A Logical Spreadsheet

Overview

Motivations

- A platform for show-casing our work in
  - elaboration tolerance
  - non-monotonic reasoning
  - reasoning about change
  - context (including ontology construction)
  - obviousness
- A grant deliverable

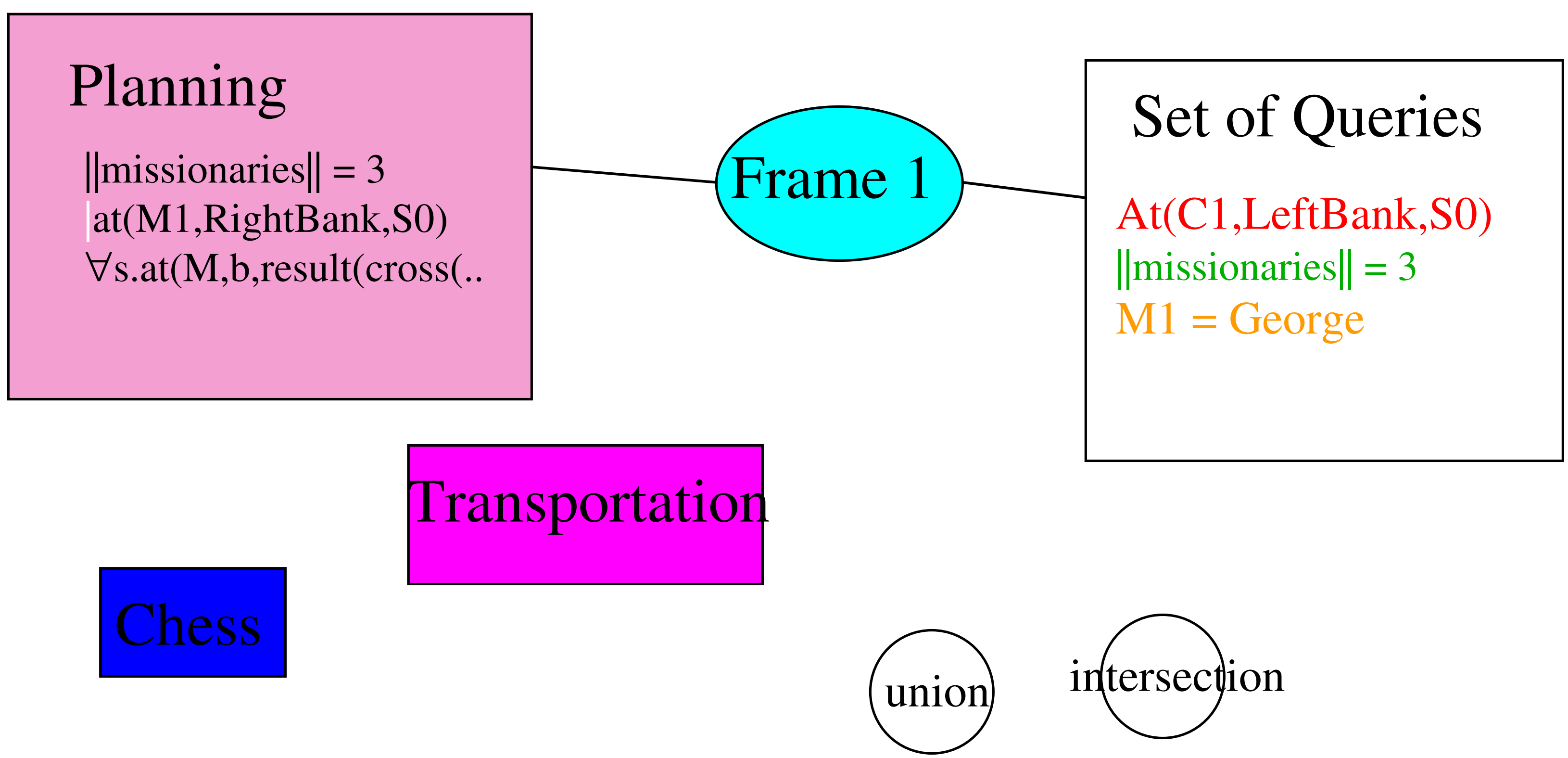
What and When

- First prototype by end of August
- Minimal features
  - updates
  - basic operations on theories
  - CLIM 2.0 front end
  - Based on Epilog engine

What is a logical Spreadsheet

- Current spreadsheets contain **cells** defined by functional equations.
  - Changes in one cell are reflected in the others
- Logical spreadsheet has predicates/ sentences instead of cells
  - Equations become a theory
  - changes become a **revision.**

Picture of Logical Spreadsheet



How it works

- The set of queries tells whether a sentence follows from a given theory
- The frame connects the theory to the queries
- Frames give us a way of updating.
- When we add to the theory, the frame decides what, if anything needs to be removed.

Comparison of Update in logical and normal Spreadsheets

- Update in normal spreadsheets is always by removing an old value, and replacing it by a new value
- The functional nature supports this.
- Logic is not functional, and our language is much richer.
- Therefore a new semantics for update is needed

Cartesian Frames

- We can use the notion of Cartesian Frames.
- Consider $s = \sqrt{x^2 + y^2 + z^2}$ when $x,y,z = 2$
- Then $s = \sqrt{12}$
- If we change x to 1 then if x y and z are independent $s = 3$
- If s x and y are independent then $z = \sqrt{7}$
- The result depends on what we consider independent, our frame.

How to calculate update given a frame.

- We can express update using John's 62 idea of c and a.
- We can also use circumscription,
- We minimize the negation of the sentences that describe the frame.
- We then compute whether the queries follow from the circumscription.

How to calculate circumscriptive entailment

- First Version (August).
- We use the DLS algorithm, inplemented in LISP.
- We then use a first order theorem prover, Epilog.
- Our re-implementation uses KIF, with second order extensions. Its output is legal KIF.

User Interface Issues

- CLIM 2.0 Common Lisp Interface Manager
- Object oriented interface builder
- Runs on Steam.
- Each object has a presentation method that draws it.
- Allows quick interface building

Examples

- Five initial examples
  - Planning in Blocksworld
  - Elaborations in M&C
  - Number examples
  - Scheduling problems (TPFDD's)
  - inheritance problems?

Tools for Combining Theories

- Simple set theoretic operations, union, intersection.
- Restricting theory to a given sub-language.
  - non-trivial problem
  - Let L ⊃ L'. Given A in L find B in L' such that for all C in L' A |= C iff B |= C.
  - Can be done by removing 2nd order quantifiers.

Ways of defining Frames and other update methods

- Initially we will support
  - frames
  - circumscription policies
- Later allow
  - products and unions of other orders
  - weakest order given certain properties
  - conditional filters
  - etc.

Future Extensions to Entailment methods

- First version will use normal circumscription
- Later versions
  - cofinal entailment
  - domain-formula versions
  - limit entailment
- These will allow more expressivity and ensure consistency preservation.

Choices

- SCAN rather than DLS
- Otter rather than Epilog
- Languages other than KIF
- Avoid second order logic?

Future theorem proving approaches

- Move to obviousness based approach
  - free affine multiplicative logics
  - iterative deepening with n !
  - Allows logical chromotography
- Save proof terms
  - extract plans from proofs
  - give better analysis of plans

Future Language Extensions

- Look at description logics
- Extend description logics to first order
- Add fixed points.
- Will support obviousness reasoning

Extensions to Queries

- Values of terms
- Satisfiable as well as provable
- Truth maintainance
- Abduction, what would it take to make A true.

Conclusion

- Allows us to showcase new methods of non-mon, elab tol, belief revision
- Gives us a tool for experimentation with theories
- Gives a first set of operations on theories/ contexts
- Supports research on obviousness, planning, relatonships among KR formalisms.