ΦLOG: a Domain Specific Language for Phylogenetic Inference

Presented by Yu Pan
Department of Computer Science

ΦLOG: a Domain Specific Language for Phylogenetic Inference

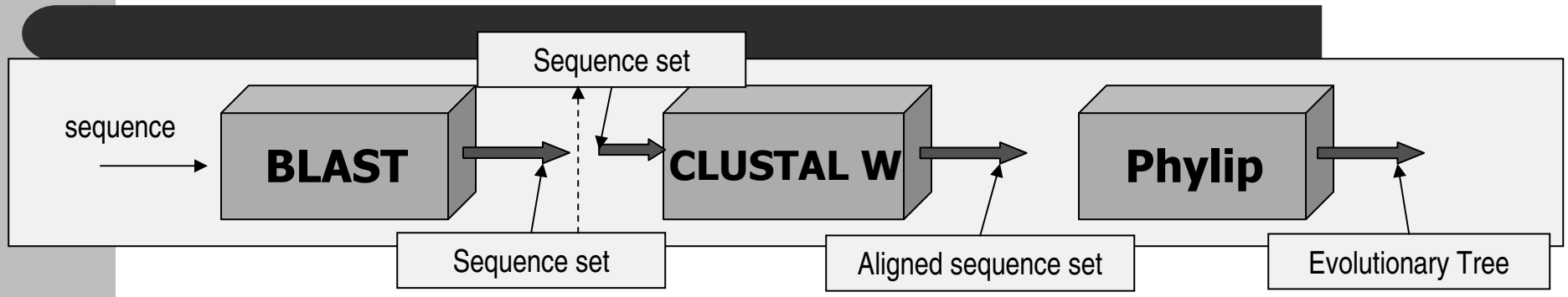
- **Phylogenetic Inference:**

  Estimate the evolutionary history of a collection of organisms (taxa) or family of genes

  - Estimation of the evolutionary tree (database search, multi-sequences align, and phylogenetic tree building).
  - The resulting trees (phylogenies) are used as analytical frameworks for further evolutionary studies

- **Biologists' Approach**
  - "Cut and Paste"
  - Programming Environment (Darwin, Mesquite)

Problem



- Complex & Suboptimal
  - complex glue between applications (data filtering, format transformations, coordination between local and remote applications)
  - fixed set of capabilities provided by existing tools
  - lots of “human tinkering”
- Limits ability to explore alternative models

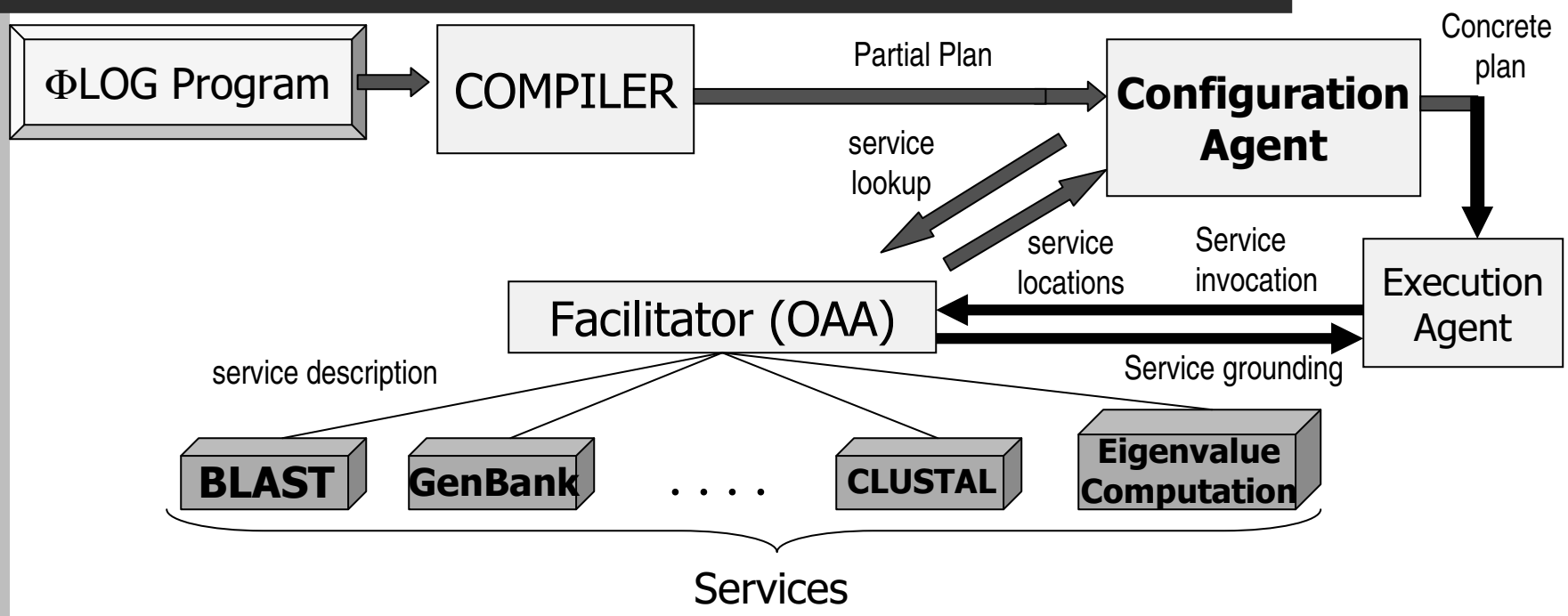
ΦLOG

- Domain Specific Language for Phylogenetic Inference Applications
  - DSL
    - High level description of biological data (NEXUS)
    - High level description of biological processes

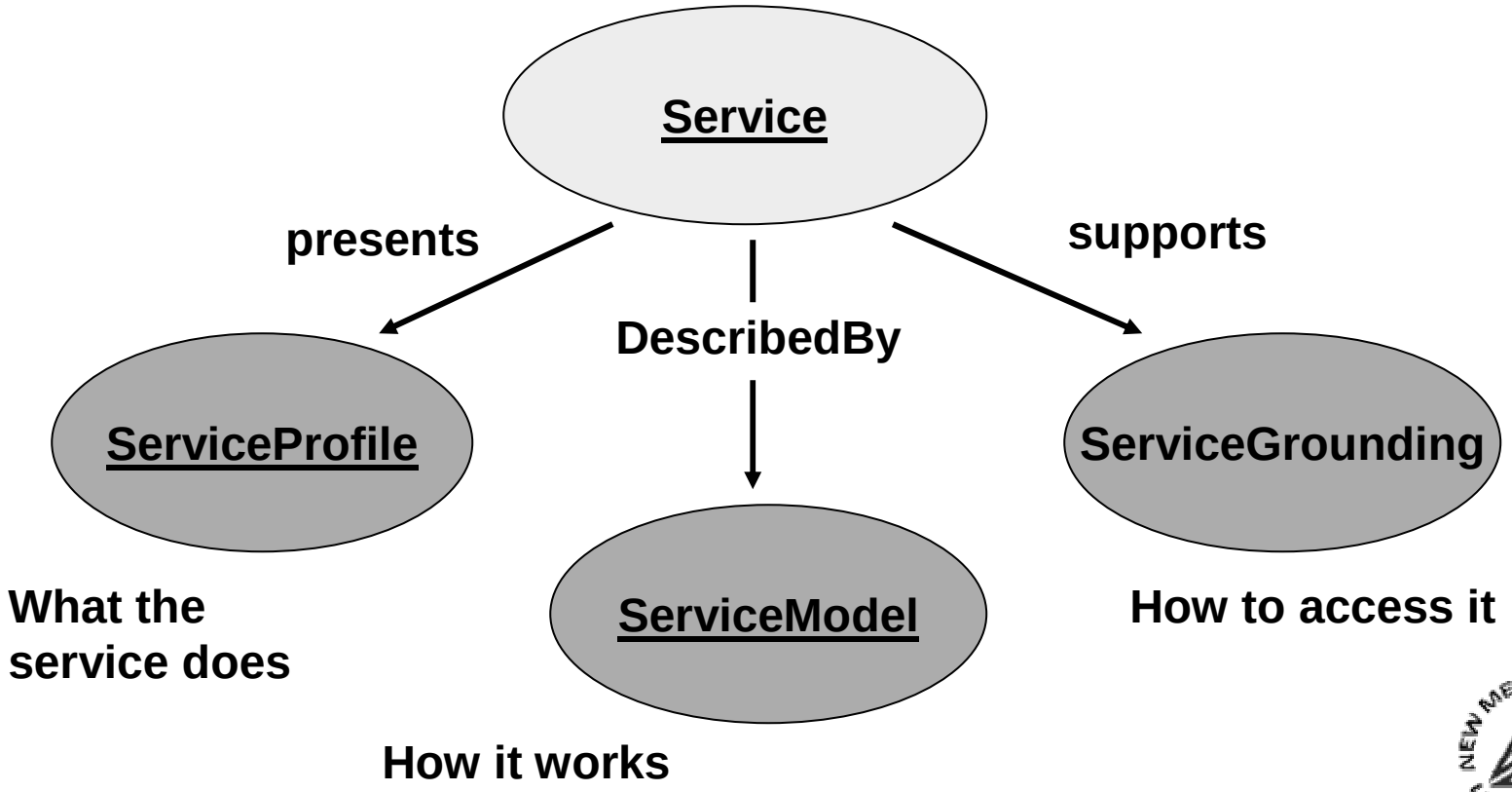
ΦLOG

- Embed intelligence
  - automatic configuration of logical pipeline
    - select services required to complete a given task
    - identify service locations, contact servers, etc.
    - automatic invocation of input/output data filters
    - optimal plan generating based upon trust level

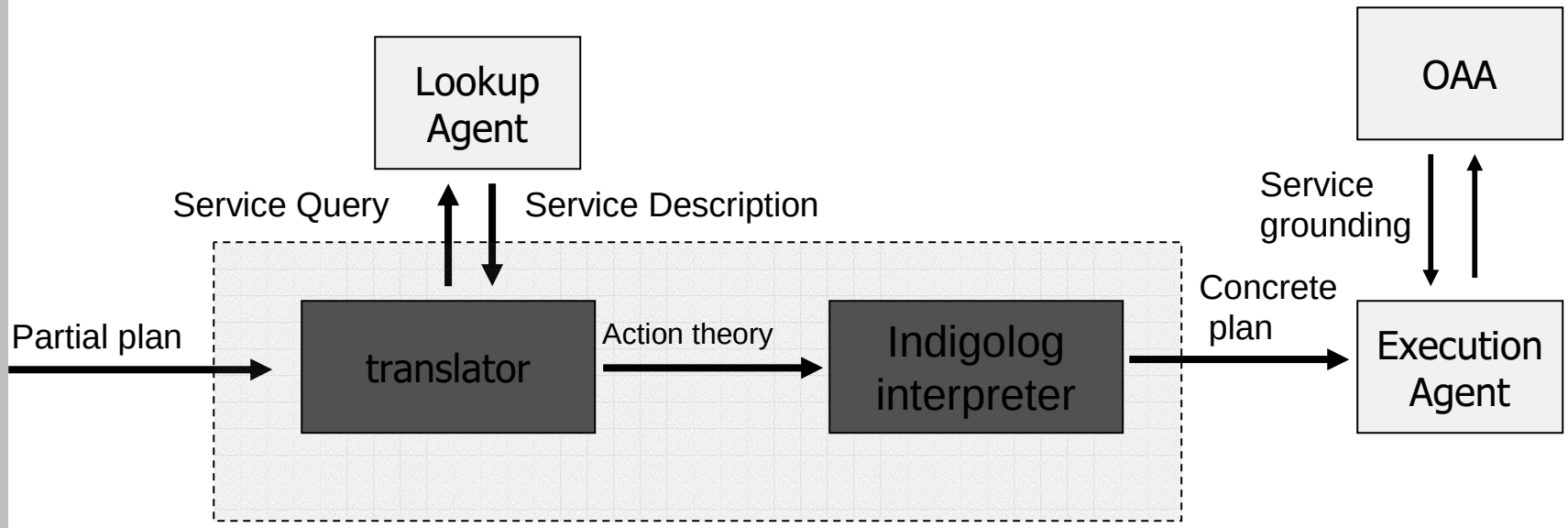
ΦLOG


ΦLOG Program
COMPILER
Partial Plan
Configuration Agent
Concrete plan
service lookup
service locations
Service invocation
Execution Agent
Facilitator (OAA)
service description
Service grounding
BLAST
GenBank
. . . .
CLUSTAL
Eigenvalue Computation
Services

DARPA Agent Markup Language



Configuration Agent (CA)



Sample ΦLOG Program

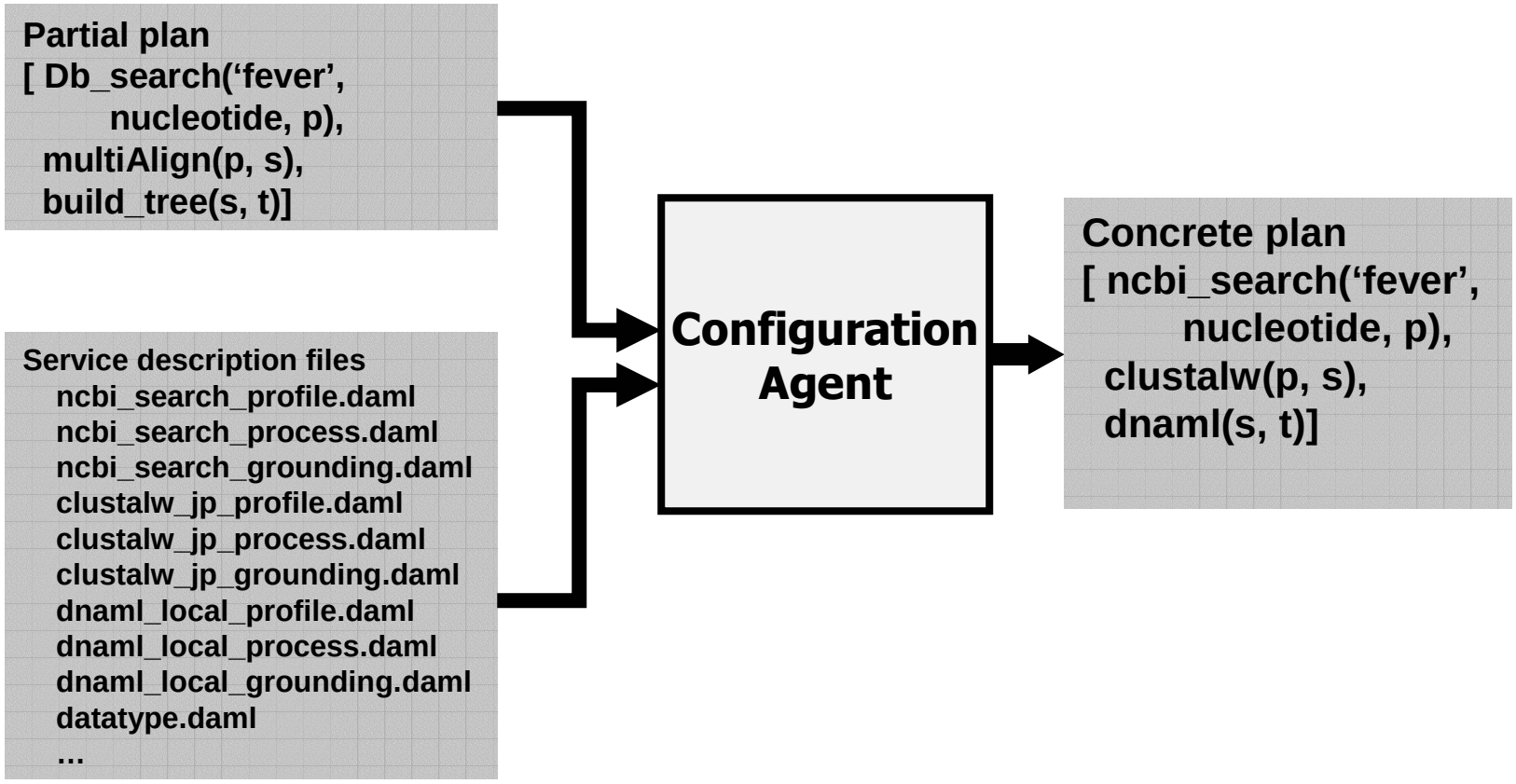
Sample1.pl:

p is { x : x is gene and name(x) contains “fever” }.

s is align(p).

t is build_tree(s).

Sample ΦLOG Program



Conclusion

- High level description of biological data and declarative language nature make it easy to operate by biologists
- Automation of service discovery free biologists from trivial human tinkering and more concentrated on biological nature
- Planning and OAA-based architecture allow biologists to explore different models and not restricted by specific package limitations.

Current Status

- preliminary design of parts of the language [IEEE Int. Conf. on Bioinformatics 2002]
- design of XML schema for service descriptions
- OAA-based architecture under development