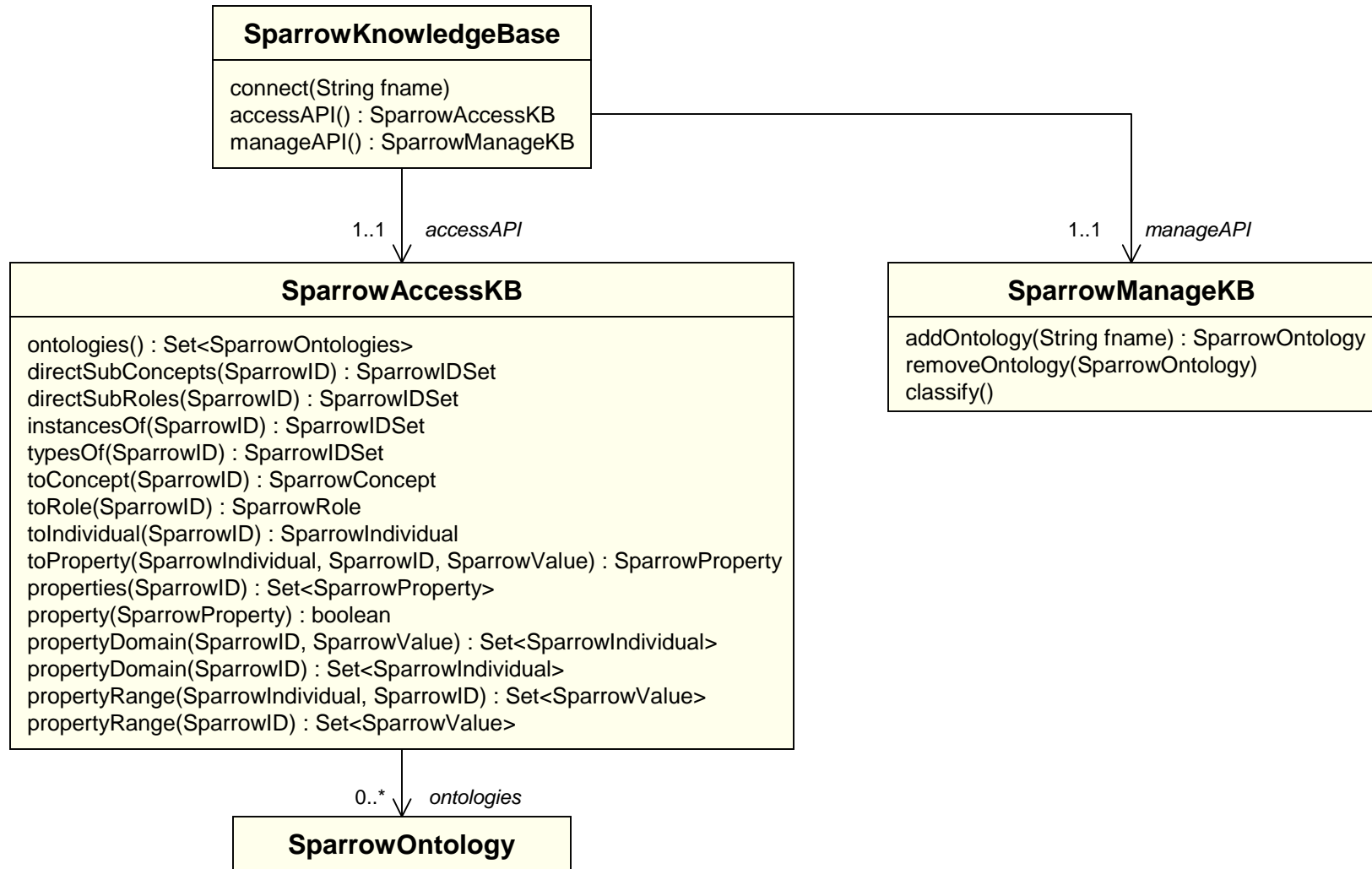
NOTES: Sparrow Concepts, Roles, Individuals, and Properties do not have public constructors. Instead, build a new, plain SparrowID object o, and call KnowledgeBase.toConcept(o), etc. Identifier, Concept, Role, etc. objects are not unique ... however, the value of an id is unique, i.e., the triple (uri, name, label) forms a key.

# sparrow java api [preliminary, minimal draft]

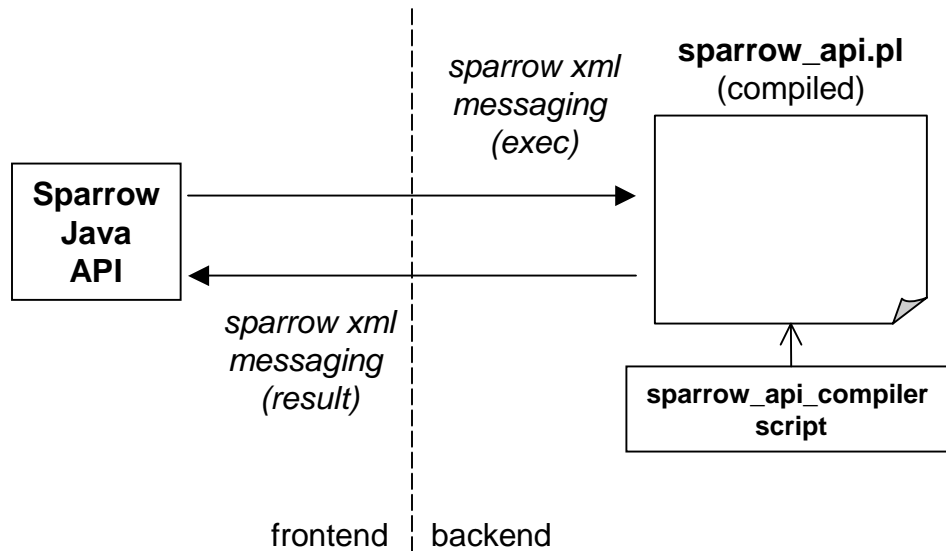
<b>SparrowOntology</b>
id() : SparrowID concepts() : SparrowIDSet roles() : SparrowIDSet individuals() : SparrowIDSet properties() : SparrowIDSet defines(SparrowID) : boolean uses(SparrowID) : boolean

NOTES: The operation properties() returns the identifiers of all triple predicates used in the ontology. Some of these identifiers may be defined as roles, some may not.

# sparrow java api [preliminary, minimal draft]



# sparrow prolog interface



sparrow\_api\_messaging.dtd

```
<!ELEMENT SparrowExec #PCDATA>
<!ATTLIST SparrowExec
    goal      cdata      implied
    relname  nmtoken    implied
    attlist  [nmtoken list] implied>
<!ELEMENT SparrowResult (SparrowExec, SparrowStats, SparrowResultSet)>
<!ELEMENT SparrowStats (...)>
<!ELEMENT SparrowResultSet ANY>
```

where ANY = ( <%SparrowExec.relname SparrowExec.attlist[1]="val" ... SparrowExec.attlist[n]="val"%> )\*



# dynamic data and actor views using ontologies

