



The MathServe System for Semantic Web Reasoning Services

Jürgen Zimmer and Serge Autexier

Universität des Saarlandes & German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany

This work was partially supported by the CALCULEMUS IHP Training Network HPRN-CT-2000-00102

Motivation

Many **specialized reasoning systems** are currently available:

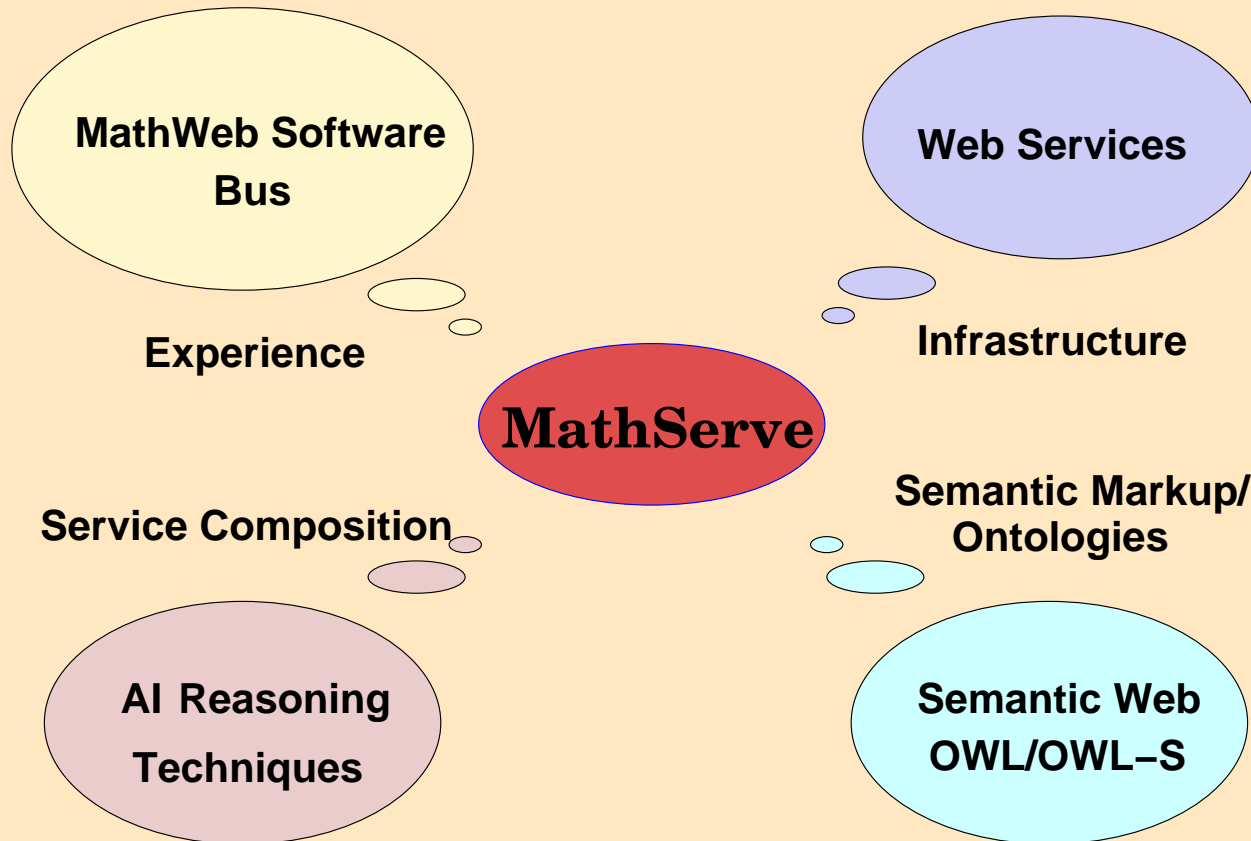
- Automated Theorem Provers (e.g., EP, SPASS, Vampire).
- SAT solvers (e.g., MiniSAT, ZChaff).
- Decision Procedures (e.g., MathSAT, CVC Lite)
- Proof/Problem transformation systems (e.g., Tramp, Otterfier).

Problem: Many systems are **not open** and **only usable by experts**.

Aims: Make services provided by reasoning systems:

- **interoperable** and **easily accessible** (problem descriptions).
- available for **automated retrieval and service composition**.

The MathServe System



- Semantics of services described in OWL-S.
- Domain ontology standardises answers from reasoning systems.

MathServe Services



- **Clause normal form generators** tptp2X and Flotter.
- Latest versions of **ATP systems** DCTP, EP, SPASS, Vampire, Paradox and Waldmeister.
- **Finite model generators** Mace, SEM.
- **Proof transformation tools** Tramp, Otterfier.

The Vampire ATP Service



profile VampireATP:

inputs: *tptp_problem* :: mw#TptpProblem

time_res :: mw#TimeResource

outputs: *atp_result* :: mw#FoAtpResult

preconds:

effects: resultFor(*atp_result*, *tptp_problem*) \wedge

(status(*atp_result*, *szs#Theorem*) \vee ... \vee status(*atp_result*, *szs#Unsatisfiable*))

categs:

params: problemClass(*tptp_problem*, mw#FOF_NKC_EPR)

\Rightarrow status(*atp_result*, *szs#Theorem*) (0.93) (4755ms)

...

problemClass(*tptp_problem*, mw#CNF_NKS_RFO_SEQ_NHN)

\Rightarrow status(*atp_result*, *szs#Unsatisfiable*) (0.52) (20984ms)

(Performance measured on TPTP Library v3.1.0.)

Submitting Problems – Queries



```
query FoATP-Proof-Query:  
  inputs:      fof_problem :: mw#TptpFOFProblem  
  outputs:     atp_result :: mw#FoAtpResult  
  preconds:  
  effects:     proofOf(atp_result, fof_problem)  
  categs:  
  timeout:     10 secs
```

- Service matchmaking by simple class subsumption test on input/output parameters

Submitting Problems – Queries



```
query FoATP-Proof-Query:  
  inputs:      fof_problem :: mw#TptpFOFProblem  
  outputs:     atp_result :: mw#FoAtpResult  
  preconds:  
  effects:     proofOf(atp_result, fof_problem)  
  categs:  
  timeout:     10 secs
```

- Service matchmaking by simple class subsumption test on input/output parameters
- If no single service can answer a query, try to compose services

Service Composition in MathServe

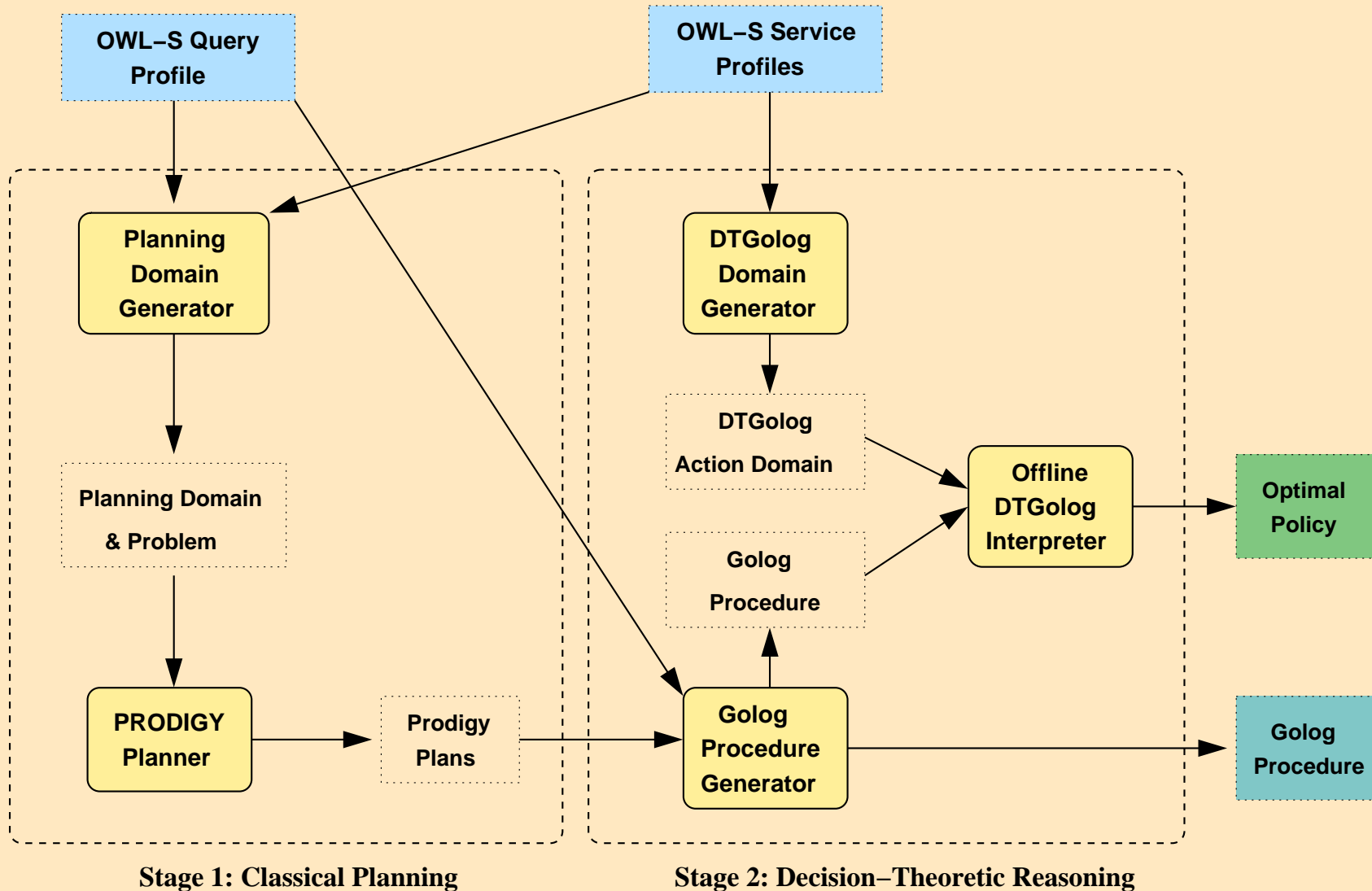


Example query requiring service composition:

```
query ND-Proof-Query:
  inputs:      fof_problem :: mw#TptpFOFProblem
  outputs:     nd_proof :: mw#NDProof
  preconds:
  effects:     proofOf(nd_proof, fof_problem)
  categs:
  timeout:     10 secs
```

- Combines **classical planning** and **decision-theoretic reasoning**.
- **Composite service** represented in **Golog** language.
- **Golog** = high-level programming language based on situation calculus.

Automated Service Composition



Optimal Policy for ND-Proof-Query



```
proc ndProofQueryOpt(puz_problem, nd_proof)  
  trampCNF(puz_problem, cnf_problem, delta_relation) ;  
  tptpAnalyser(cnf_problem, prob_class) ;  
  if problemClass(cnf_problem, mw#FOF_NKC_EPR) then  
    spassATP(cnf_problem, time_res_1, atp_result) ;  
    otterfier(atp_result, new_result) ;  
    trampNDforFOF(puz_problem, new_result, delta_relation, nd_proof)  
  else  
    if problemClass(cnf_problem, mw#CNF_NKS_RFO_SEQ_NHN) then  
      vampireATP(cnf_problem, time_res_1, atp_result) ;  
      otterfier(atp_result, new_result) ;  
      trampNDforFOF(puz_problem, new_result, delta_relation, nd_proof)  
    else  
      ...  
endProc
```

Evaluation on CASC-20



System	Problems given	Problems solved	Percentage of given	Percentage complete
MathServe 0.62	660	392	59.4%	59.4%
Ep 0.9pre3	660	409	62.0%	62.0%
Vampire 8.0	540	430	79.6%	65.2%
Vampire 7.0	300	262	87.4%	39.7%
MathServe 0.71	660	440	67.7%	67.7%

- MathServe 0.62 used older versions of EP (0.81) and Vampire (7.0).
- MathServe 0.62 could not handle large problems (> 2MB).
- MathServe 0.71 after CASC-20 with EP 0.9 and Vampire 8.0.

Availability and Usability



- Binary and source distribution available at
<http://www.ags.uni-sb.de/~jzimmer/mathserve.html>
- MathServe is used by
 - ▶ [Omega/Core](#) proof assistant,
 - ▶ [Hets toolset](#) (University of Bremen)
 - ▶ [Verifun project](#) (Technical University Darmstadt)

See [MathServe 0.8](#) at CASC-J3!

Related and Future Work

Related

- Projects MathBroker and MONET (Computation Web Services).
- SATzilla (algorithm portfolio for SAT).

Future Work:

- Add SAT solvers, decision procedures, computer algebra services and theory exploration tools.
- Support for higher-order provers (LEO, TPS).
- Translation services (for inputs and outputs: proofs, models, etc.)
- Stateful Web Services (*Web Service Resource Framework*).