KLAP Research Focus

Enrico Pontelli

Knowledge-representation, Logic, and Advanced Programming (KLAP) Laboratory

Overall Focus

Logic Programming Systems

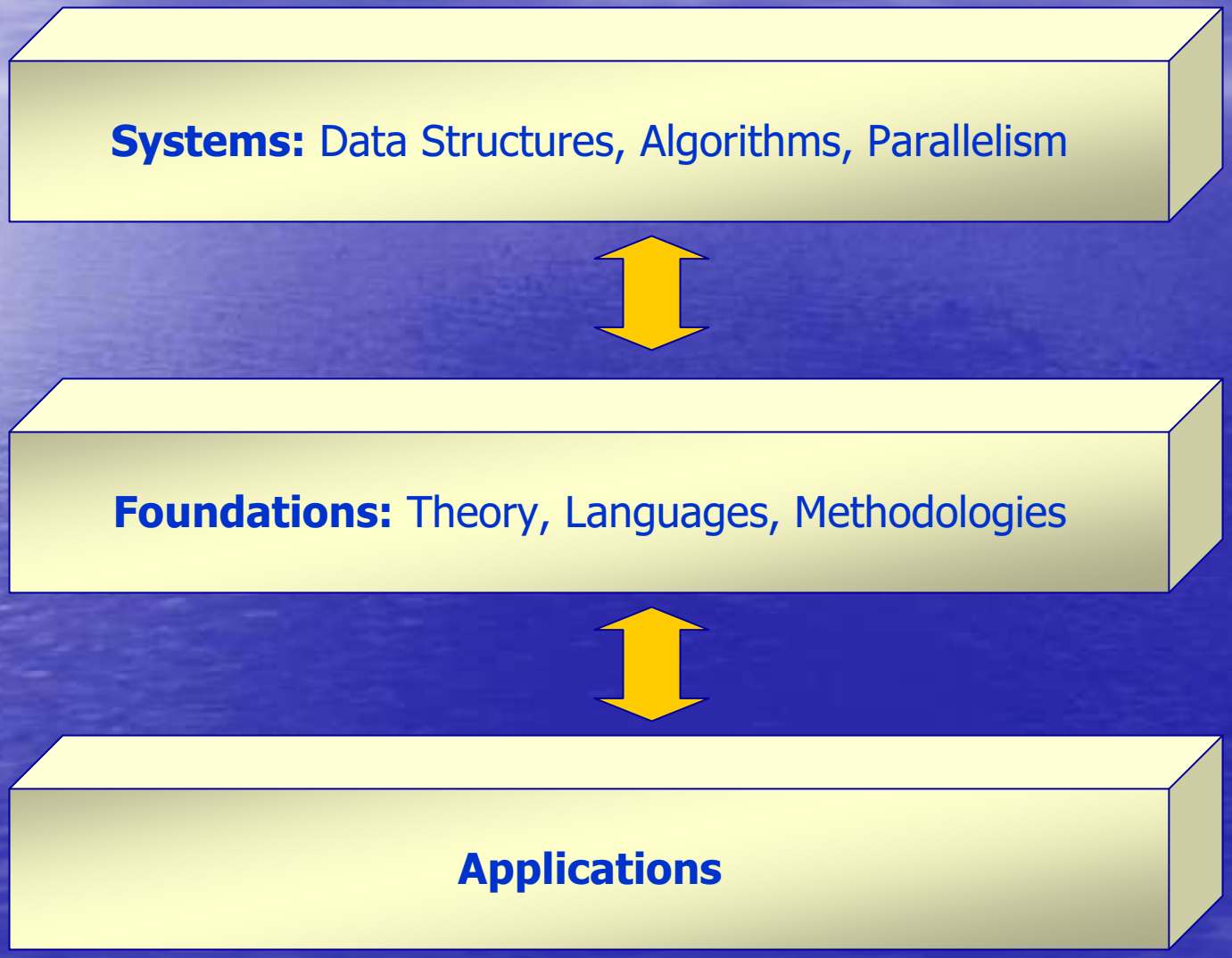
for

Knowledge Representation and Reasoning

and

Applications

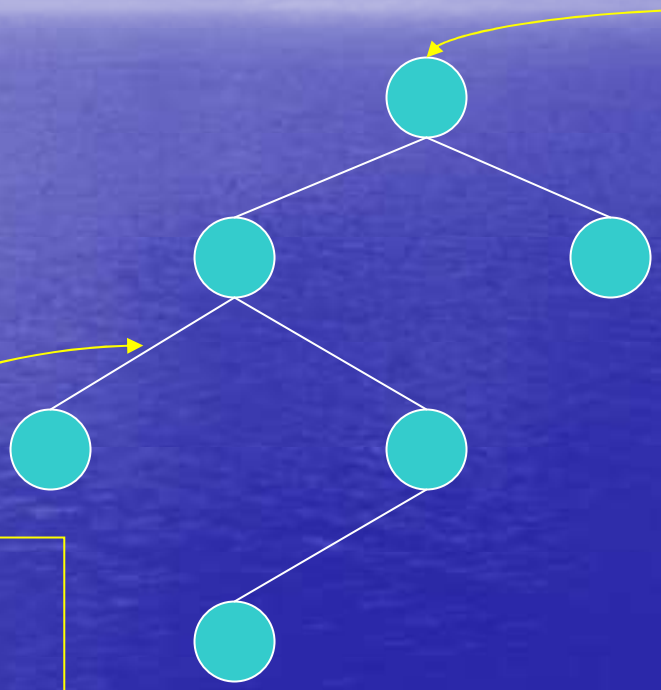
KLAP Activities



Systems

- Goals:
  - Efficiency and Scalability
  - Logic Programming and Constraint Programming
- Focus:
  - Non-ground Computation
  - Automatic Parallelization
  - Advanced Data Structures
  - Constraint Solving Algorithms

Automatic Parallelization of Logic and Constraint Programming



Non-deterministic Step:
Prolog: clause selection
ASP: literal choice
CP: labeling

Or-Parallelism
Search-Parallelism

Deterministic Steps:
Prolog: goal selection
ASP: model expansion
CP: constraint propagation

And-Parallelism
Parallel Propagation

Automatic Parallelization: Theory

- Advanced Data Structures
  - Abstract Modeling of Aspects of Parallel And- and Or-parallelism
  - Lower bound studies
  - Efficient (possibly optimal) data structures
- Recent Accomplishments
  - Matching lower and upper bound for environments management in or-parallelism
  - Efficient data structures for and-parallelism
  - Lower bounds for shared variables management in and-parallelism
  - Constant-time data structure for handling of order-sensitive computations

Automatic Parallelization: Practice

- Parallel Execution of Prolog:
  - *Or-parallelism* on scalable platforms
    - Branch splitting
    - Distributed bottom-most scheduling
  - *Independent and-parallelism*
    - ACE model
  - *Dependent and-parallelism*
    - Static Analysis
    - Filtered-binding Model
  - Success Stories…
    - ULTRA, Artwork, Aquarius, CIAO

Automatic Parallelization: Practice

- Answer Set Programming
  - Search Parallelism:
    - Novel strategies for environment management (hybrid copying/recomputation)
    - Novel strategies for dynamic task scheduling
  - Parallel Propagation:
    - Parallel Look-ahead
    - Parallel Model expansion
  - Pipelined Parallel Model Computation

Constraint Solvers for Set Expressions

- CLP(SET): Constraint Logic Programming over Hereditarily Finite Sets
  - Language definition
  - Constraint Solvers and Complexity Results
- Recent Directions
  - Constraints on Intensional Sets
  - Hybrid Constraints
  - From CLP(SET) to smodels(SET)

Foundations of KRR

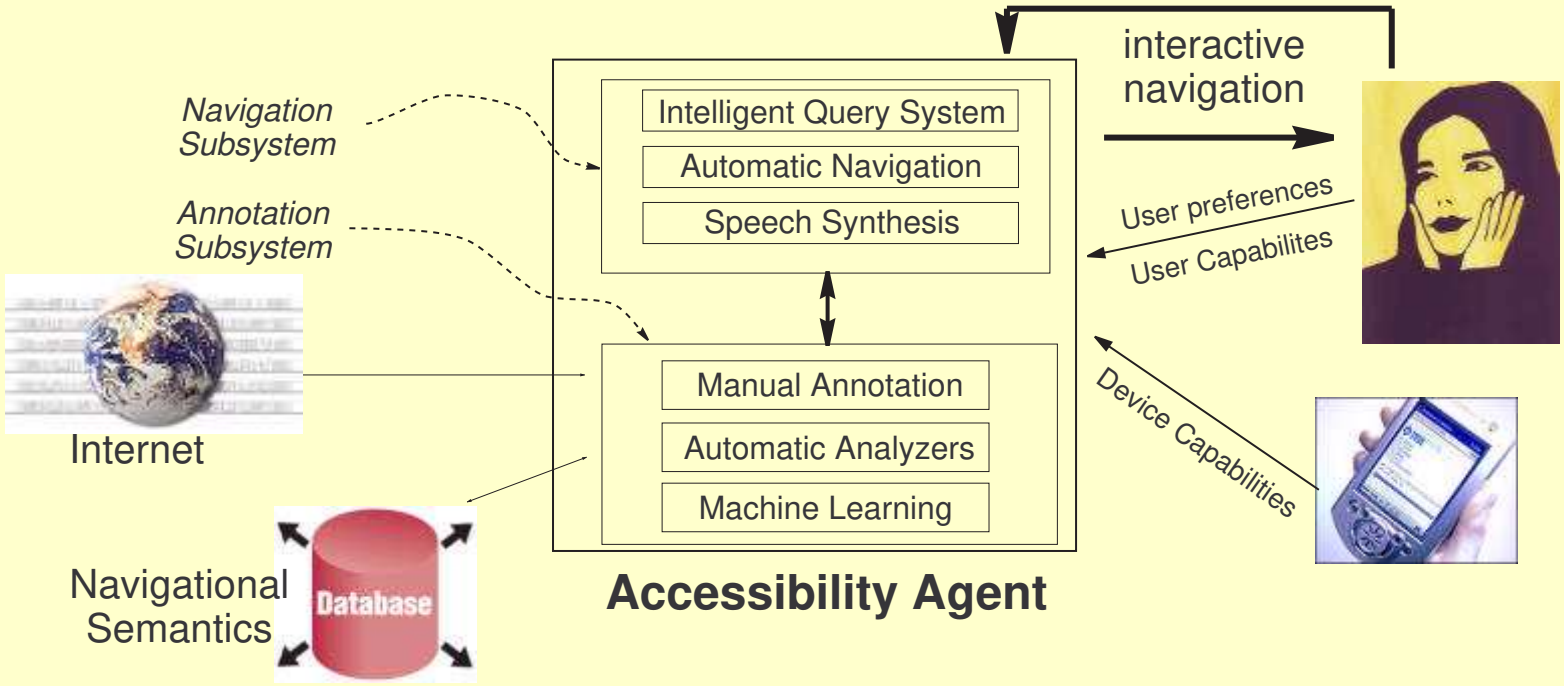
- Reasoning about Action and Change
  - Planning with Preferences
  - Planning with Knowledge-producing Actions
  - Planning with Domain-dependent Knowledge
  - …
- Knowledge Maintenance
  - Revision Programming
  - Preferences in Database Updates

Applications

- Properties
  - Complex domains
  - Logic and Constraint Programming Technology
  - Use of Domain Specific Methodologies
- Domains
  - Universal Accessibility
  - Phylogenetic Inference

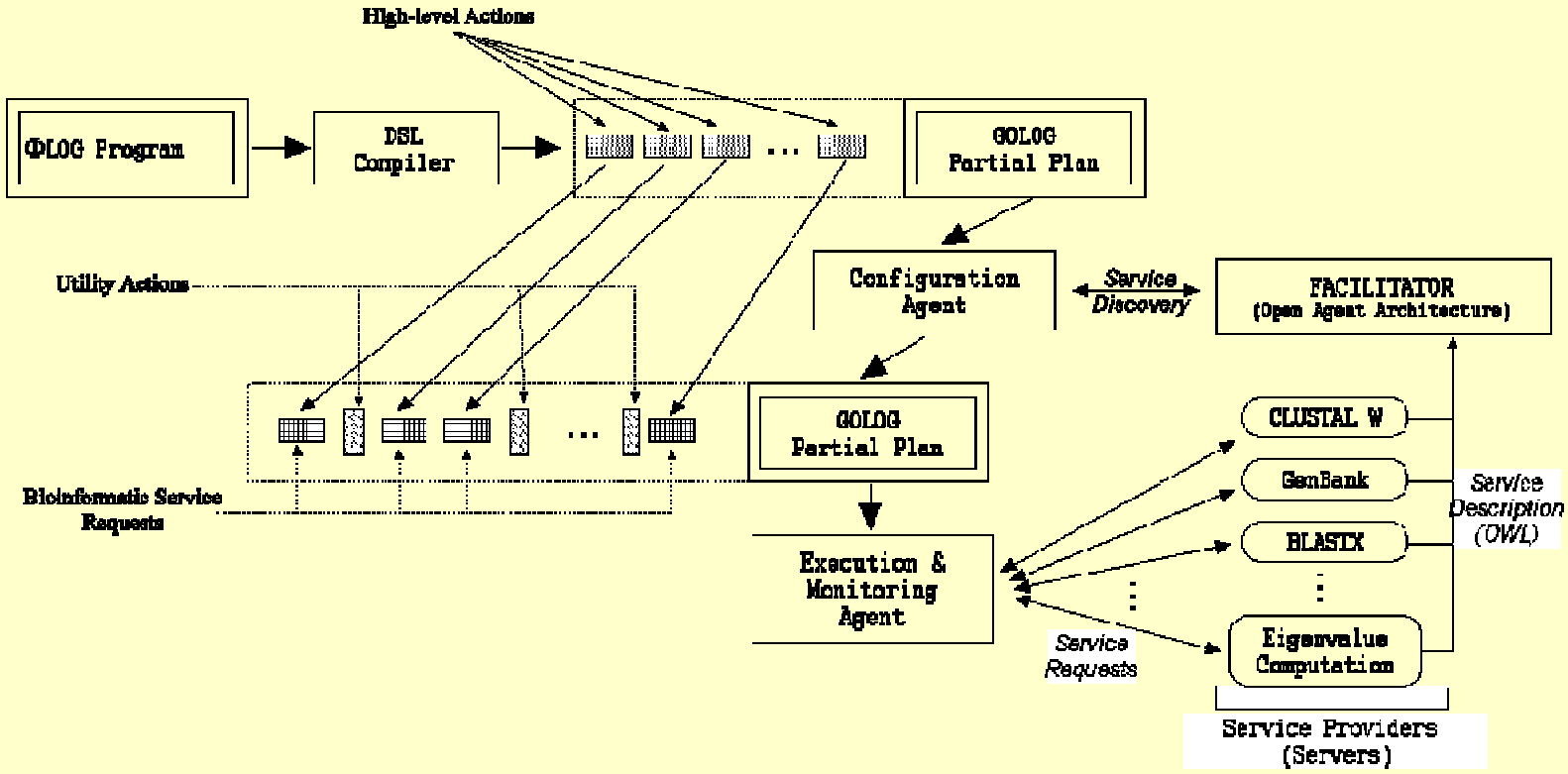
Web Accessibility

- Web accessibility for visually impaired
  - domain specific ontology for Web navigation
  - agent-based architecture for user-directed navigation



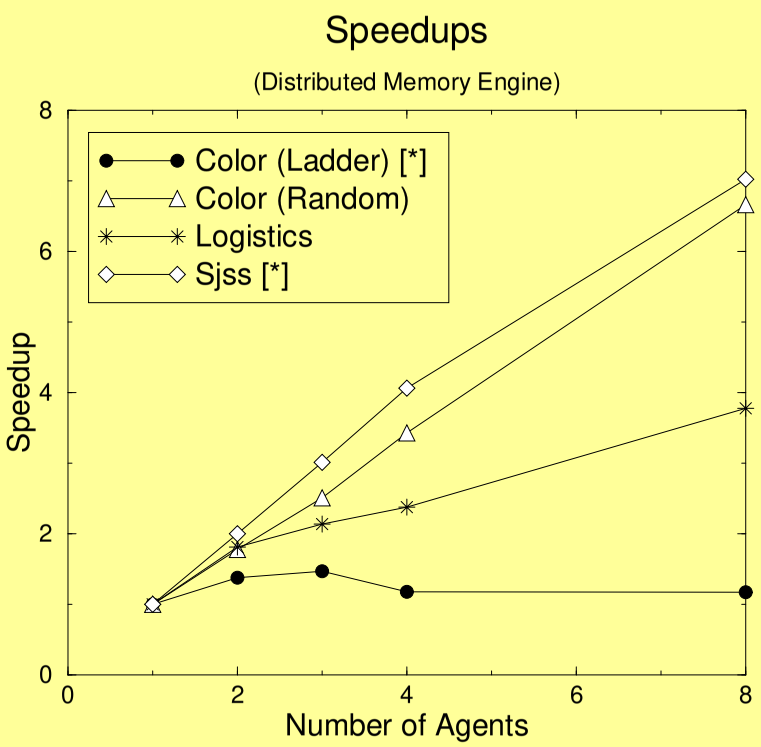
Frameworks for Phylogenetic Inference

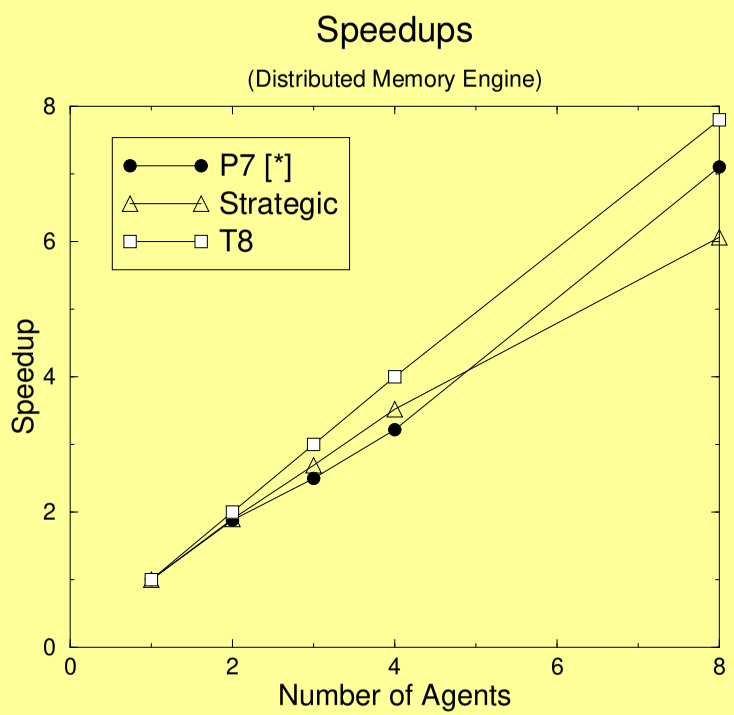
- **Domain Specific Language for Phylogenetic Inference and Evolutionary Biology**
- **Agent-based Execution Platform**



Projects Participants

- **Principal Investigators**
  - I. Pivkina
  - E. Pontelli
  - D. Ranjan
  - S. Cao Tran
  - K. Villaverde
- **Doctoral Students**
  - Mei Li (Phylogenetic Inference)
  - Islam Elkabani (Reasoning and Declarative Languages)
  - Emad Saad (Reasoning and Declarative Languages)
  - Hung Le Viet (Parallelism)
  - Omar El-Khatib (Reasoning, Parallelism)
  - Phan Tu (Planning)
  - Chongbing Liu (Parallelism)
  - Yu Pan (Phylogenetic Inference)
  - B. Palmer (Accessibility)
- **Master's Students**
  - D. Cuaron (Parallelism)
  - A. Vezolainen (Parallelism)
  - C. Ngo (Accessibility)
  - S. Wu (Phylogenetic Inference)
  - P. Nguyen (Web Services)
  - X. Xiao (Consraints)
  - H. Poogalia (Answer Set Programming)
  - I. Strascina (Agents and ASP)
  - R. Reddy (Accessibility)


Speedups
(Distributed Memory Engine)
Color (Ladder) [*]
Color (Random)
Logistics
Sjss [*]
Speedup
Number of Agents
0
2
4
6
8


Speedups
(Distributed Memory Engine)
P7 [*]
Strategic
T8
Speedup
Number of Agents
0
2
4
6
8

Frameworks for the Design & Development of Domain Specific Languages (DSL)

- Horn-logic Denotations
  - Denotational Semantics in Logic Programming
  - Formal framework for specification of DSLs (Syntax & Semantics)
- Impact
  - Automatic derivation of provably correct interpreters (w.r.t. DSL specification)
  - Automatic derivation of a compiler via partial evaluation
  - Denotations as logic programs:
    - verification
    - automated debugging