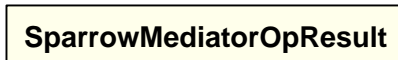
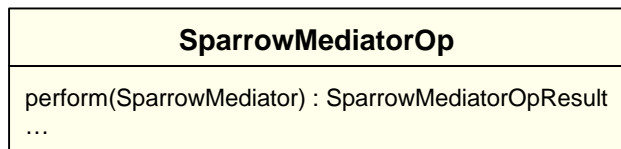
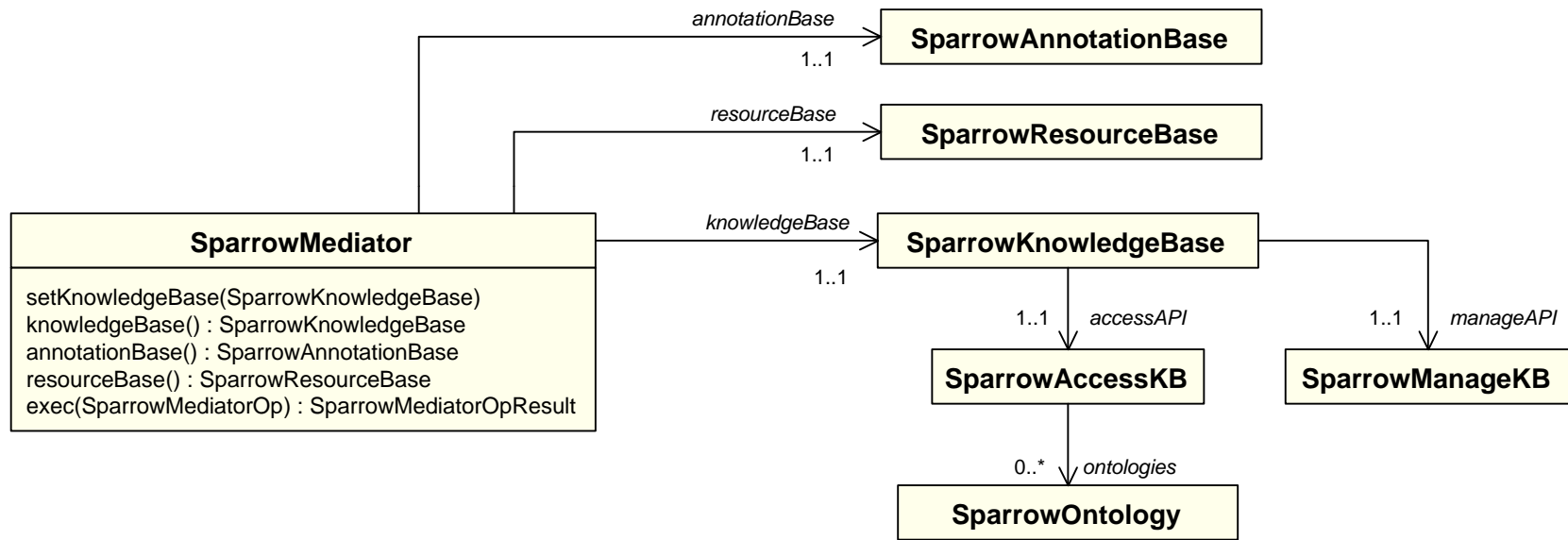


# **sparrow-api: ontology data model and mediator**

Preliminary Draft, June - July 2004

by Shawn Bowers  
([bowers at sdsc.edu](mailto:bowers@sdsc.edu))

# sparrow java api [preliminary, minimal draft]



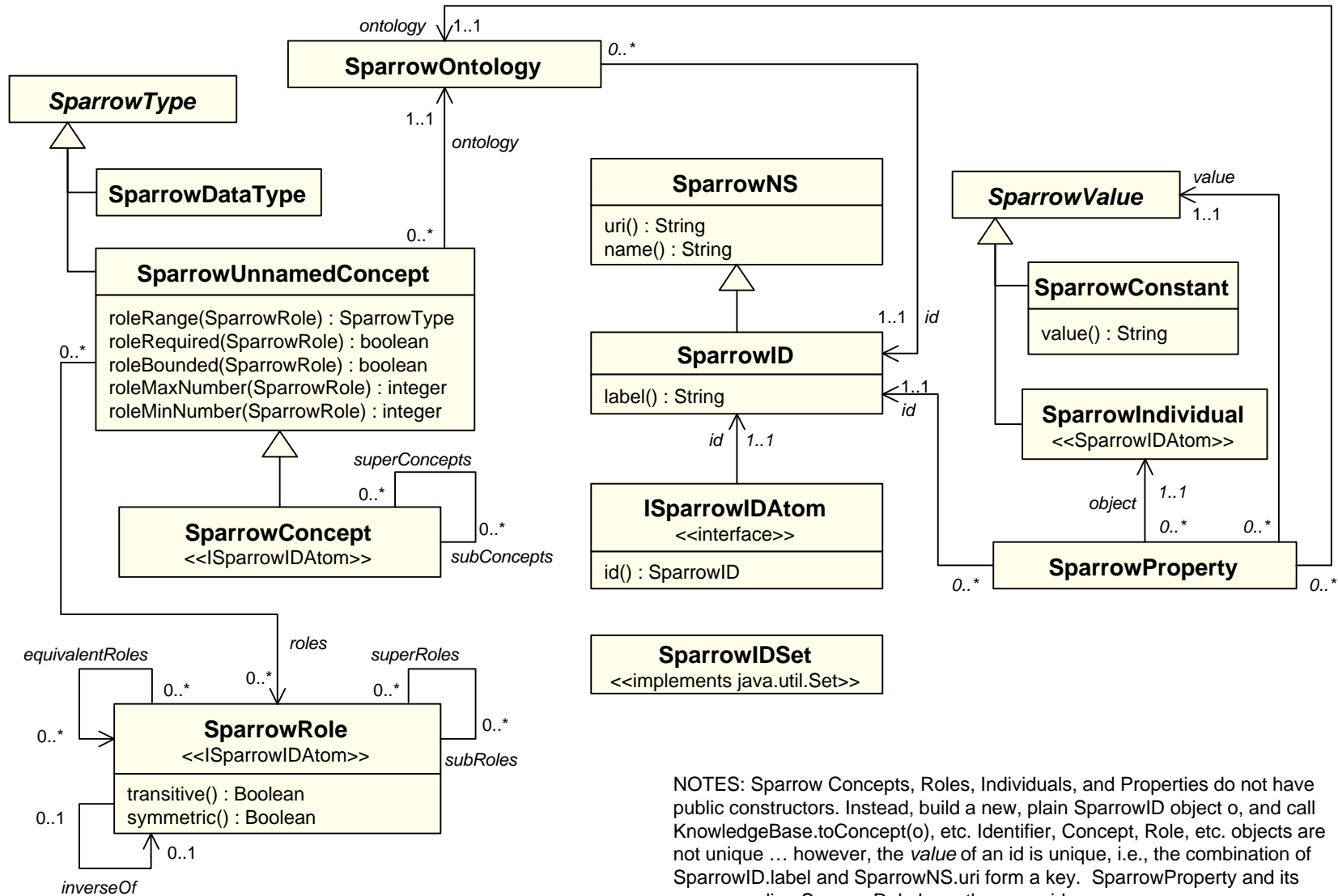
Operations pass control of execution to the mediator, and allow for easy addition of new operations. The perform method is available at the package level only

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.

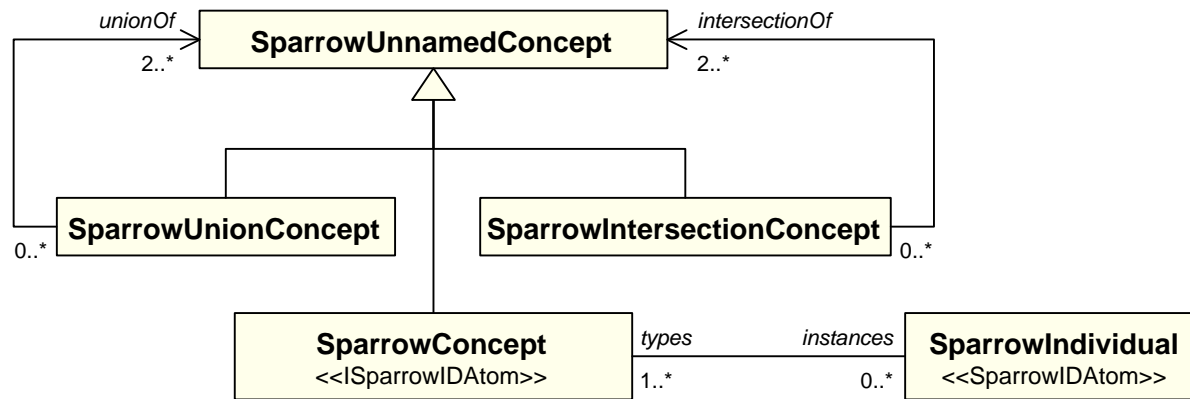
Resources should have SparrowIDs, as should annotations

# sparrow java api [preliminary, minimal draft]

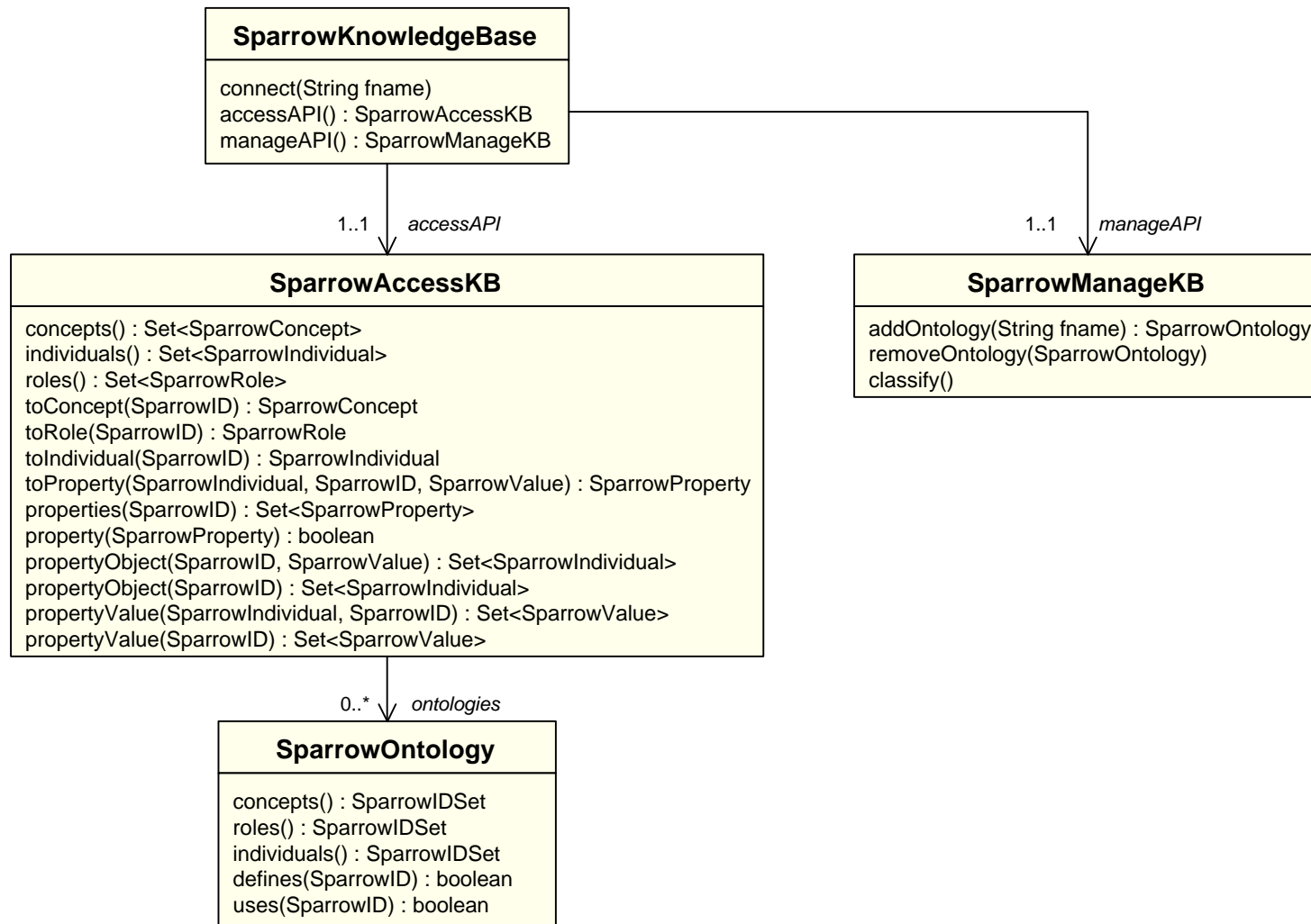


NOTES: Sparrow Concepts, Roles, Individuals, and Properties do not have public constructors. Instead, build a new, plain SparrowID object o, and call KnowledgeBase.toConcept(o), etc. Identifier, Concept, Role, etc. objects are not unique ... however, the value of an id is unique, i.e., the combination of SparrowID.label and SparrowNS.uri form a key. SparrowProperty and its corresponding SparrowRole have the same ids.

# sparrow java api [preliminary, minimal draft]



# sparrow java api [preliminary, minimal draft]



# dynamic data and actor views using ontologies

