GAMS: A High Performance Modeling System for Large-Scale Modeling Applications

Armin Pruessner
Alexander Meeraus

GAMS Development Corporation
Washington, DC

INFORMS - Denver
October 24, 2004

Agenda

- Overview of Modeling Systems
  - Why use Modeling Systems? (Past and Present)
  - Language Syntax
  - GAMS Features?
  - Solvers and Other Features
- Illustrative Examples
  - Solving Models and Quick Modifications
  - Interfacing with Office Applications
  - Full Web Integration Example
- Conclusions

Modeling Systems

Software systems for decision making:

- Optimization problems (max profit/min cost)
- Simplify the model building and solution process
- Create maintainable models
- Adapt models quickly to new situations

Modeling: Past

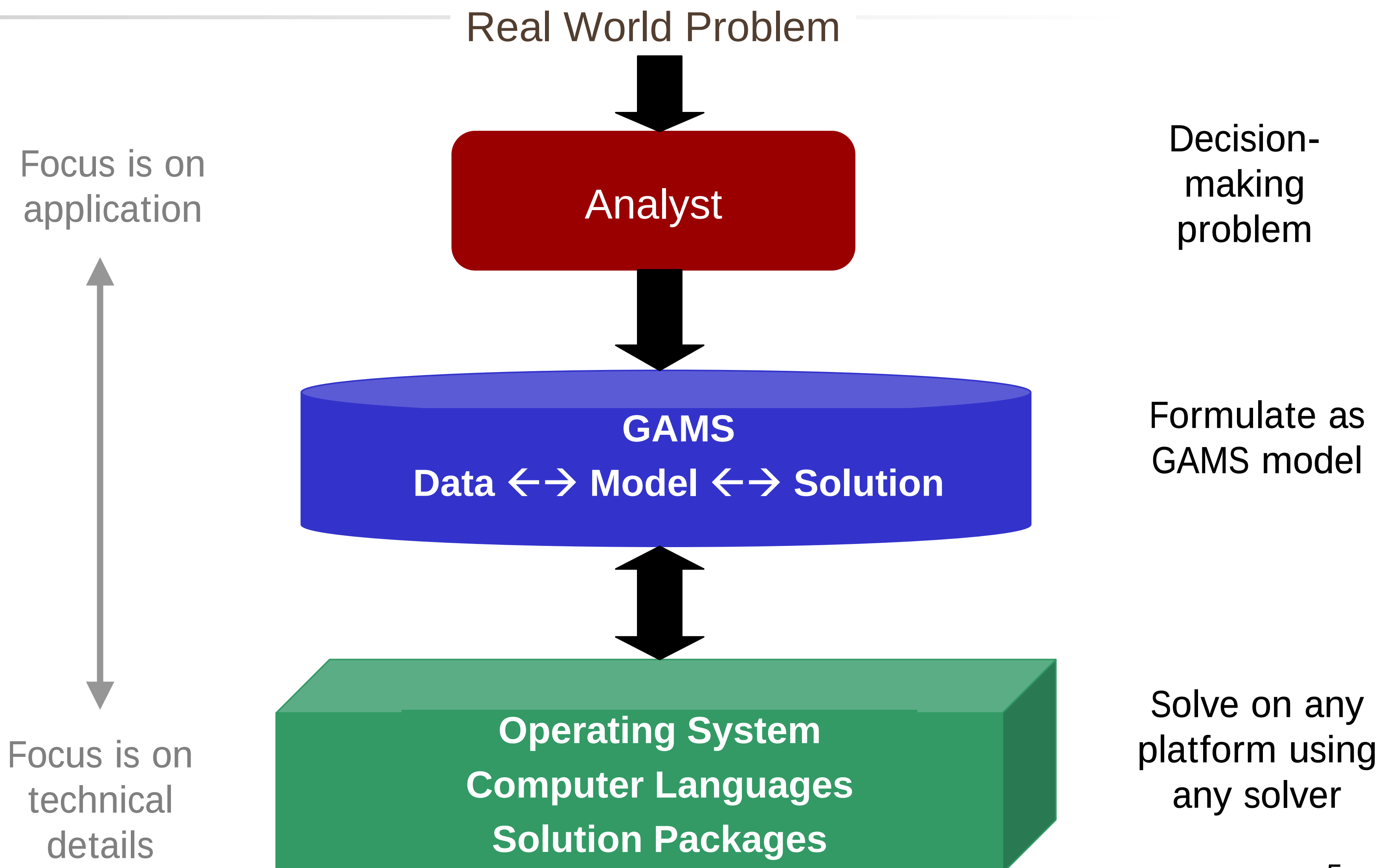
In the early years: Implementation (software) was cumbersome:

- Problem-specific
- Platform-specific
- Required high technical skills
- Models not easily maintainable or adaptable

Today: modeling languages are (roughly)

- a means of describing problems to a computer system in the same way that people describe those problems to each other.

Modeling Systems: Present



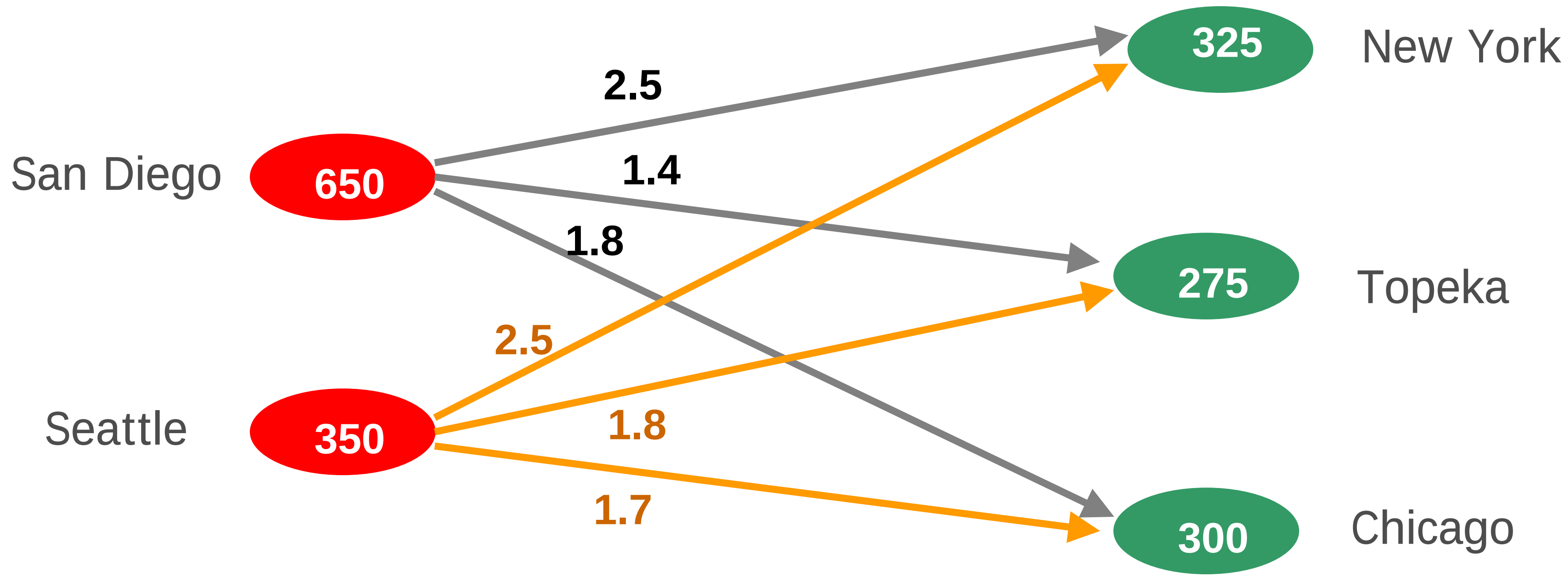
Basic Technical Principles

GAMS principles:

- Separation of model and solution methods
- Computing platform independence
- Multiple model types, solvers, platforms
- Balanced mix of declarative and procedural approaches
- Model is a data base operator and/or object

GAMS Language Example

Transport:



Minimize: **Transportation cost (distance & units)**
Subject to: **Demand satisfaction at markets**
Supply constraints

Transport Model (IDE)

gamside: C:\gamsprojects\conferences\informs2004\models\p.gpr - [C:\gamsprojects\conferences\informs2004\models\t1.gms]

File Edit Search Windows Utilities Help

m2.gms | m3.gms | m4.gms | m5.gms | portfolio_access.gms | portfolio_excel.gms | portfolio_jdbc.gms | t1.gms | trnsport.gms

```
$Offtext

 Sets
      i   canning plants    / seattle, san-diego /
      j   markets           / new-york, chicago, topeka / ;

 Parameters
      a(i)  capacity of plant i in cases,
      b(j)  demand at market j in cases;

 Parameter c(i,j)  transport cost in thousands of dollars per case ;

 Variables
      x(i,j)  shipment quantities in cases
      z       total transportation costs in thousands of dollars ;
 Positive Variable x ;

 Equations
      cost         define objective function
      supply(i)    observe supply limit at plant i
      demand(j)    satisfy demand at market j ;

 cost ..        z  =e=  sum((i,j), c(i,j)*x(i,j)) ;
 supply(i) ..   sum(j, x(i,j))  =l=  a(i) ;
 demand(j) ..   sum(i, x(i,j))  =g=  b(j) ;

 Model m1 /cost, supply, demand/ ;
 Solve m1 using lp minimizing z ;

 Display x.l, x.m ;
```

35: 78 Modified Insert

Model data structures and constraints (no data)

Transport Model With Data

gamside: C:\gamsprojects\conferences\informs2004\models\p.gpr - [C:\gamsprojects\conferences\informs2004\models\m1.gms]

File Edit Search Windows Utilities Help

m1.gms | m2.gms | m3.gms | m4.gms | m5.gms | portfolio_access.gms | portfolio_excel.gms | portfolio_jdbc.gms | t1.gms | trnsport.gms

```
Sets
     i   canning plants   / seattle, san-diego /
     j   markets          / new-york, chicago, topeka / ;

Parameters
     a(i)  capacity of plant i in cases
       /    seattle     350
            san-diego   600  /
     b(j)  demand at market j in cases
       /    new-york    325
            chicago     300
            topeka      275  / ;

Table d(i,j)  distance in thousands of miles
                  new-york       chicago      topeka
    seattle          2.5           1.7          1.8
    san-diego        2.5           1.8          1.4  ;

Scalar f  freight in dollars per case per thousand miles  /90/ ;
Parameter c(i,j)  transport cost in thousands of dollars per case ;
          c(i,j) = f * d(i,j) / 1000 ;

Variables
     x(i,j)  shipment quantities in cases
     z       total transportation costs in thousands of dollars ;
Positive Variable x ;

Equations
     cost        define objective function
     supply(i)   observe supply limit at plant i
     demand(j)   satisfy demand at market j ;
```

45: 68 Modified Insert

Model data structures and constraints with data.

Features

GAMS:

- Offers single point of support
- Large number of model types
- Multiple platforms
- State of the art solvers
- Large user base (10,000 of customers in over 100 countries)
- Large model library and public models
- GAMS User List

Multiple Model Types

Includes:

- LP - Linear Programming
- MIP - Mixed Integer Programming
- QCP - Quadratically Constrained Programming
- NLP - Nonlinear Programming
- CNS – Constrained Nonlinear Systems
- MINLP - Mixed Integer Nonlinear Programming
- MPEC - NLP with Complementarity Constraints
- MPSGE - General Equilibrium Models
- Stochastic Optimization

Supported Platforms

Solver/Platform availability - 21.4 September 6, 2004							
	Intel		Sun Sparc	HP 9000	DEC Alpha	IBM RS-6000	SGI
	Windows 95/98/Me/NT/2000/XP	Linux	Solaris	HP-UX 11	Digital Unix 4.0	AIX 4.3	IRIX
BARON 7.2	✔	✔				✔	
BDMLP	✔	✔	✔	✔	✔	✔	✔
CONOPT 3	✔	✔	✔	✔	✔	✔	✔
CONVERT	✔	✔	✔	✔	✔	✔	✔
CPLEX 9.0	✔	✔	✔	✔	8.1	✔	✔
DECIS	✔	✔	✔	✔	✔	✔	✔
DICOPT	✔	✔	✔	✔	✔	✔	✔
LGO	✔	✔	✔	✔	✔		✔
MILES	✔	✔	✔	✔	✔	✔	✔
MINOS	✔	✔	✔	✔	✔	✔	✔
MOSEK 3.0	✔	✔	✔	✔			
MPSGE	✔	✔	✔	✔	✔	✔	✔
MSNLP	✔	✔					
NLPEC	✔	✔	✔	✔	✔	✔	✔
OQNLP	✔	✔					
OSL V3	✔	✔	✔	V2		✔	V2
PATH	✔	✔	✔	✔	✔	✔	✔
PATHNLP	✔	✔	✔	✔	✔	✔	✔
SBB	✔	✔	✔	✔	✔	✔	✔
SNOPT	✔	✔	✔	✔	✔	✔	✔
XA	✔	✔	✔	✔	✔	✔	
XPRESS 2004	✔	✔	✔			✔	

For backward compatibility we maintain older versions of operating systems and solvers. Please call.

Supported Solvers

BARON	Branch-And-Reduce Optimization Navigator for proven global solutions from The Optimization Firm
BDMLP	LP solver that comes with any GAMS system
COIN	Link to the solvers in the COIN-OR project (Computational Infrastructure - Operations Research).
CONOPT	Large scale NLP solver from ARKI Consulting and Development
CONVERT	Frame work for translating models into scalar models of other languages
CPLEX	High-performance LP/MIP solver from Ilog
DECIS	Large scale stochastic programming solver from Stanford University
DICOPT	Framework for solving MINLP models. From Carnegie Mellon University
EXAMINER	A tool for examining solution points and assessing their merit
GAMSBAS	A Program for Saving an Advanced Basis for GAMS
GAMSCHK	A System for Examining the Structure and Solution Properties of Linear Programming Problems Solved using GAMS
LGO	Lipschitz global optimizer from Pinter Consulting Services
MILES	MCP solver from University of Colorado at Boulder that comes with any GAMS system
MINOS	NLP solver from Stanford University
MOSEK	Large scale LP/MIP plus conic and convex non-linear programming system from EKA Consulting
MPSGE	Modeling Environment for CGE models from University of Colorado at Boulder
MPSWRITE	MPS file generator that comes with any GAMS System
MSNLP	Multi-start method for global optimization from Optimal Methods Inc.
NLPEC	MPEC to NLP translator that uses other GAMS NLP solvers
OQNLP	Multi-start method for global optimization from Optimal Methods Inc.
OSL	High performance LP/MIP solver from IBM
OSLSE	OSL Stochastic Extensions for solving stochastic models
PATH	Large scale MCP solver from University of Wisconsin at Madison
PATHNLP	Large scale NLP solver for convex problems from University of Wisconsin at Madison
SBB	Branch-and-Bound algorithm from ARKI for solving MINLP models
SCENRED	A tool for the reduction of scenarios modeling the random data processes
SNOPT	Large scale SQP based NLP solver from Stanford University
XA	Large scale LP/MIP system from Sunset Software
XPRESS	High performance LP/MIP solver from Dash

29 supported solvers (including *global solvers*) plus several **contributed** plug and play solvers

[Live Software Demo]

- Using the GAMS IDE
- Modifying models to handle new situations

Model trnsport.gms - LP

```
Sets
     i   canning plants   / seattle, san-diego /
     j   markets          / new-york, chicago, topeka / ;

Parameters

     a(i)  capacity of plant i in cases
       /    seattle     350
            san-diego   600  /

     b(j)  demand at market j in cases
       /    new-york    325
            chicago     300
            topeka      275  / ;

Table d(i,j)  distance in thousands of miles
                  new-york       chicago      topeka
    seattle          2.5           1.7          1.8
    san-diego        2.5           1.8          1.4  ;

Scalar f  freight in dollars per case per thousand miles  /90/ ;

Parameter c(i,j)  transport cost in thousands of dollars per case ;

          c(i,j) = f * d(i,j) / 1000 ;
```

Solve using “s=m1” option

(to save model info)

Min/Max Shipments – MIP

Add additional constraints: (m2.gms)

```
Min/max shipments

Scalars   xmin  / 100 /,
          xmax  / 275 /;

Binary variables    ship(i,j)           decision variable to ship

Equations           minship(i,j)        minimum shipments
                    maxship(i,j)        maximum shipments ;

Minship(i,j)…       x(i,j) =g= xmin*ship(i,j);

Maxship(i,j)…       x(i,j) =l= xmax*ship(i,j);

Model m2 / cost, supply, demand, minship, maxship /;

Solve m2 using mip minimizing z;
```

Restart using “r=m1” option

Nonlinear Cost – NLP

Change objective function: (m3.gms)

```
Scalar              beta;

Equations           nlcost)             Nonlinear cost function ;

Nlcost…             z =e=  sum( (i,j),  c(I,j)*x(i,j)**beta);

Model m3 / nlcost, supply, demand /;

Beta = 1.5;
Solve m3 using nlp minimizing z;

Beta = 0.6;
Solve m3 using nlp minimizing z;
```

Restart using “r=m1” option

Min/max and NL Cost – MINLP

Add both the nonlinear function nlcost and the max/min shipment constraints: (m4.gms)

```
Model m4 / nlcost, supply, demand, minship, maxship /;

Option minlp = baron;

Solve m4 using minlp minimizing z;


Option minlp = sbb;  Option nlp=snopt;
Option optcr=0;

Solve m4 using minlp minimizing z;
```

Restart using “r=m1” option

[Done With Demo]

Other Language Features

- For and while loops
- If – else constructs
- Unix-like data manipulation utilities for Windows (grep, awk, sed, etc.)
- Statistical functions (distributions)

GAMS

Change in Focus (Future)

Computation
Past

- **Algorithm limits** applications
- Problem representation is low priority
- Large costly projects
- Long development times
- Centralized expert groups
- High computational cost, mainframes

- **Users left out**

Model
Present

- **Modeling skill limits** applications
- Algebraic model representation
- Smaller projects
- Rapid development
- Decentralized modeling teams
- Low computational cost, workstations
- Machine independence

- **Users involved**

Application
Present/Future

- **Domain expertise limits** application
- Off-the-shelf graphical user interfaces
- Links to other types of models
- Models embedded in business applications
- New computing environments
- Internet/web

- **User hardly aware of model**

New Challenges

Model only *small component* of a complex application:

Particular emphasis on

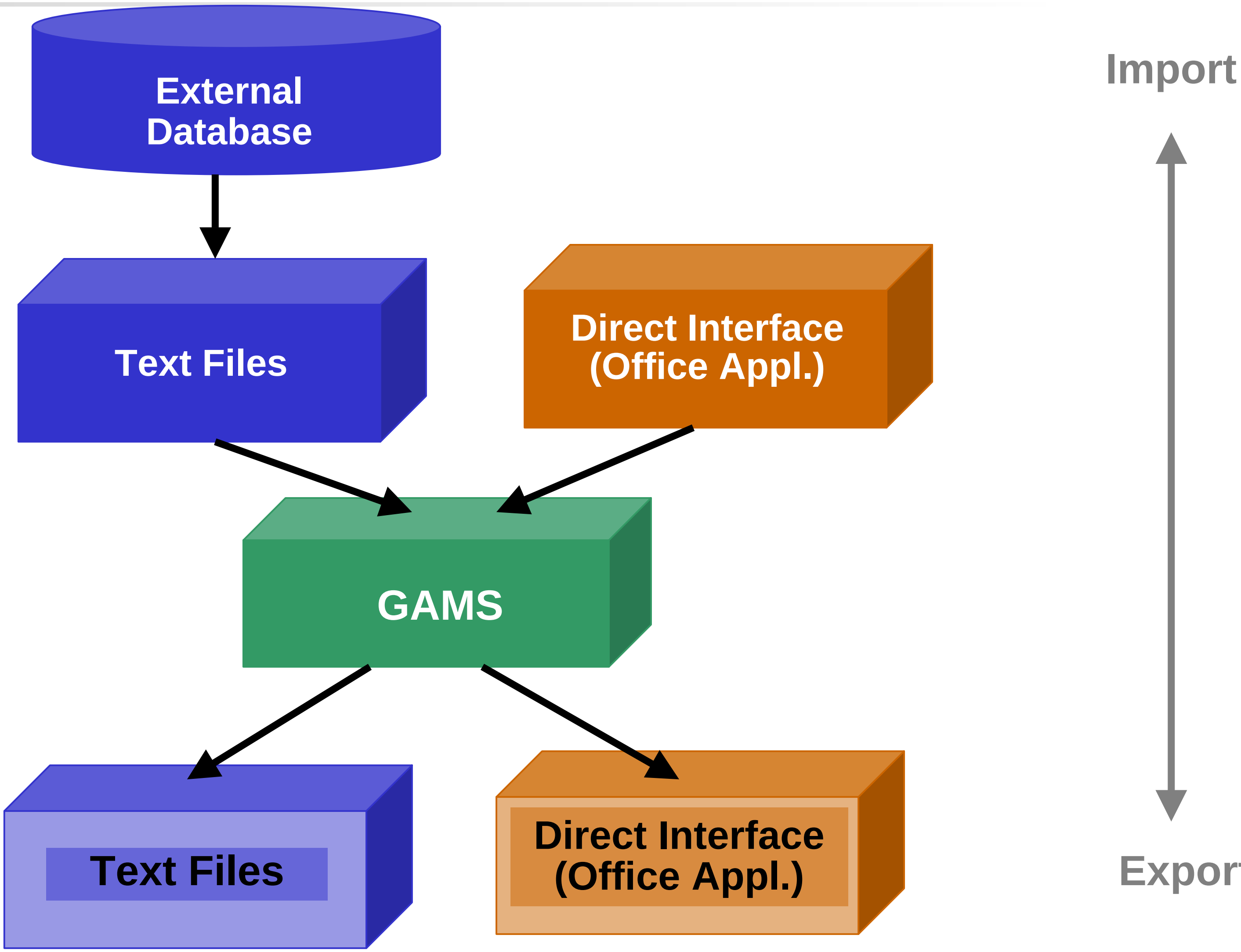
- Need to interface with databases
- Interface with other analytic/visualization engines
- Embed in web-based applications

Interfacing

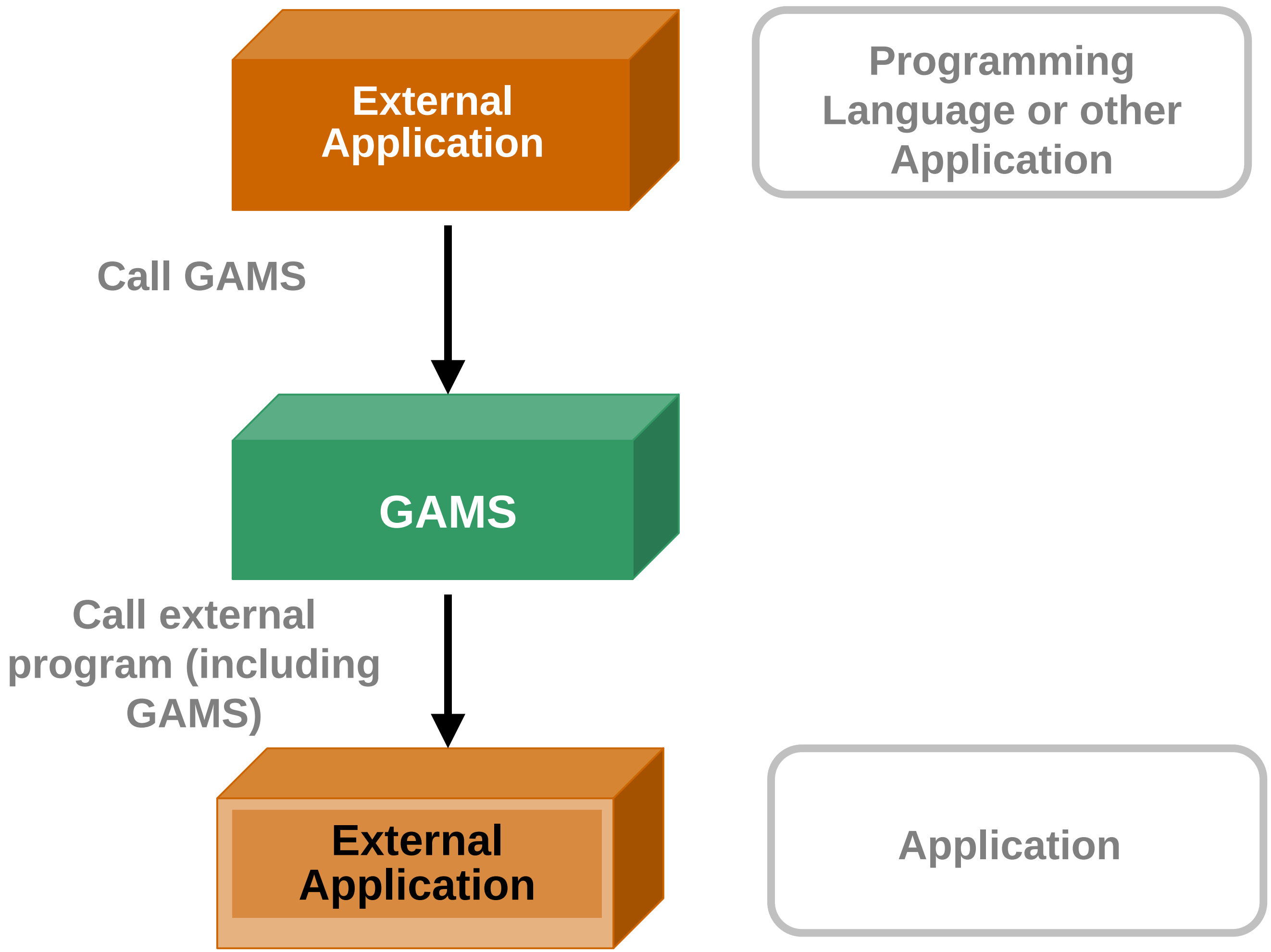
GAMS offers full range of interfacing capabilities:

- Data Import/Export from *Standard Applications*
  - MS Office, Databases, Text Files,...
- Execute programs from within GAMS (including another GAMS job itself)
- Call GAMS from complex applications
- Visualization tools

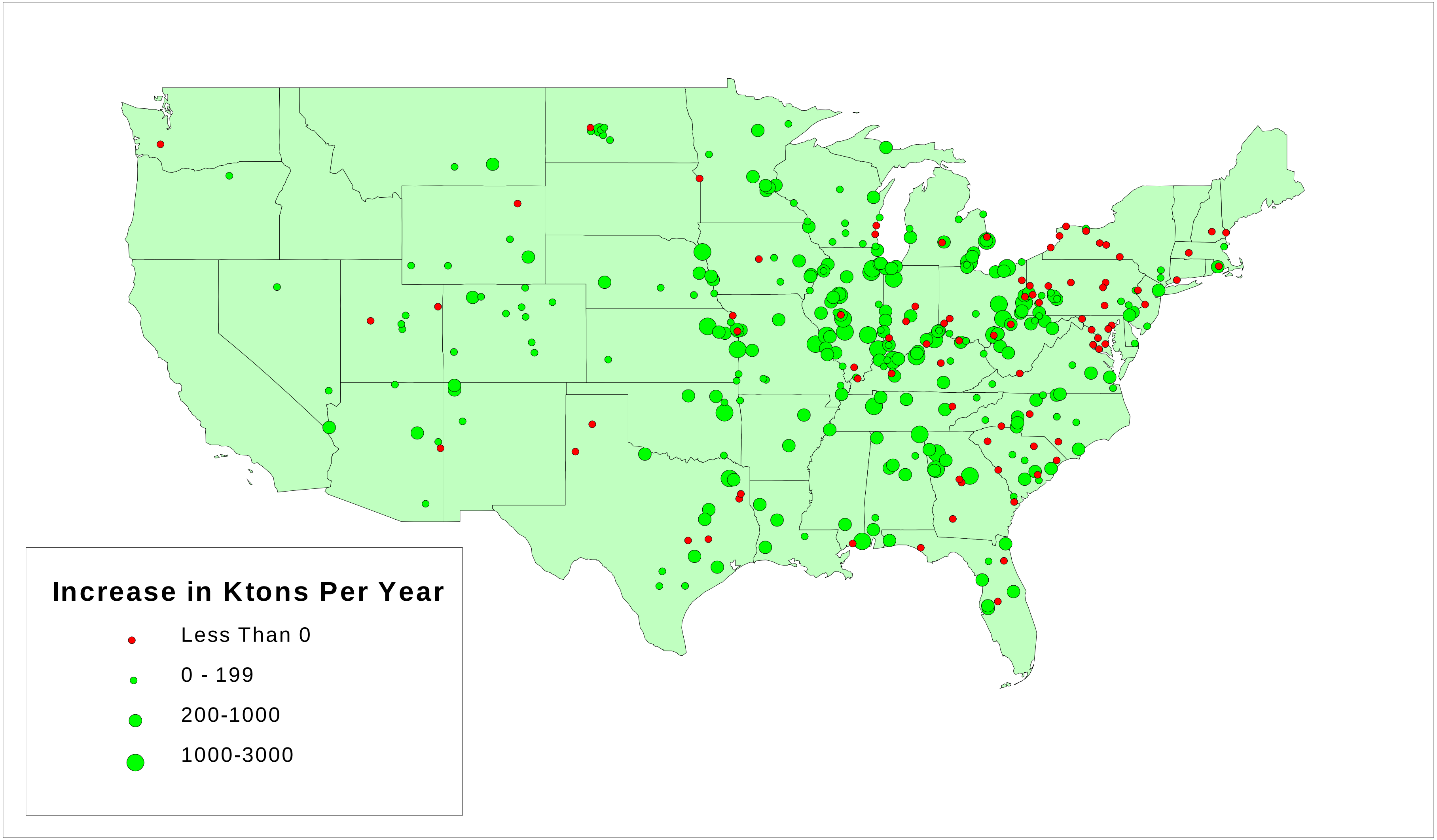
Scenario: Import/Export Data



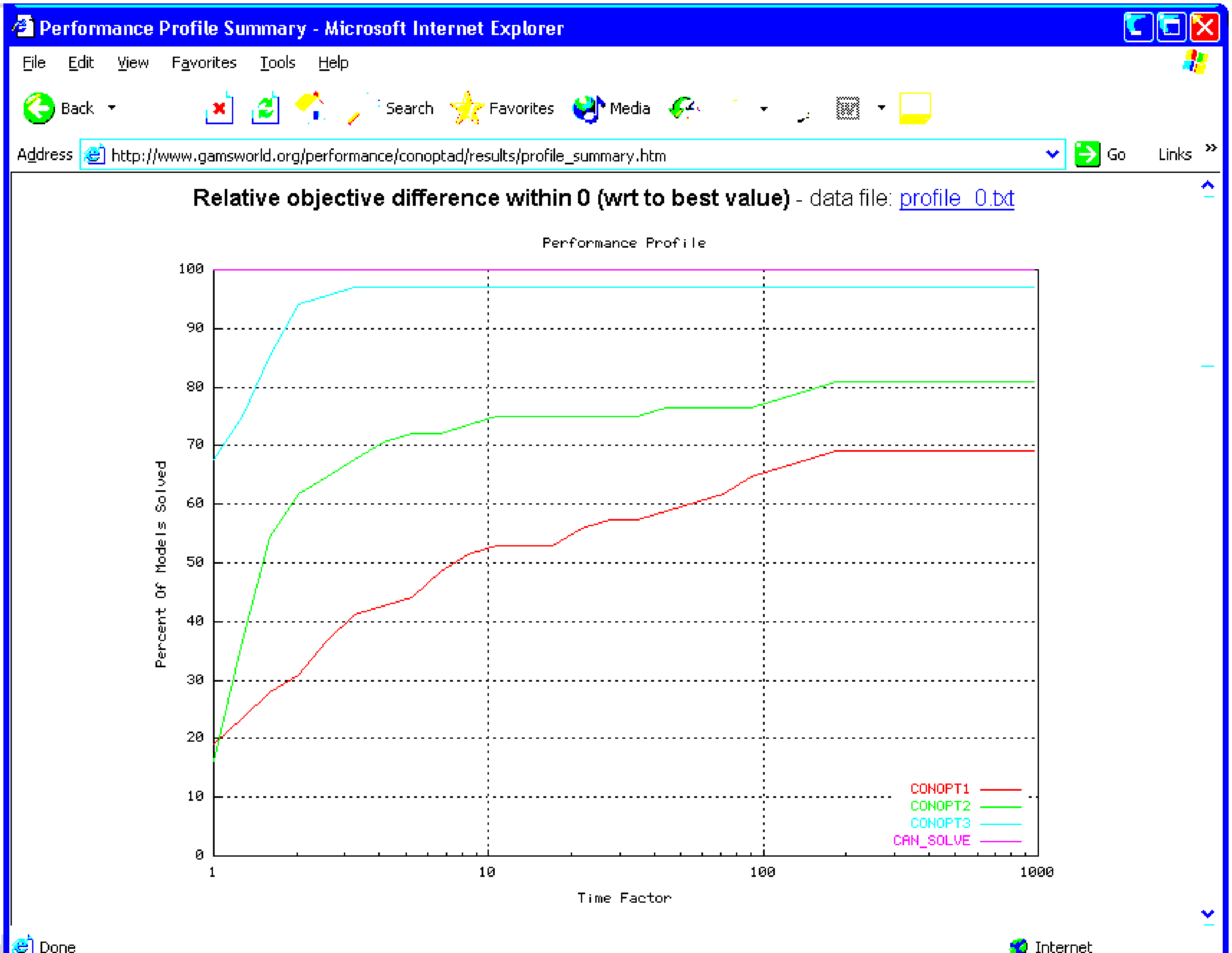
Embedding


External Application
Programming Language or other Application
Call GAMS
GAMS
Call external program (including GAMS)
External Application
Application

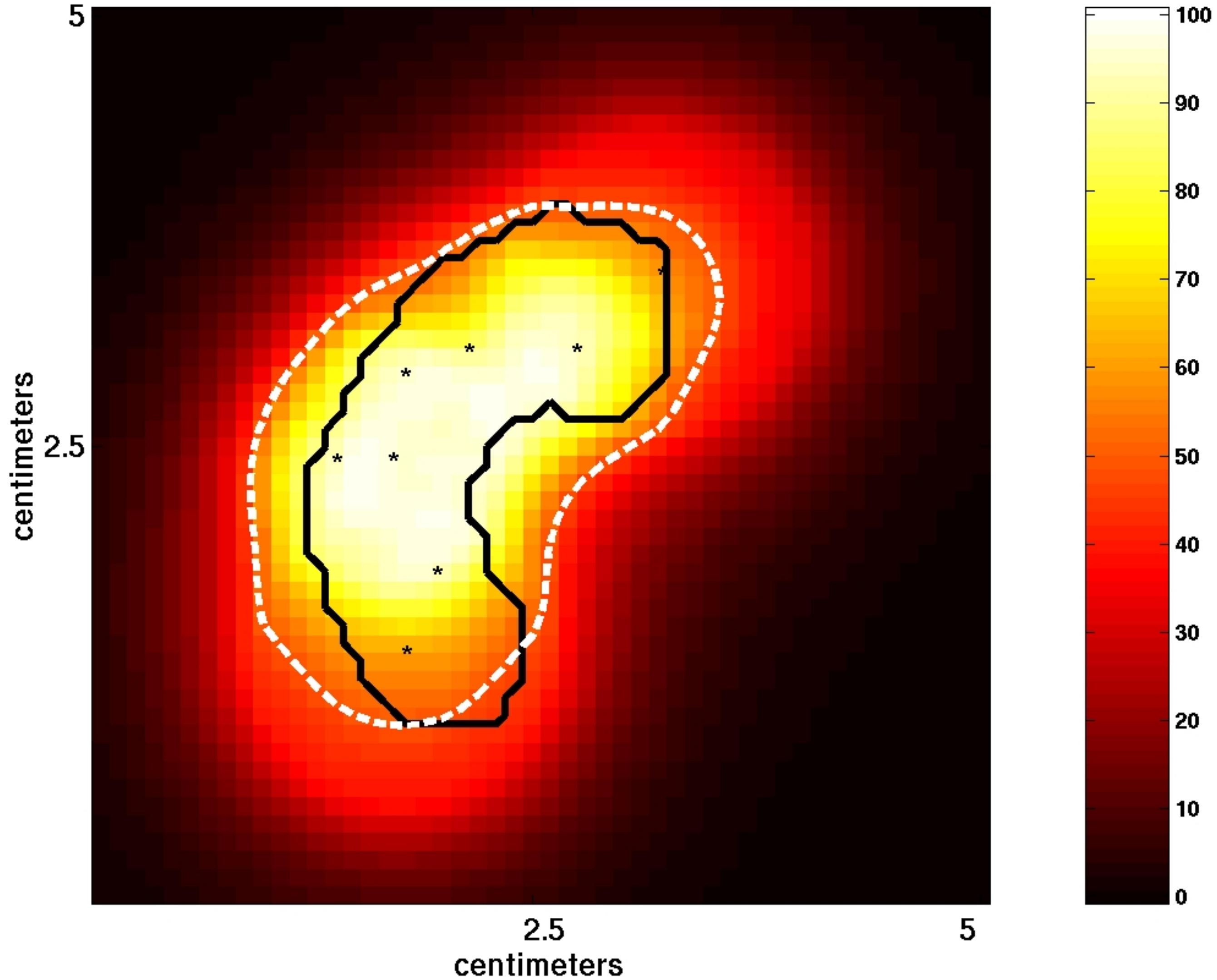
Examples – MapInfo



Examples - GnuPlot



Examples – GAMS/MATLAB

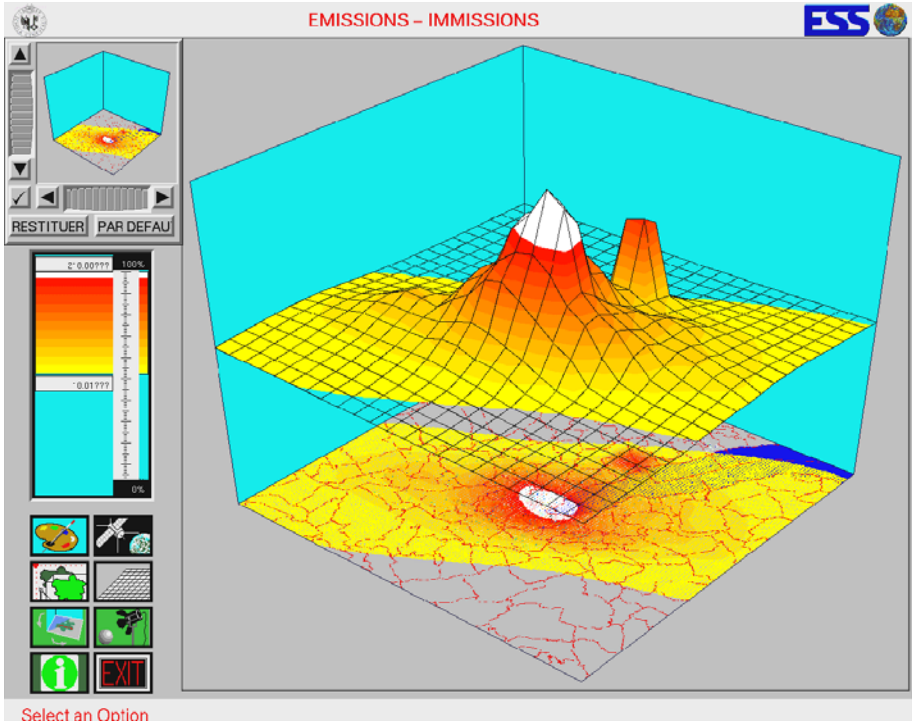


Examples – GIS Mapping

Total average yearly **NO_x** immission

Coupling GIS databases with models permits efficient data handling, preparation of scenarios, displays of results, animations.

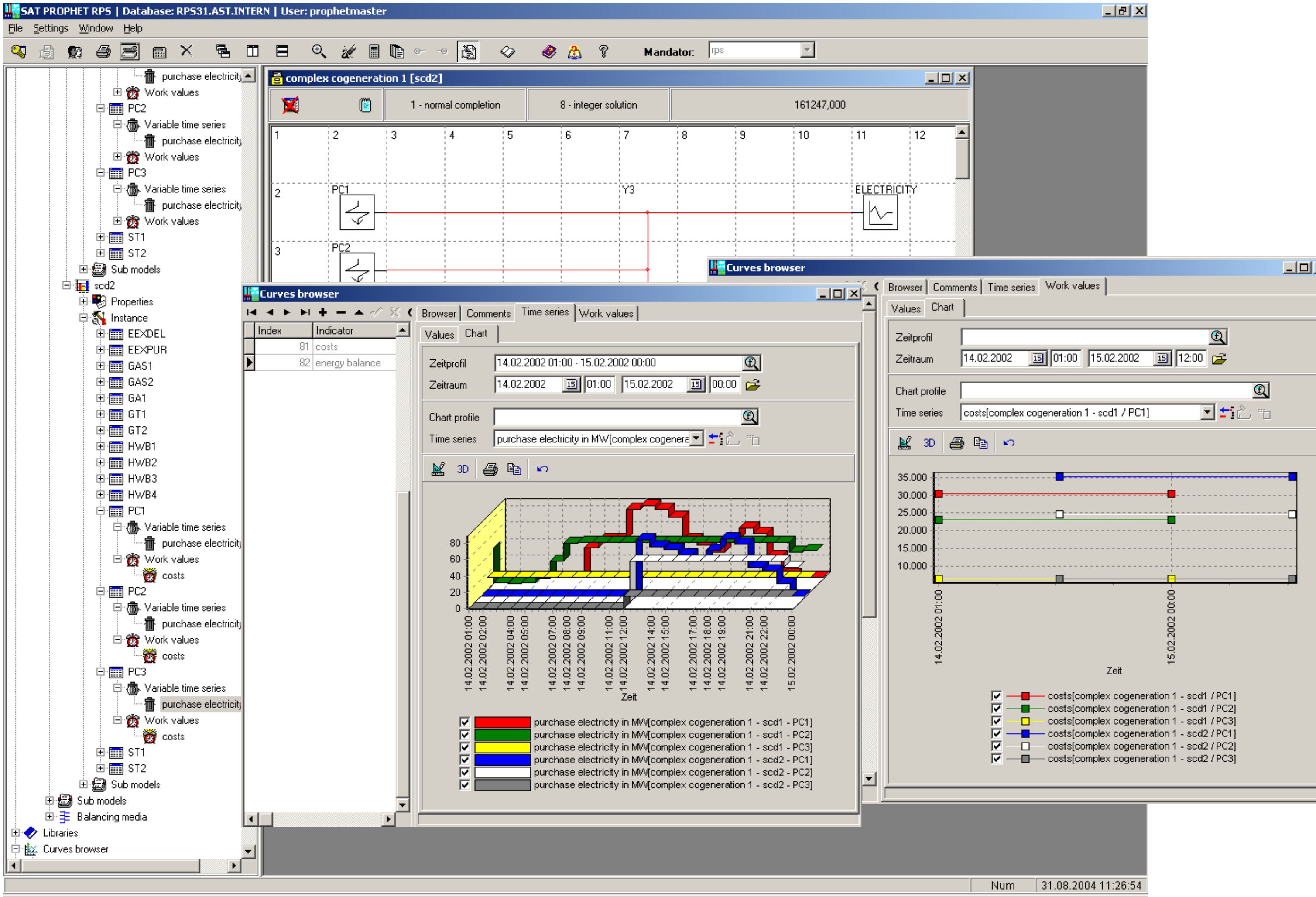
GIS computations permit the evaluation of spatially related costs or technical coefficients.



scale: 10 μg/m^3-200 μg/m^3

Various Front Ends

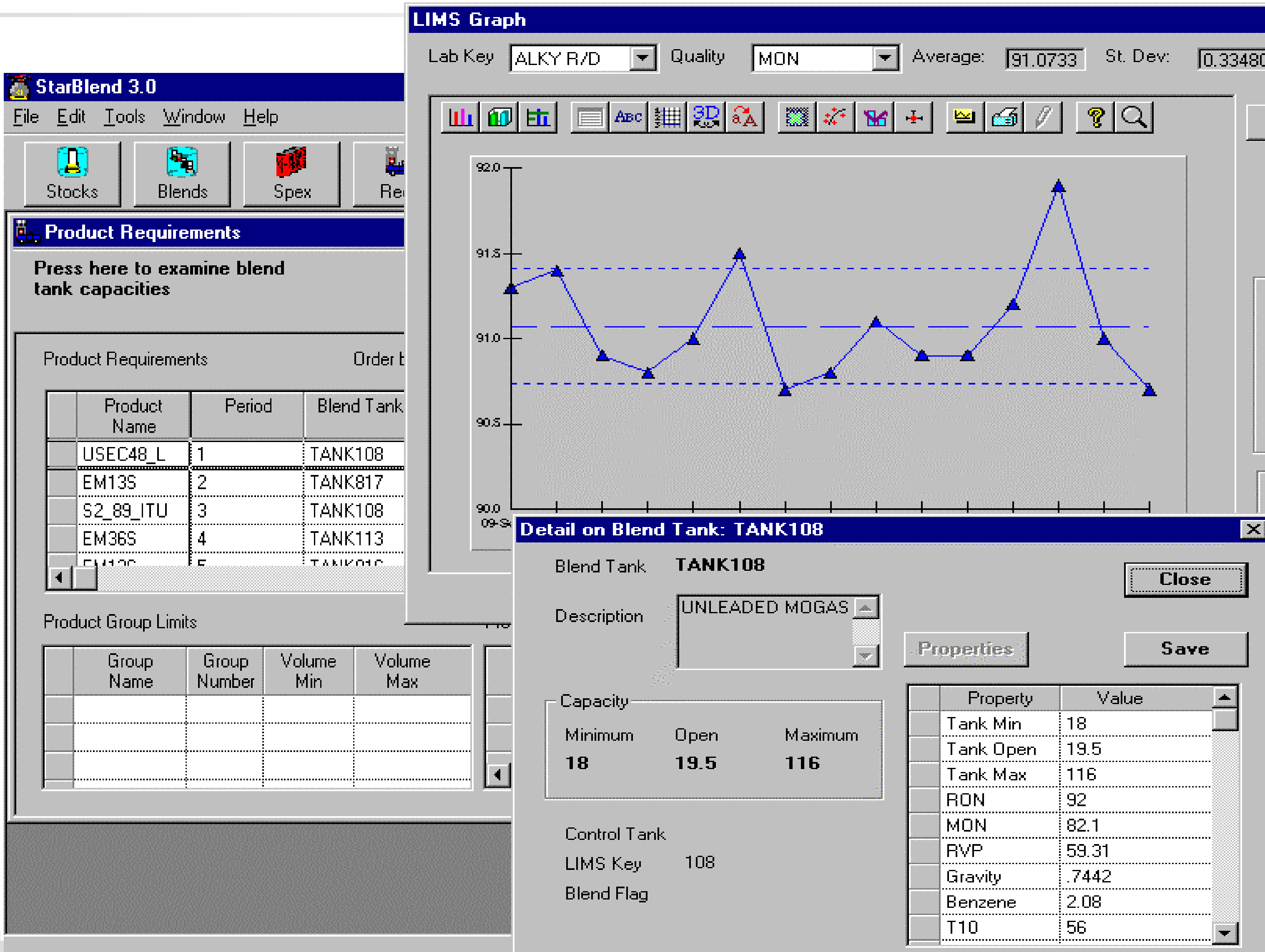
Client Model: Electricity Market



SAT-PROPHET

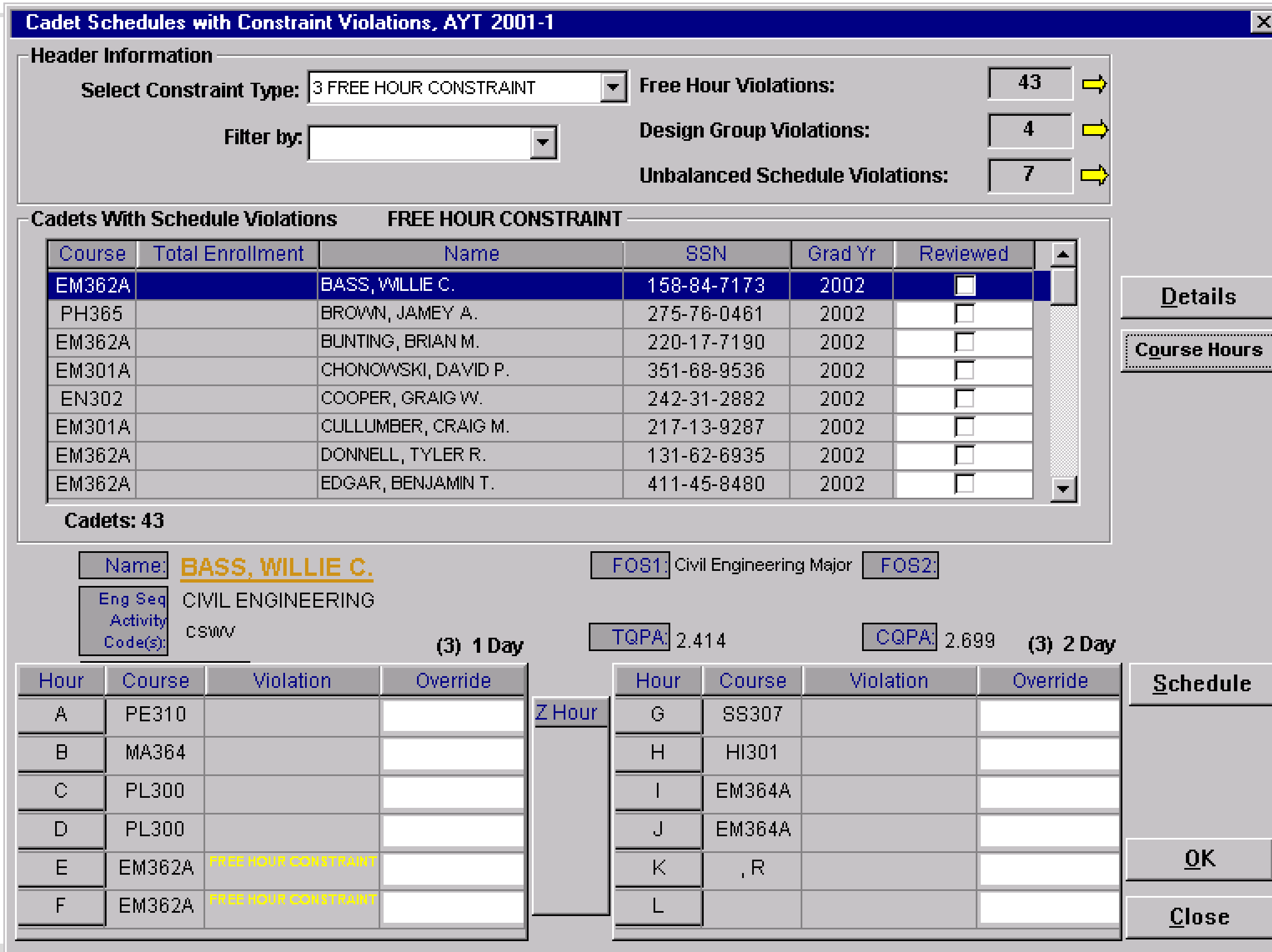
Various Front Ends

Client Model: Gasoline Blending



Various Front Ends

Client Model: Course Scheduling at West Point



Illustrative Examples

- Writing to Excel
- Reading from Access
- Embedding in another application

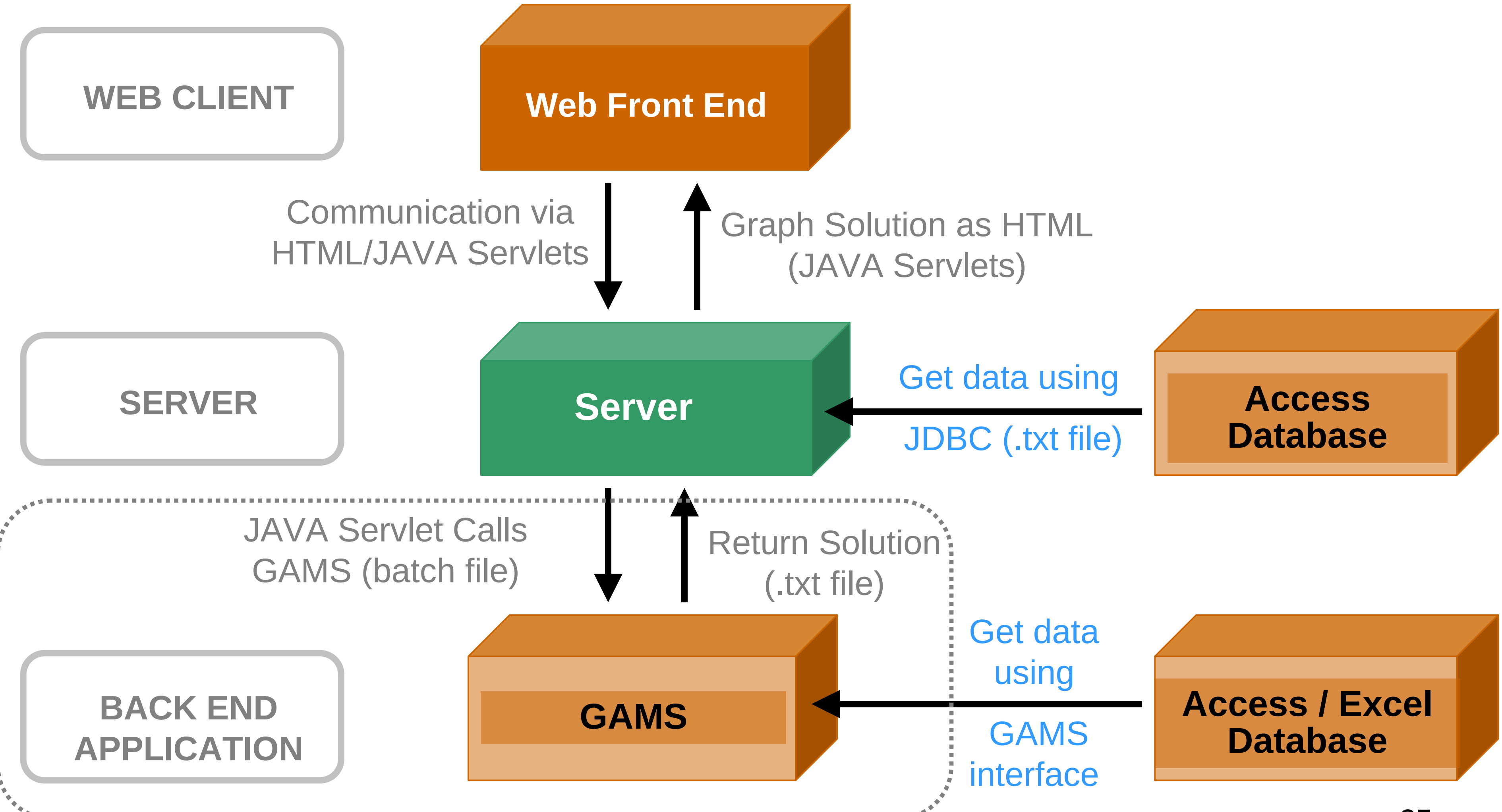
Real-world scenario

- Likely will involve the model being only a small component of the application

Excel Example

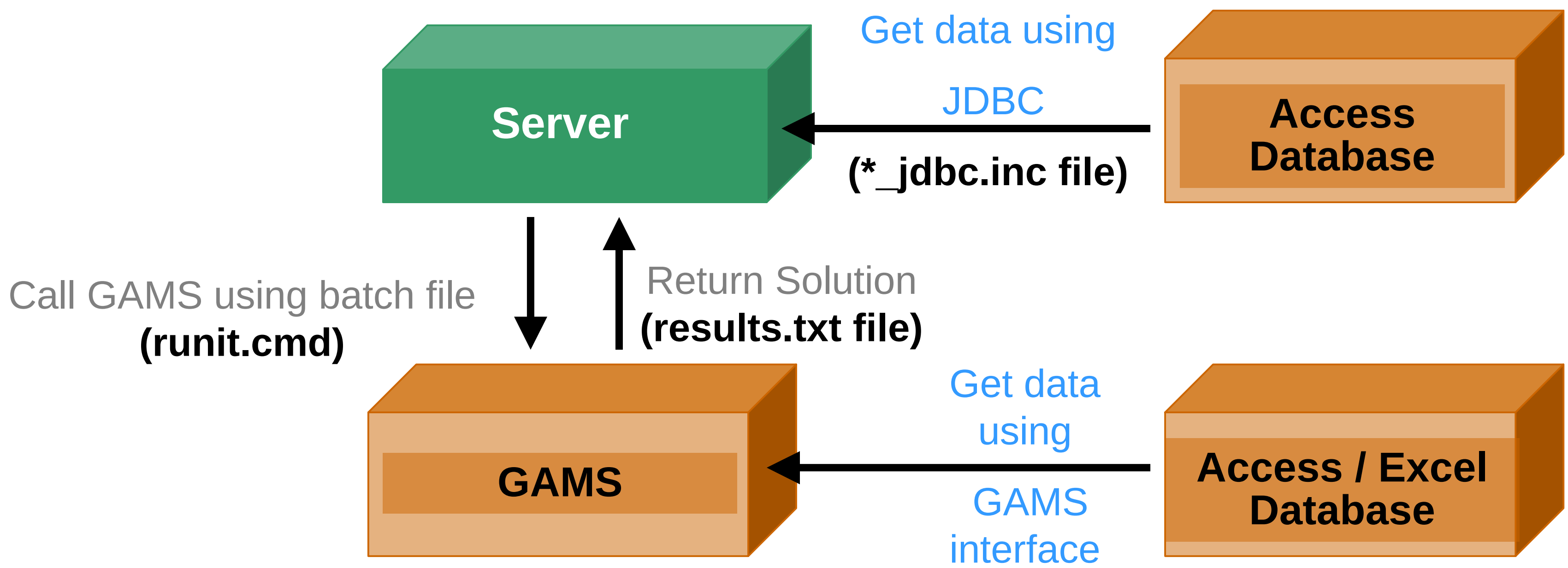
		lp	mip	nlp-convex	nlp-noncon	minlp
seattle	new-york	50	150	142.38408		132.5577
seattle	chicago	300	200	130.92994	300	117.4423
seattle	topeka			76.685987		100
san-diego	new-york	275	175	182.61592	325	192.4423
san-diego	chicago		100	169.07006		182.5577
san-diego	topeka	275	275	198.31401	275	175

Fully Embedded Application



Communication via .txt files

Comma-delimited format most common for large scale applications:



Sample File Formats

runit.cmd (Server executes GAMS)

```
cd C:\gamsprojects\conferences\informs2004\models\
C:\gams\21.4\gams portfolio_%1.gms lo=3 pf=pf.inc
REM in case of problems send zip file to administrator
C:\gams\21.4\gmszip log.zip portfolio_%1.gms portfolio_%1.lst *.log *.inc
```

value_jdbc.inc (Database sample input to GAMS)

```
GAB,951127,9.25
GAB,951128,9.25
GAB,951129,9.25
GAB,951130,9.5
```

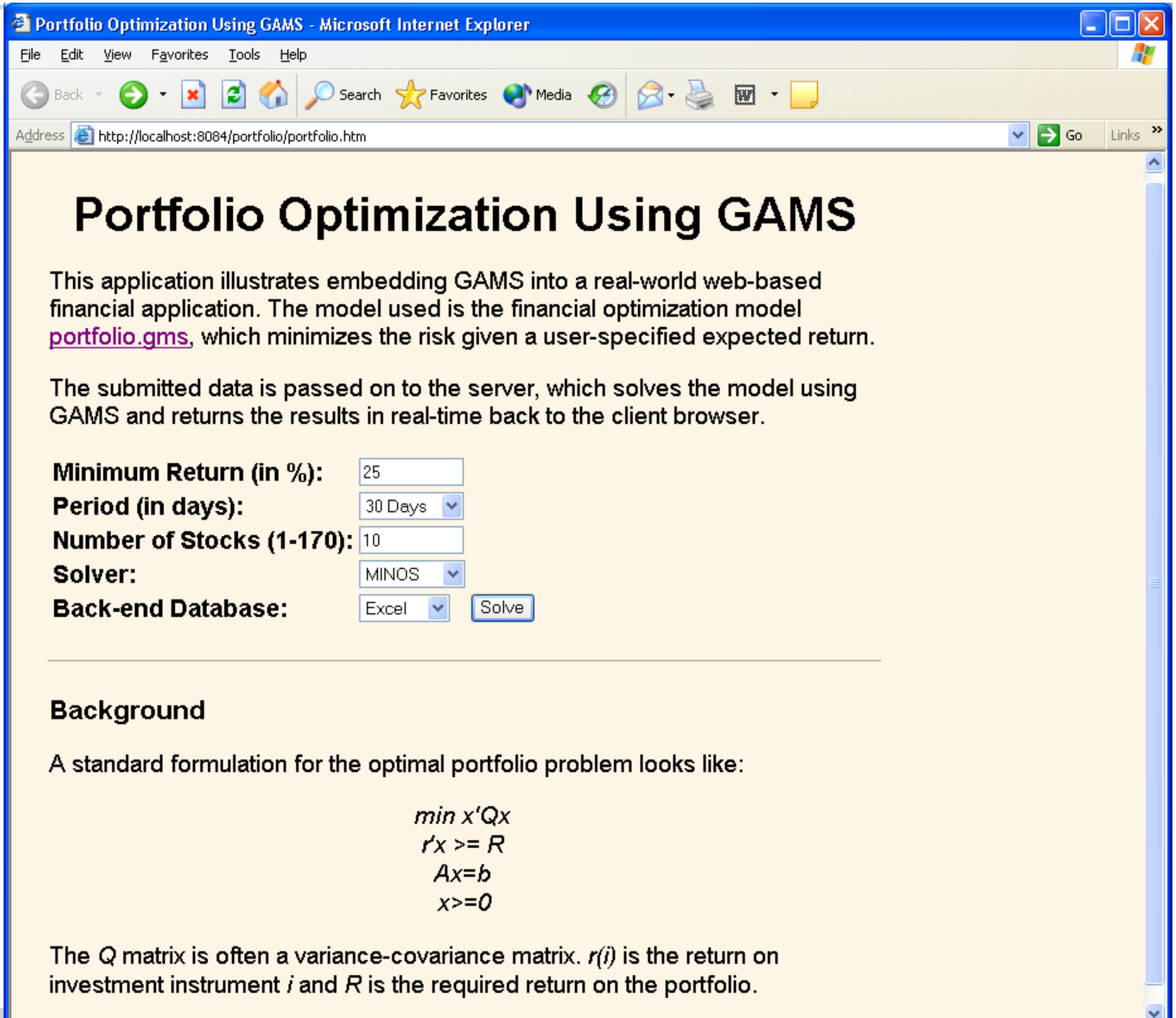
results.txt (GAMS solution input to server)

```
MODELSTATUS,2.00
SOLVERSTATUS,1.00
GAB,28.05
GAP,0.00
GDW,0.00
GE,0.00
```

[Live Software Demo]

- Writing to Excel (m5.gms)
- Reading from Access (portfolio_access.gms)
- Web application (portfolio.htm using a local server)

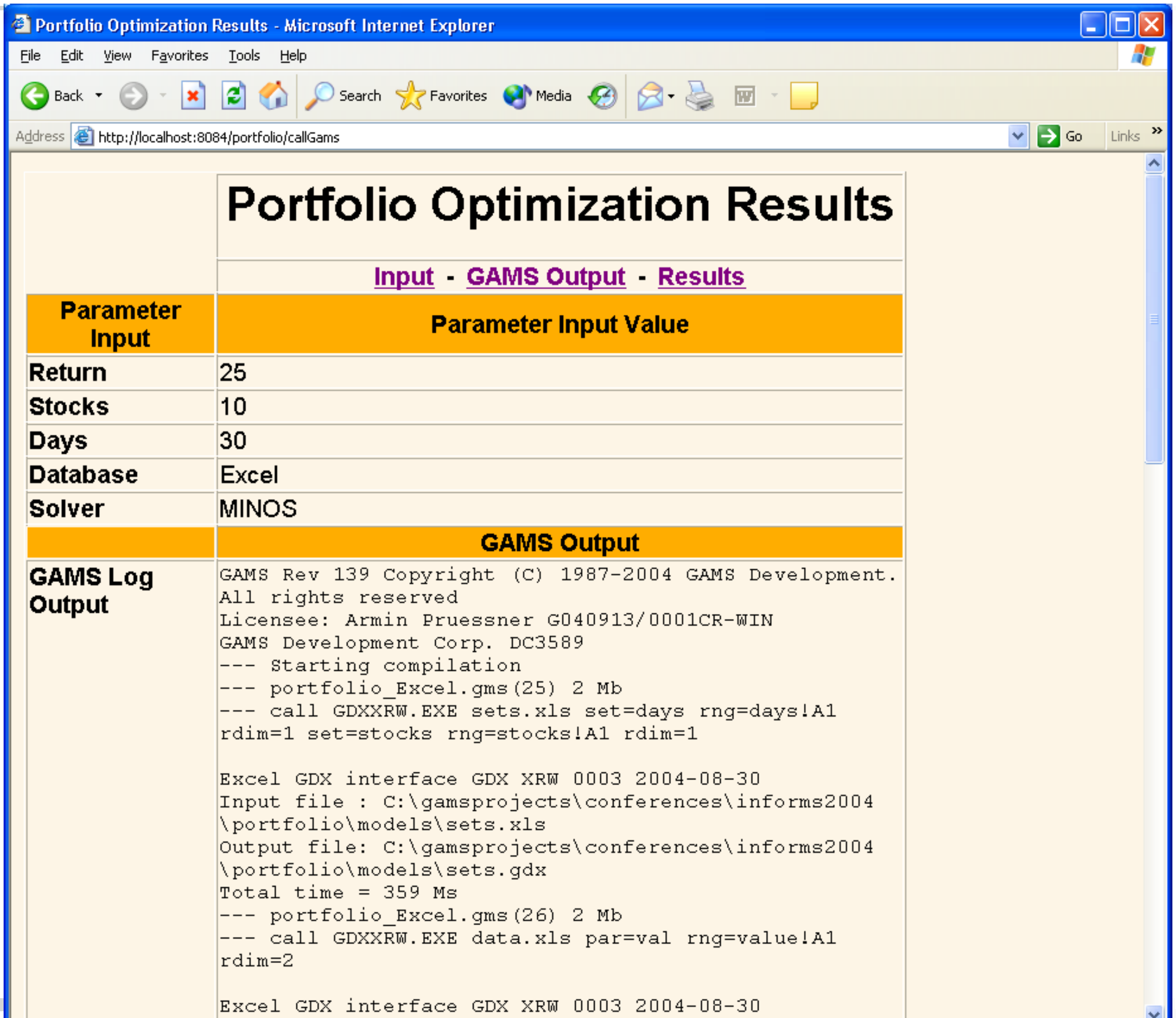
Web Front End



User may specify:

- Expected Return
- Investment Period (Days)
- Number of Stocks
- Solver
- Database (Excel or Access using GAMS interface or Access using JAVA JDBC)

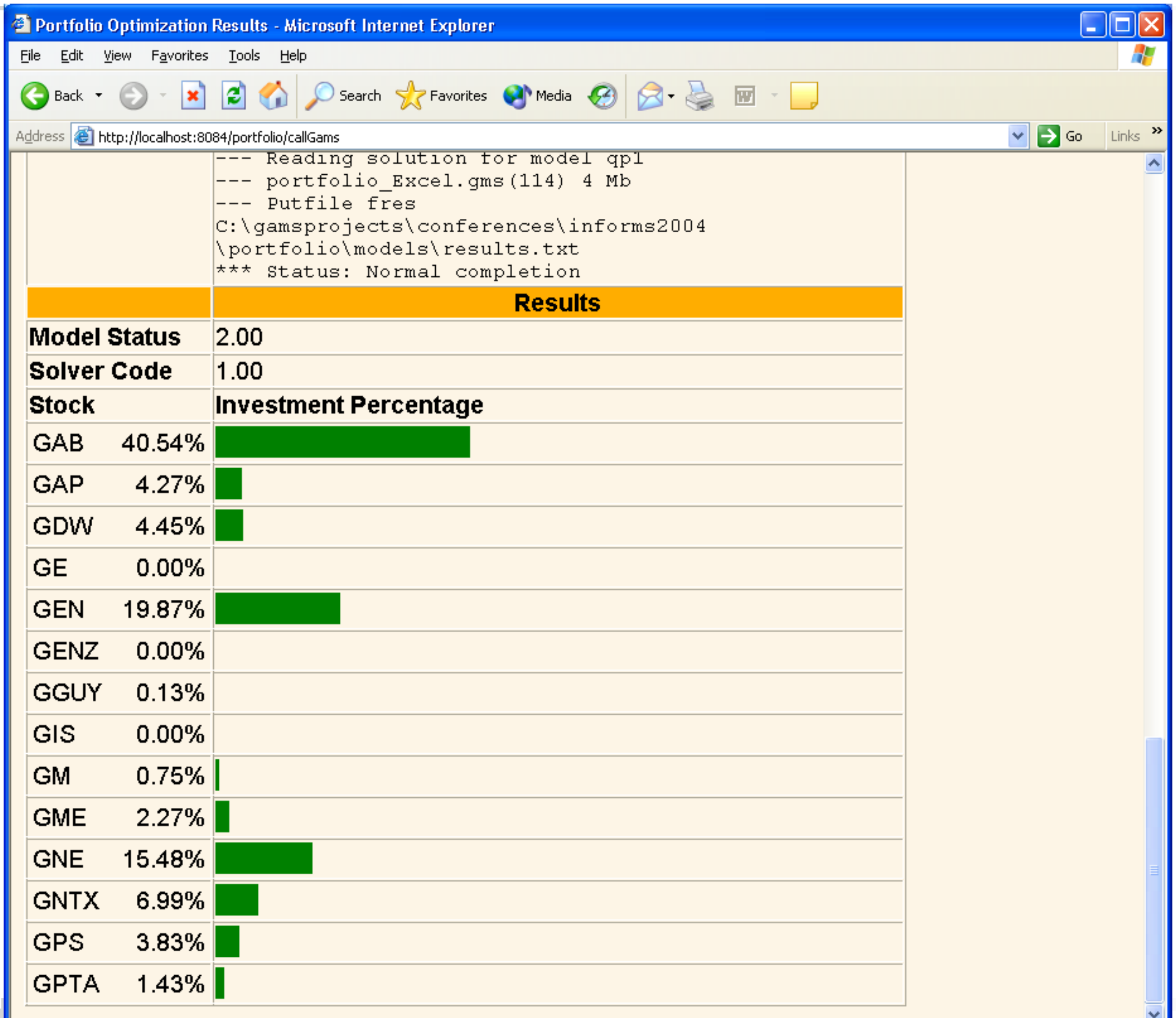
Output



Output:

- User input parameters
- Real-time GAMS output of model solve

Output (continued)



Output:

- GAMS Return Codes
- Investment distribution
- Graphical output

[Done With Demo]

Download GAMS

Free fully functional demo available for download

www.gams.com/download

Contains all available GAMS solvers.

Contacting GAMS

In the US:

GAMS Development Corporation
1217 Potomac Street NW
Washington, DC 20007
Phone:(202) 342-0180

General Information and Sales: sales@gams.com
Technical Support: support@gams.com

In Europe:

GAMS Software GmbH
Eupener Str. 135-137
50933 Cologne
Germany
Phone +49 (221) 949-9170
Information/Sales/ Technical Support: info@gams.de