Adding Common Logic Support to the Heterogeneous Toolset

John Bateman / Till Mossakowski

Universität Bremen

The story so far...

- Work in a variety of projects concerned with diverse ontologies: need for **heterogeneity**

- Formal and computational tools adopted:

  - CASL
    Common Algebraic Specification Language
    (for specification, structuring and relating of theories:
    including ontologies)

  - HeTS
    Heterogeneous Tool Set
    (for connecting to a range of reasoners
    and for working with structured specifications)

  :

Formalization choice: CASL
Common Algebraic Specification Language

- Standardised first-order specification language

- designed by CoFI "Common Framework Initiative for algebraic specification and development" since 1995

- de facto standard approved by IFIP WG 1.3 "Foundations of Systems Specifications" (1998)

- extensive User Manual and Reference Manual now available from Springer (LNCS 2900, LNCS 2960)

- straightforwardly extensible: higher-order logic, modal logic, ...

- **supports structured specifications including imports, hiding, renaming, union, extensions, etc.**

Hets
The Heterogeneous Tool Set

- Hets is a multi-logic parsing, analysis and proof tool

- Logics covered include:
  - propositional logic
    (with SAT solvers minisat and zChaff),
  - OWL (with provers Pellet and FACT++),
  - first-order logic
    (with provers SPASS, Vampire, Darwin and MathServe),
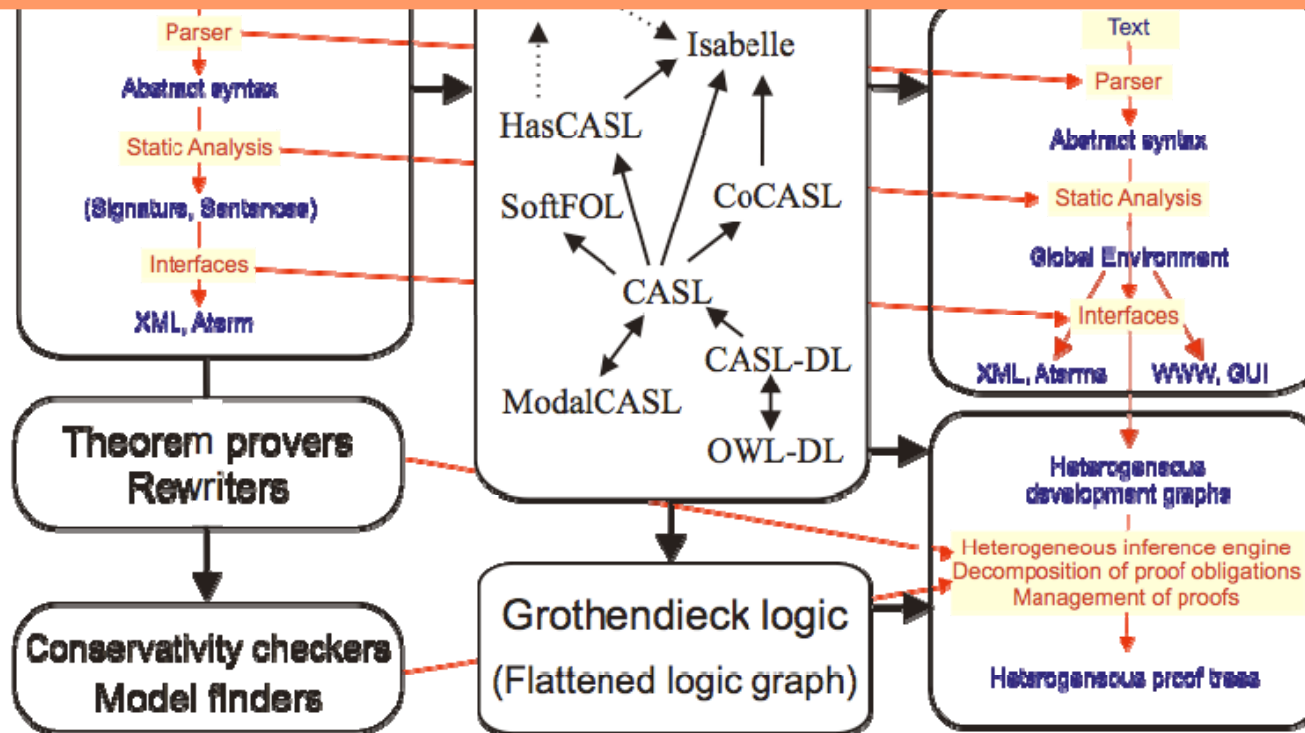 - higher-order logic (with prover Isabelle).

The Heterogeneous Tool Set

Structuring mechanisms: potentially applicable to **any specification language** as an additional layer of 'meta'-organisation for semantic modularity

HetCASL

Institution Theory

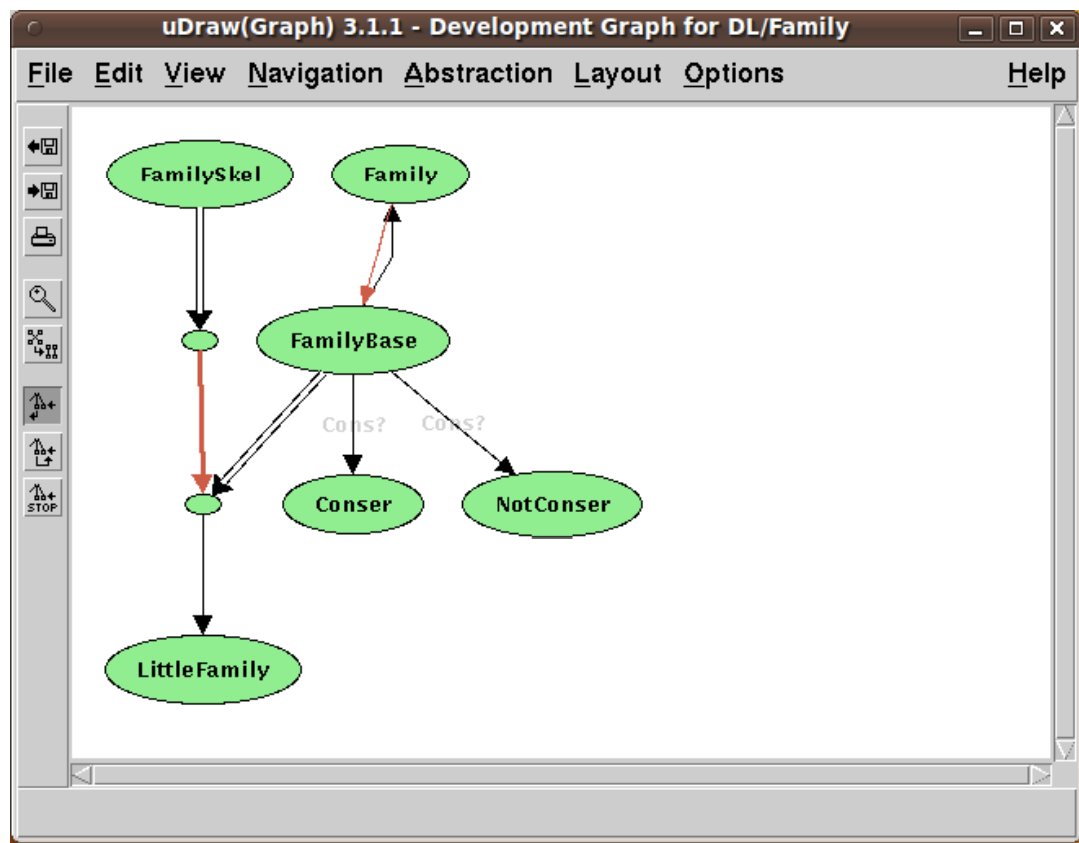**5**

Hets
The Heterogeneous Tool Set

Hets is freely available from:

www.dfki.de/sks/hets

Structuring of theories

Development Graph

showing dependencies between specifications and proof obligations

Links: theory morphisms

- **imports of theories**
- relative interpretations of theories
  - **open**
  - **proved**

Flashback:
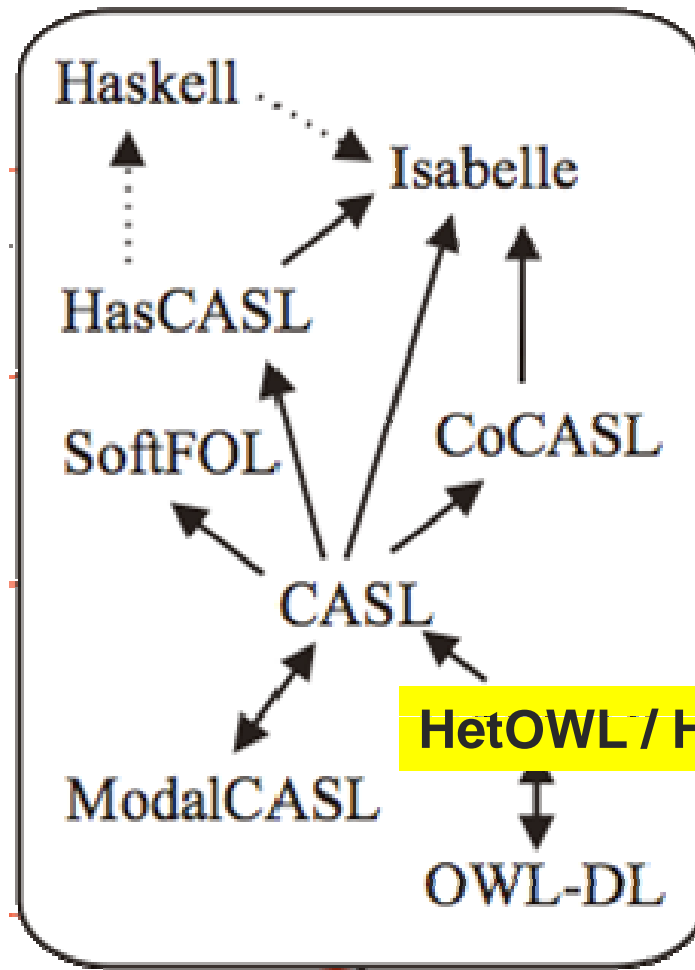## Further Steps... (Ontolog, March 2010)

- we have already added OWL-DL to the family of logics supported

- we are exploring **combining** the structuring principles of CASL and description logics

Now:

- we are planning to add Common Logic as a HETS logic node

- thereby providing access to all the tools already linked to HETS

HeTS:
Continuing extension of the treated logics

logic OWL

spec Family_OWL =
  Class: Female
  Class: Person
  ObjectProperty: hasChild
  Class: Mother
  EquivalentTo:
     Female and hasChild some Person
  Individual: Carol Facts: hasChild Peter
end

HetOWL / HOWL

Flashback:
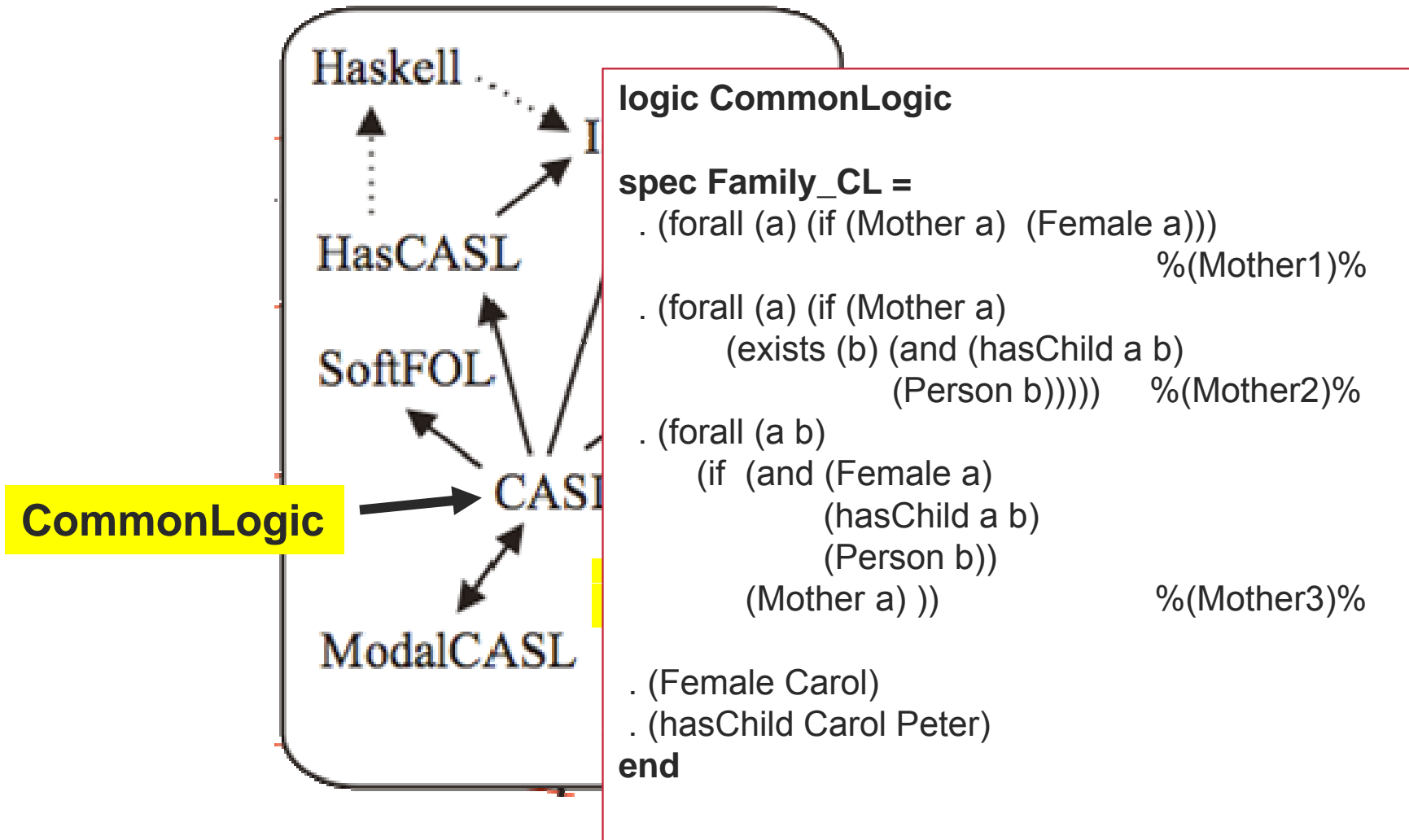## Further Steps... (Ontolog, March 2010)

- we have already added OWL-DL to the family of logics supported

- we are exploring **combining** the structuring principles of CASL and description logics

Now:

- we are planning to add Common Logic as a HETS logic node

- thereby providing access to all the tools already linked to HETS

10

HeTS:

Continuing extension of the treated logics

Haskell

HasCASL

SoftFOL

CommonLogic

CASL

ModalCASL

```
logic CommonLogic

spec Family_CL =
  . (forall (a) (if (Mother a)  (Female a)))
                                          %(Mother1)%
  . (forall (a) (if (Mother a)
         (exists (b) (and (hasChild a b)
                       (Person b)))))     %(Mother2)%
  . (forall (a b)
       (if  (and (Female a)
               (hasChild a b)
               (Person b))
       (Mother a) ))                      %(Mother3)%

  . (Female Carol)
  . (hasChild Carol Peter)
end
```

Current Hets capabilities

- Capable of reading CLIF files, e.g., the specifications of the COLORE Repository

- SPASS, Vampire, Darwin and MathServe available for proofs in Common Logic

- Isabelle available for proofs (if induction on lists needed)

- Translation from OWL into Common Logic provided

- HetCASL provides heterogeneous structuring and refinement (.het files with HetCASL structuring for Common Logic)

  - A **view** expresses a refinement (= interpretation of theories = logical entailment between theories)
    view v : Family_OWL to Family_CL **end**

12

Demonstration

- simple example: Cat(kitty)
  - Parsing, proving
- duration.clif from Colore
- Hets logic graph
- View from OWL ontology to CL ontology
- Translation OWL $\rightarrow$ CL
- Consistency

Ongoing work

- Using the module structure of Common Logic Specifications

  - The semantics of the module construct in Common Logic must be agreed upon

  - A compositional semantics would be much easier to implement (but need to check details)

- Support for disproving

Conclusion

- Hets continues to provide support for the kinds of diverse heterogeneous ontologies we require, now extended to Common Logic

- Link to the OMDoc world is under development → will provide web interface and repository for Hets specifications with XML database and nice querying

- Link to the OOR world building on this would also be highly desirable