Linking GAMS to Solvers Using COIN-OSI

Michael Bussieck
Steve Dirkse
GAMS Development Corporation

ICS - Annapolis
January 5-7, 2005

Outline

- Background & motivation
- Common agenda: COIN & GAMS
- GAMS/COIN links
  - Helper class for GAMS models
  - Requirements for GAMS solvers
- Future work
  - Nonlinear extensions
  - Model interface in addition to solver

Background

- GAMS has been and remains *user-focused*
  - User needs trump solver conventions
  - Focus on the model, not on the solver
- Typical user expectations:
  - “out-of-the-box” installation
  - GAMS solver conventions (e.g. optcr, reslim)
  - Tech support when things don’t work well

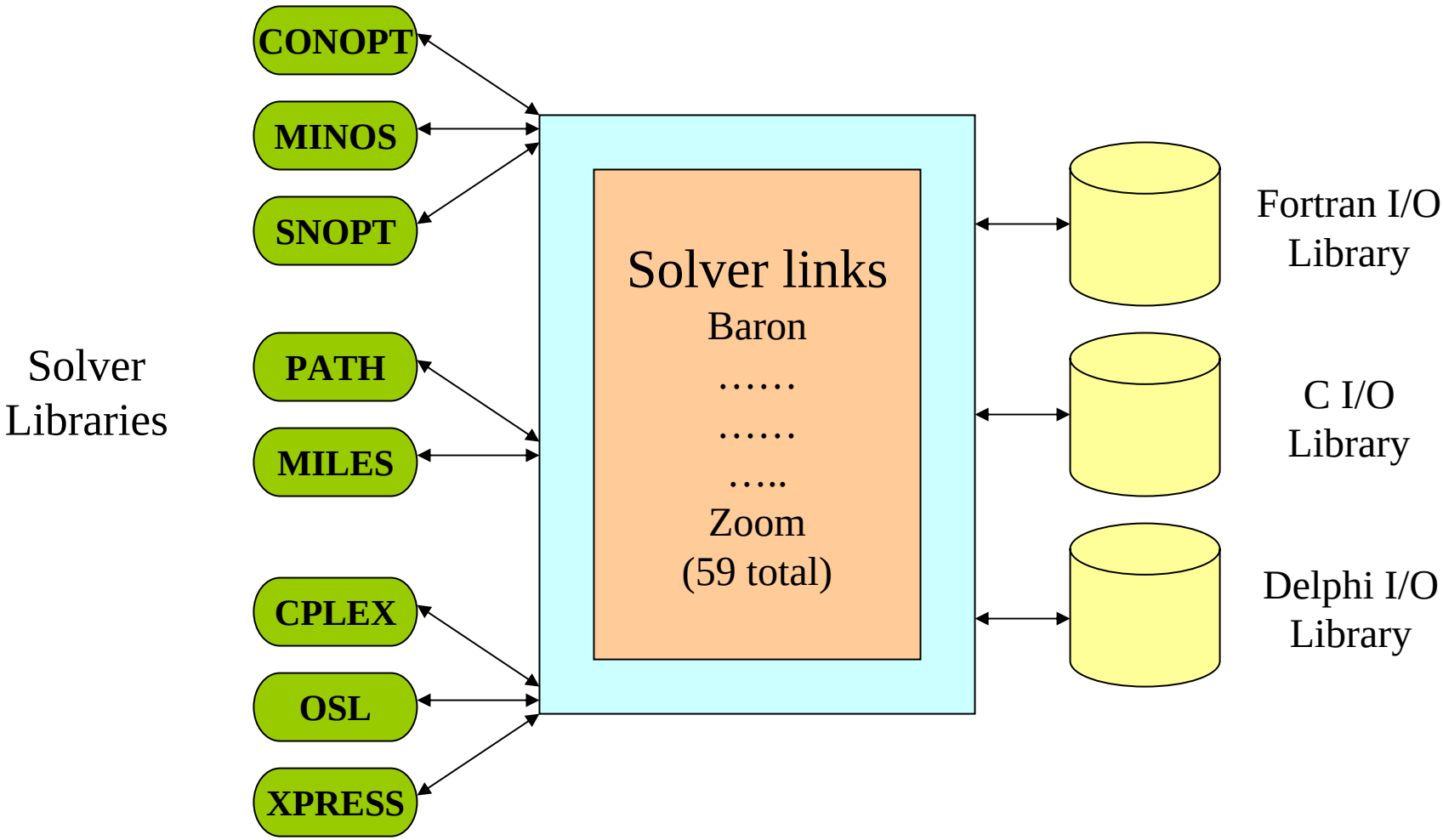
Why add COIN?

- Offer something to non-typical users
  - Willing & able to build solvers themselves
  - Don’t require GAMS conventions or support
- Does not destroy existing markets
  - Preserves the value of our existing systems
- Can be done easily
- A good start on further development

COIN Agenda (Ralphs)

- Accelerate the pace of R&D
  - Reuse instead of reinvent.
  - Reduce dev. time & increase robustness.
  - Increase interoperability (open standards).
- Define standards and interfaces
  - Peer reviewed and freely available
- Provide tools for practitioners

Re-use? What's that?!?



GAMS Agenda

- Eliminate redundant I/O libraries
- Find and eliminate common link code
  - Move code into I/O library
  - Optional presolving layer
  - More uniform interface to solvers
- Requires solver interface standards
  - Must be done by the algorithm R&D groups
  - We can facilitate this, bring them together

GAMS' Open Stance

- Our approach is an open one
- Independent Modeling System
  - Solvers from different vendors, multiple platforms
- GAMS fits into larger projects, doesn't dictate
  - One piece of the puzzle
  - We must work well with other software
  - Interface with other data sources and formats
  - Can be driver or driven
- Translation tools to and from GAMS

GAMS/COIN links

- Included in the GAMS CD
  - CoinGLPK
  - CoinSBB
- Source and build instructions available
  - http://www.gams.com/gamscoin/
  - Required libs available from GAMS
  - Migration to COIN repository ongoing

- Current builds: Windows and Linux
  - Other (GAMS) platforms should work as well
  - Windows build uses MinGW and MSYS
- Links are very lightweight
  - Primarily use generic (OSI) interface.
- Link source to be freely available (CPL?)
- Use a helper class GamsModel
  - Also to be available under CPL
  - Insulate link from GAMS details and changes

GamsModel class

- Model class for LP and MIP problems
- Encapsulate *most* of what is GAMS-specific
  - Problem data (matrix, bounds, integer types)
  - Objective function versus objective variable
  - Algorithm parameters (iterlim, ??)
  - Input/output conventions
- Use C++ to interface with COIN-OSI
- Use OSI compatible data structures (matrix)

What's missing?

- Time limit
  - All GAMS solvers honor a time limit
  - Links use solver-dependent layer for this
- "Best bound" and optcr – the GAMS way
  - Solver links monitor progress of B&B search
  - Quit when incumbent is within optcr of the best bound
  - Return status that indicates this condition – distinct from proven optimality

- Other GAMS controls not implemented
  - MIP: cutoff, cheat, priorities, optca, nodlim
  - Option file (default solver options only)
- A facility to query a COIN-OSI solver re: its capabilities would be useful
  - Allows for unimplemented options
  - Avoids the solver-dependent layer
  - Enforces uniformity of implemented options
- Similar for returns (e.g. node count, best bound)

Early success

- Client (OR consultant) developed cutting stock model for **small** business solution provider (cheap!)
- Reasons to implement this with GAMS
  - Cutting stock model in GAMS Model library
  - Excellent extended user support (modeling help)
  - Attractive pricing due to “free” GAMS/CoinSBB: $5,600 versus $17,850 (with GAMS/CPLEX)

Conclusions – current OSI

- Springboard for GAMS and COIN for further cooperation
- Working prototype of seamless connection between GAMS and OSI solvers
- “Trend setter” for other commercial *problem providers:*
  - MPL will also interface to OSI

LP vs. NLP

- Problem specification: what format?
- What does the solver see?
  - Point-based information (func, grad, hess)
  - More info needed for global solvers
- Must handle arithmetic errors (sqrt(-1))
  - Quit on first error or use non-stop arithmetic?
  - Helpful messages about errors are crucial
- Algorithmic issues

NLP Solvers

- Solver links more difficult to write/maintain
  - Not specific to the GAMS case
- Input format may be solver-specific
  - SIF, conic programming, callbacks, C/F77
  - Makes changing or adding solvers difficult
- Can the model be presolved?
  - May depend on problem format used
  - Presolving steps may change model structure

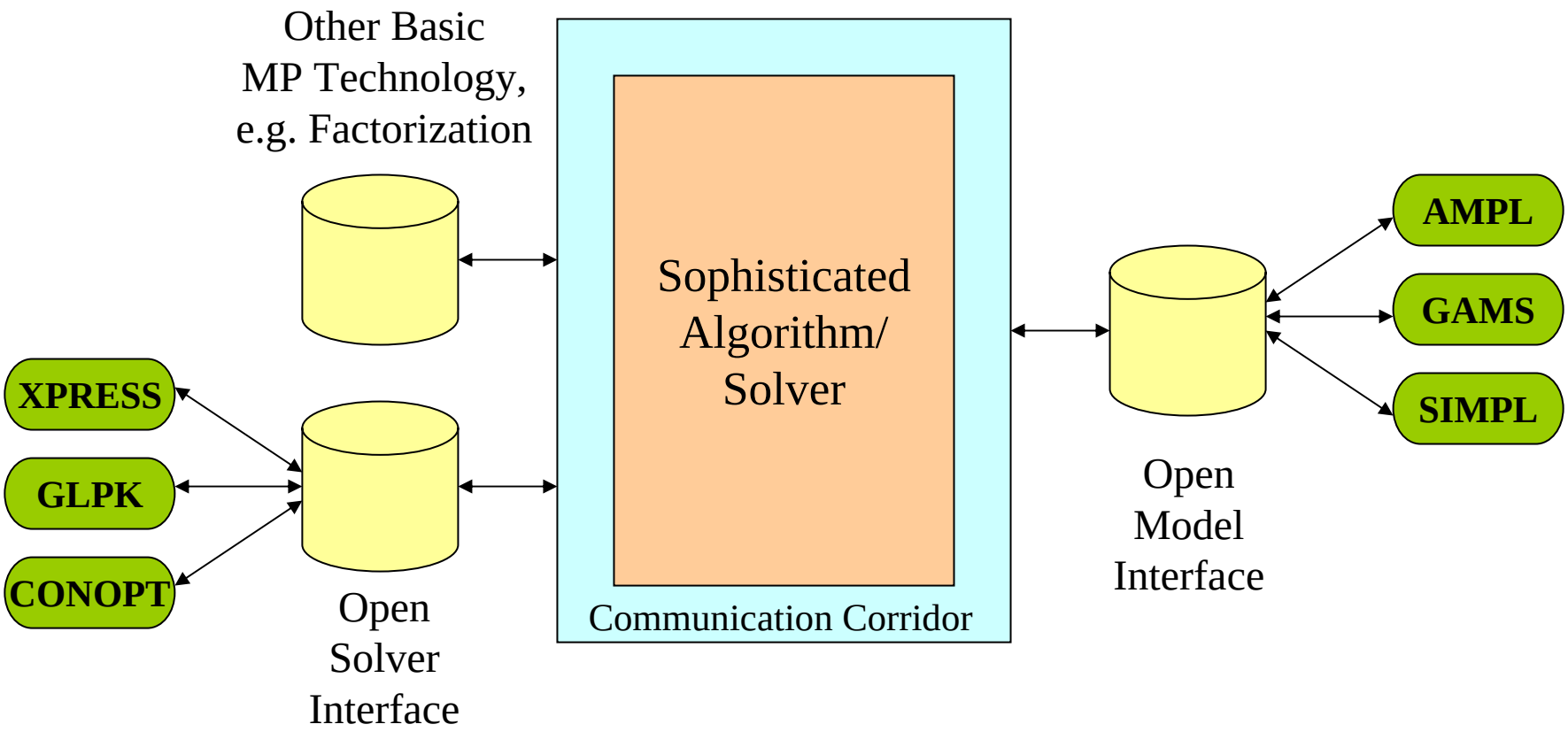
Difficulty = Opportunity

- OSI-NLP extensions could have great impact
- Improves area that needs it most
- GAMS interested in NLP standards
  - Reduce development/maintenance costs
  - Improve robustness and software quality
- Standards must be a cooperative effort
  - We have experience and links to developers
  - Large user base and model collections

Expert Users

- With interface to problem providers COIN could become *the* way of implementing complex algorithms (research & commercially)
- Sophisticated *solvers* use basic MP technology:
  - SBB (Arki/GAMS - B&B requiring NLP technology)
  - DICOPT (OA requiring NLP+MIP)
  - BARON (NLP+LP)
  - LogMip (NLP+MIP)
  - DEA (LP)

Architecture for Expert Users



Conclusions

- Overlap in COIN and GAMS agenda
- GAMS/COIN extends COIN user base
- Adds freebie solvers to GAMS solver stable
- Future work
  - Improved OSI MIP interface
  - Generic interface to problem providers
  - OSI NLP interface
- Potentially optimal architecture for expert users