RWTH AACHEN UNIVERSITY

BUSINESS ANALYTICS AND OPTIMIZATION

INTERNATIONAL CONFERENCE ON
OPERATIONS RESEARCH 2014 IN AACHEN

SEPTEMBER 2-5, 2014

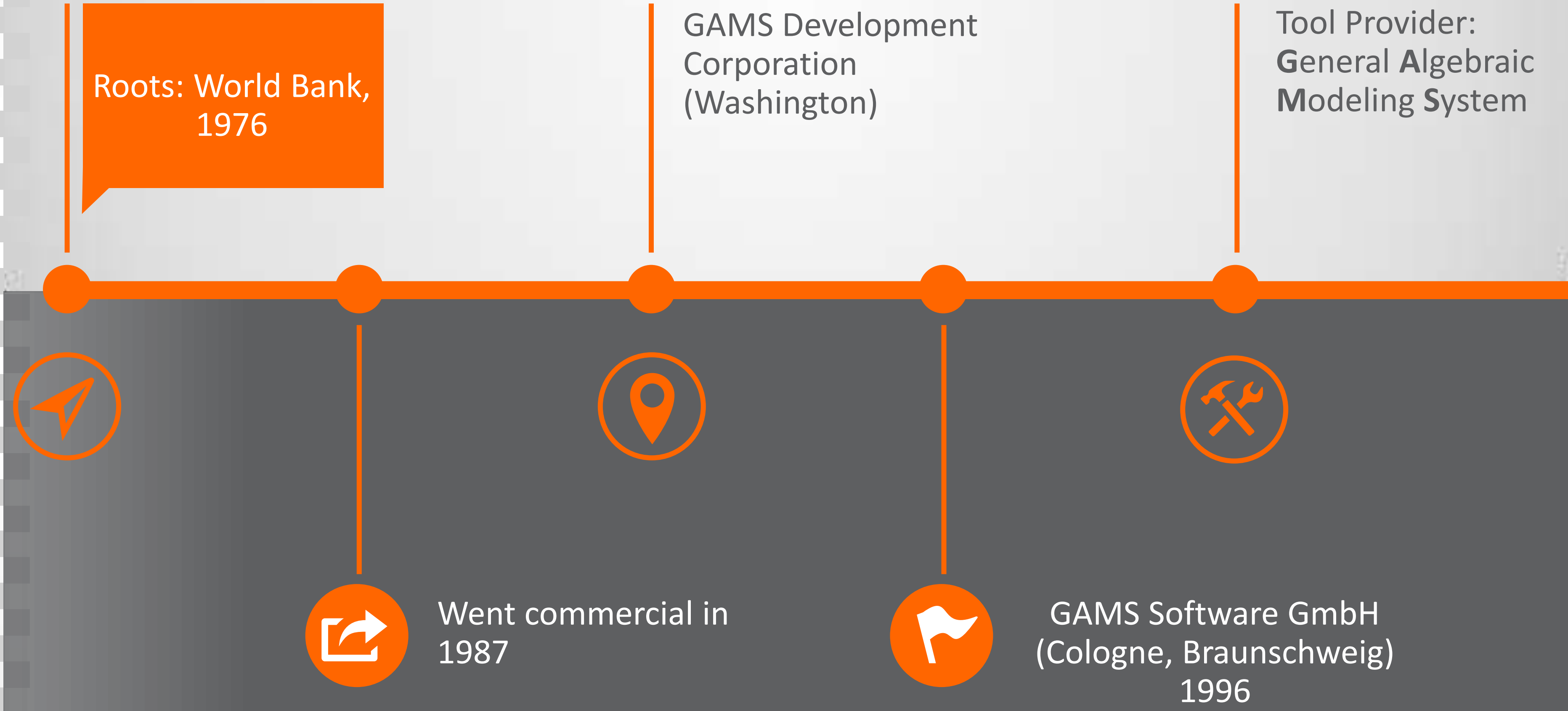
GOR

GAMS

Design Principles that Make the Difference

Franz Nelissen: FNelissen@gams.com

Company Background



Roots: World Bank, 1976

Went commercial in 1987

GAMS Development Corporation (Washington)

GAMS Software GmbH (Cologne, Braunschweig) 1996

Tool Provider: **G**eneral **A**lgebraic **M**odeling **S**ystem

Agenda

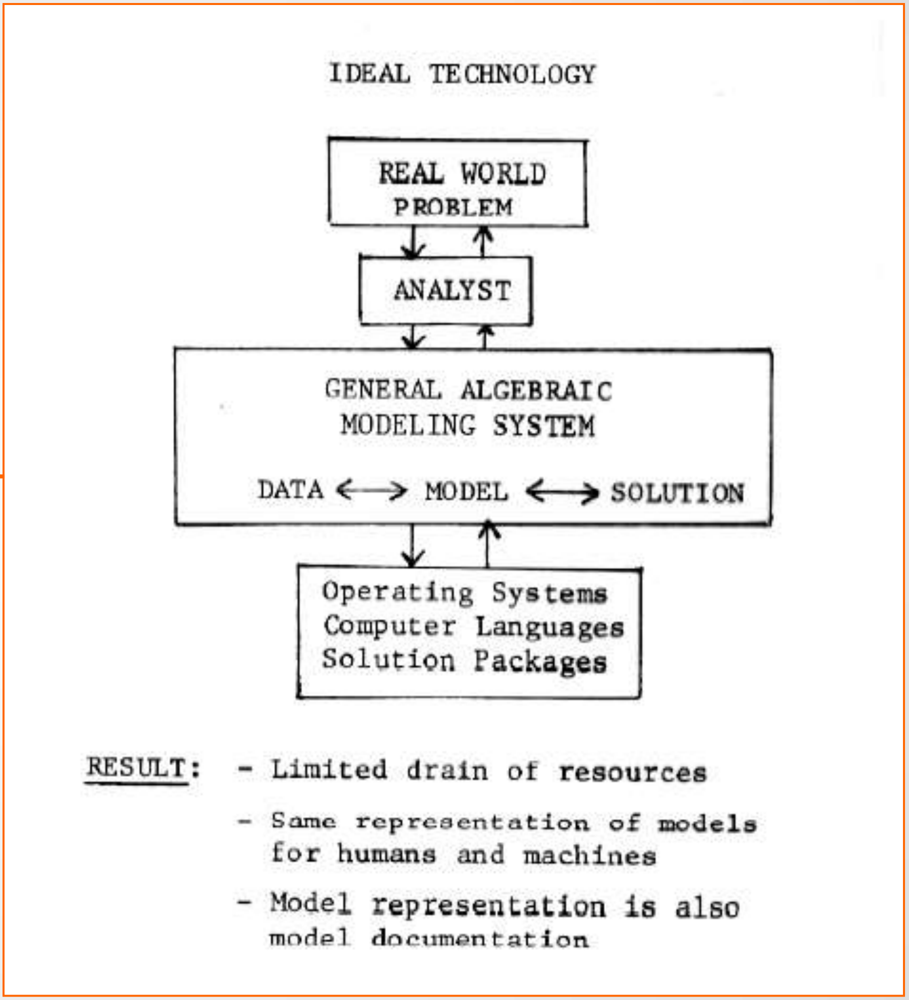
Algebraic Modeling Languages – A Success Story

GAMS – Highlights and Design Principles

Model Deployment

1976 - A World Bank Slide

The Vision



IDEAL TECHNOLOGY

REAL WORLD PROBLEM

ANALYST

GENERAL ALGEBRAIC MODELING SYSTEM

DATA ⟷ MODEL ⟷ SOLUTION

Operating Systems
Computer Languages
Solution Packages

RESULT:
- Limited drain of resources
- Same representation of models for humans and machines
- Model representation is also model documentation

Algebraic Modeling Languages (AML)

1

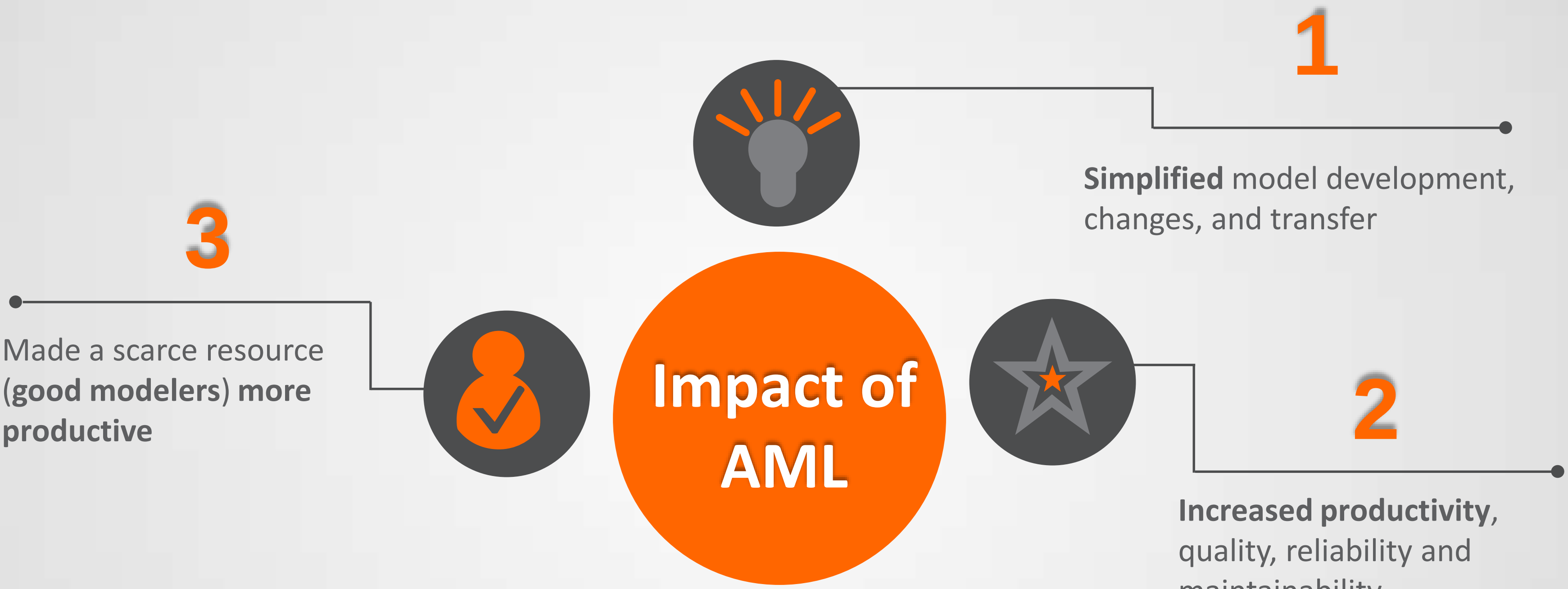
- High-level **computer programming languages**
  - Formulation of **mathematical optimization problems**
  - **Notation similar to algebraic notation**

2

- **Do not solve problems directly**, but offer links to state-of-the-art algorithms (“solver-links”)

Source: http://en.wikipedia.org/wiki/Algebraic_modeling_language

Impact of Algebraic Modeling Languages



Impact of AML

1

Simplified model development, changes, and transfer

2

Increased productivity, quality, reliability and maintainability

3

Made a scarce resource (**good modelers) more productive**

- **Important vehicle to make mathematical optimization available to a broader audience**

2012 INFORMS Impact Prize

36 Years

later

Originators of Algebraic Modeling Languages

AIMMS 3

AMPL

MAXIMAL SOFTWARE

GAMS

LINDO SYSTEMS

Agenda

Algebraic Modeling Languages – A Success Story

GAMS – Highlights and Design Principles

Model Deployment

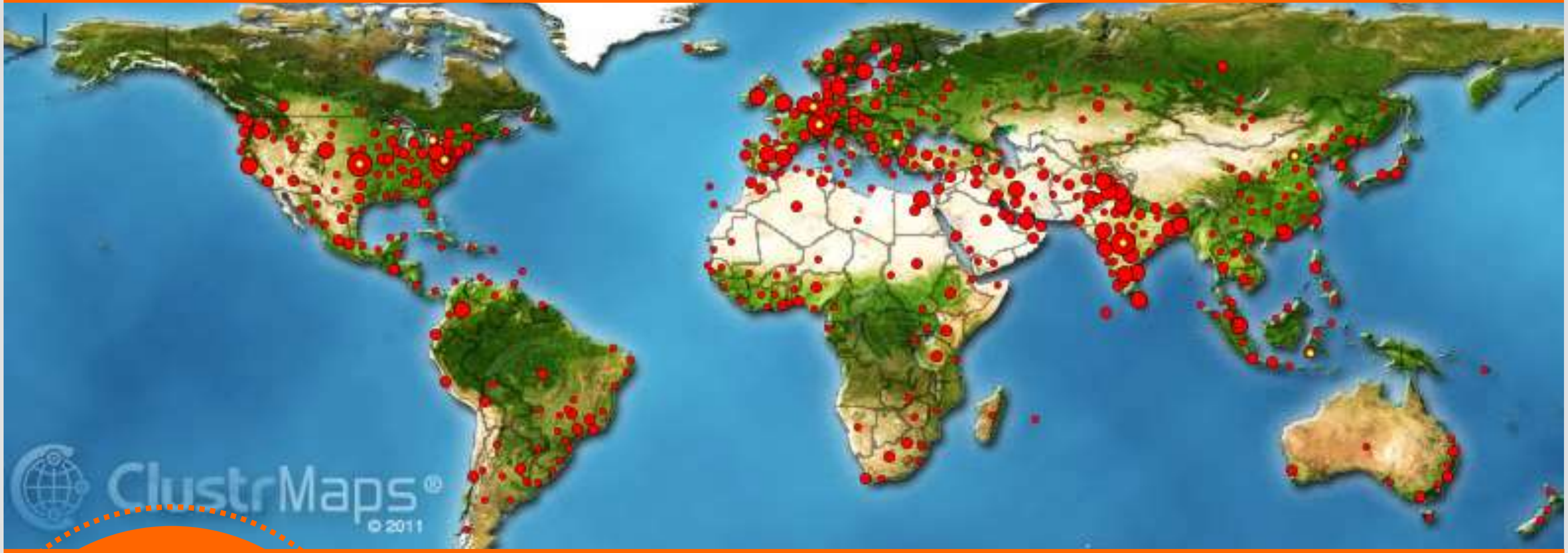
What does he have to think about?

1. Problem
2. Mathematics
3. Programming
4. Performance
5. Scalability
6. Connectivity
7. Deployment
8. Maintenance (Life Cycle)
9. ...

➢ **Why use an AML like GAMS?**

Broad User Community and Network

GAMS used in more than 120 countries

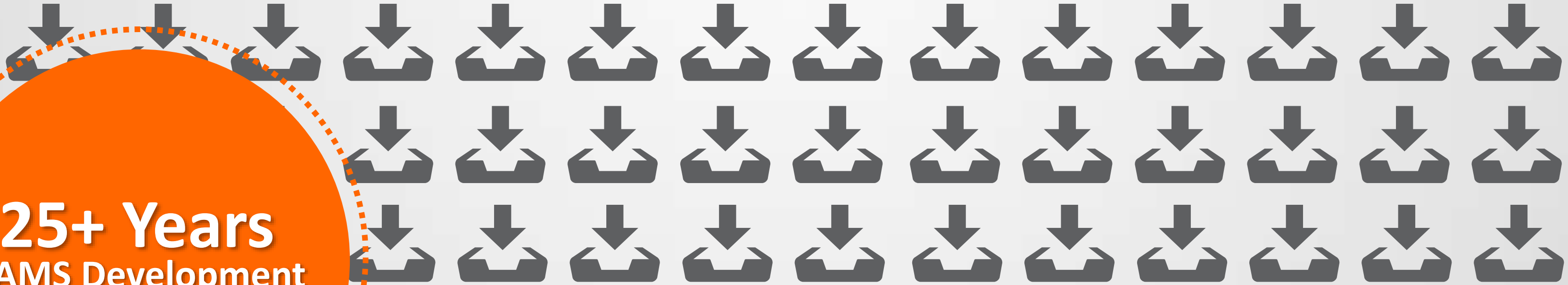


25+ Years
GAMS Development

Broad User Community and Network

More than 10,000 licenses

50% academic users, 50% commercial users



25+ Years
GAMS Development

6,000+ monthly downloads of the free system

Broad Range of Application Areas

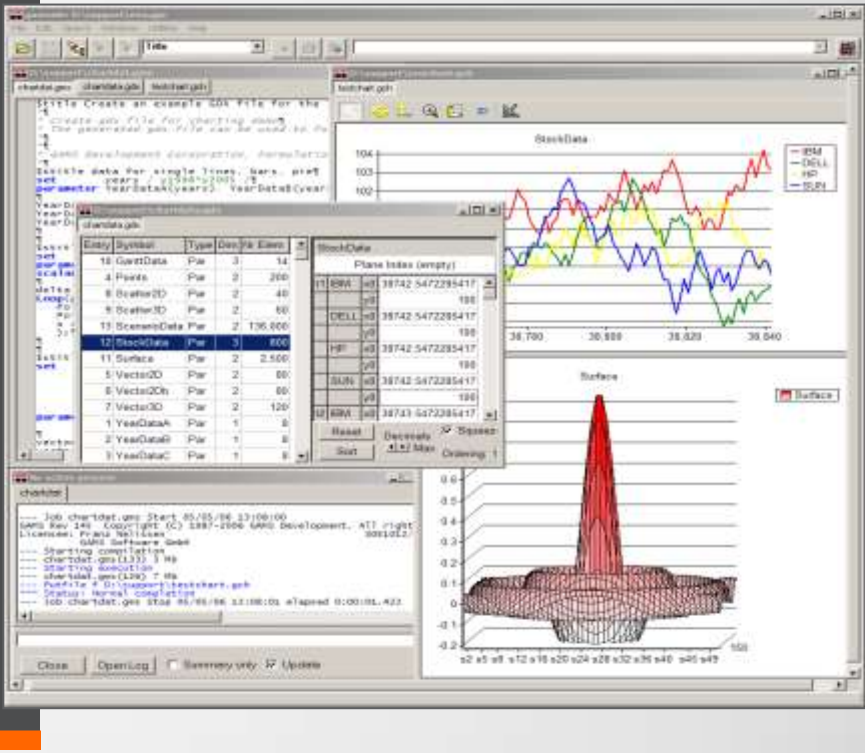
- Agricultural Economics
- Applied General Equilibrium
- Chemical Engineering
- Economic Development
- Econometrics
- Energy
- Environmental Economics
- Engineering
- Finance
- Forestry
- International Trade
- Logistics
- Macro Economics
- Military
- Management Science/OR
- Mathematics
- Micro Economics
- Physics

25+ Years
GAMS Development

Strong Development Environment

GAMS IDE

- Project management
- Editor / Syntax coloring / Spell checks
- Listing file / Tree view / Syntax-error navigation
- Model Debugging / Profiling
- Solver selection / Option selection
- Data viewer
  - Export
  - Charting
- GAMS Process Control
- Model Libraries -1250 Models included



Entry	Symbol	Type	Dim	Nr Elem
10	GanttData	Par	3	14
4	Points	Par	2	200
8	Scatter2D	Par	2	40
9	Scatter3D	Par	2	60
13	ScenarioData	Par	2	136,000
12	StockData	Par	3	800
11	Surface	Par	2	2,500
5	Vector2D	Par	2	80
6	Vector2Dh	Par	2	80
7	Vector3D	Par	2	120
1	YearDataA	Par	1	8
2	YearDataB	Par	1	8
3	YearDataC	Par	1	8

```
--- Job chartdat.gms Start 05/05/06 13:08:00
GAMS Rev 146  Copyright (C) 1987-2006 GAMS Development. All rights
Licensee: Franz Nelissen                                   G051012
          GAMS Software GmbH
--- Starting compilation
--- chartdat.gms(113) 3 Mb
--- Starting execution
--- chartdat.gms(129) 7 Mb
--- Putfile ? D:\support\testchart.gch
--- Status: Normal completion
--- Job chartdat.gms Stop 05/05/06 13:08:01 elapsed 0:00:01.422
```

→ **Everything for rapid model development**

Design Principles

1

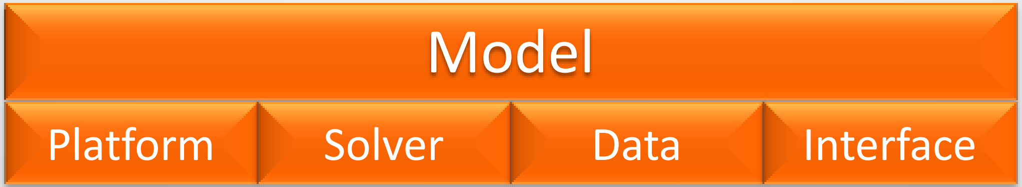
- Simple modeling language with a balanced mix of declarative and procedural elements

2

- Open architecture and interfaces to other systems

3

- Independent layers

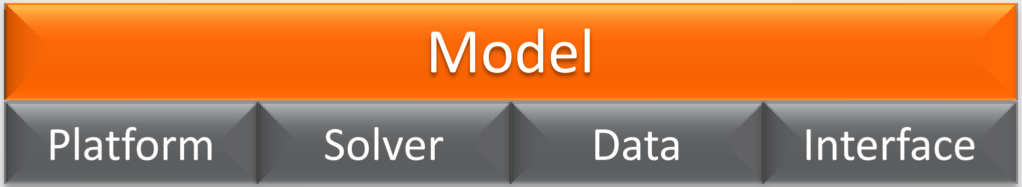


Simple Declarative Language

1
- Language similar to mathematical notation

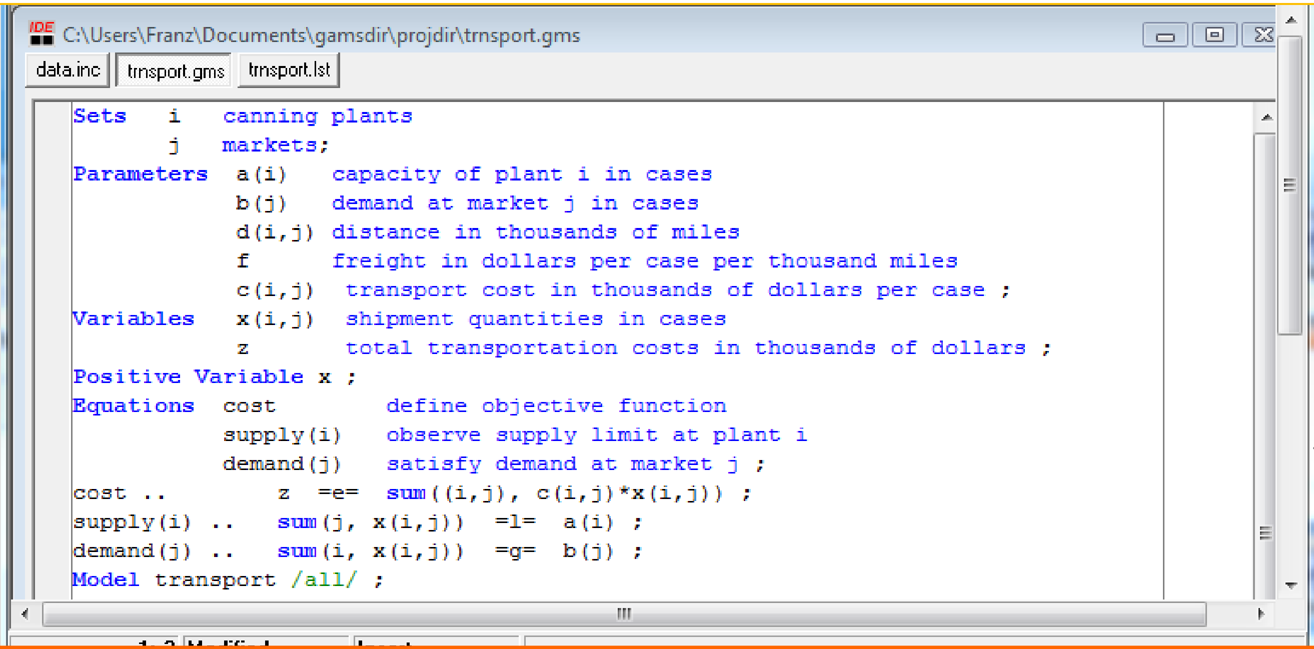
2
- Few basic language elements: sets, parameters, variables, equations, models → Easy to learn

3
- Lot's of code optimization under the hood



Example

$$
\begin{array}{llll}
\text{Objective} & \sum_i \sum_j c_{i,j} \times x_{i,j} & \longrightarrow \min & \\
\text{Observe supply limit at plant } i\text{:} & \sum_j x_{i,j} & \leq a_i & \forall i \\
\text{Satisfy demand at market } j\text{:} & \sum_i x_{i,j} & \geq b_j & \forall j \\
 & x_{i,j} & \geq 0 & \forall i,j
\end{array}
$$



```
Sets   i   canning plants
       j   markets;
Parameters  a(i)   capacity of plant i in cases
            b(j)   demand at market j in cases
            d(i,j) distance in thousands of miles
            f      freight in dollars per case per thousand miles
            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 transport /all/ ;
```

- ➢ **Model is executable description of the problem**

Mix of Declarative and Procedural Elements

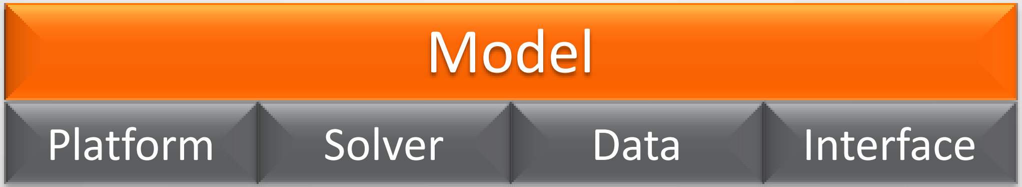
Procedural elements like loops, for, if, macros and functions

Allow to build complex problem algorithms within GAMS

Interaction with other systems:

- Job control
- Data exchange

➢ Combine models inside the language

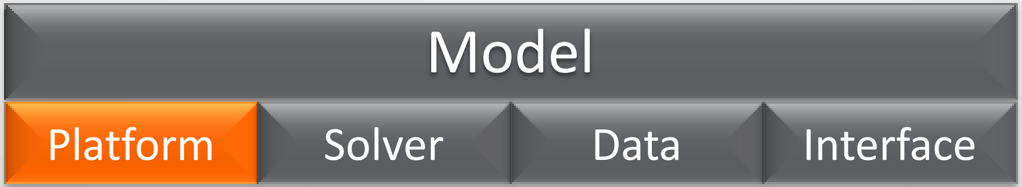


Independence of Model and Operating System

Platforms supported by GAMS:

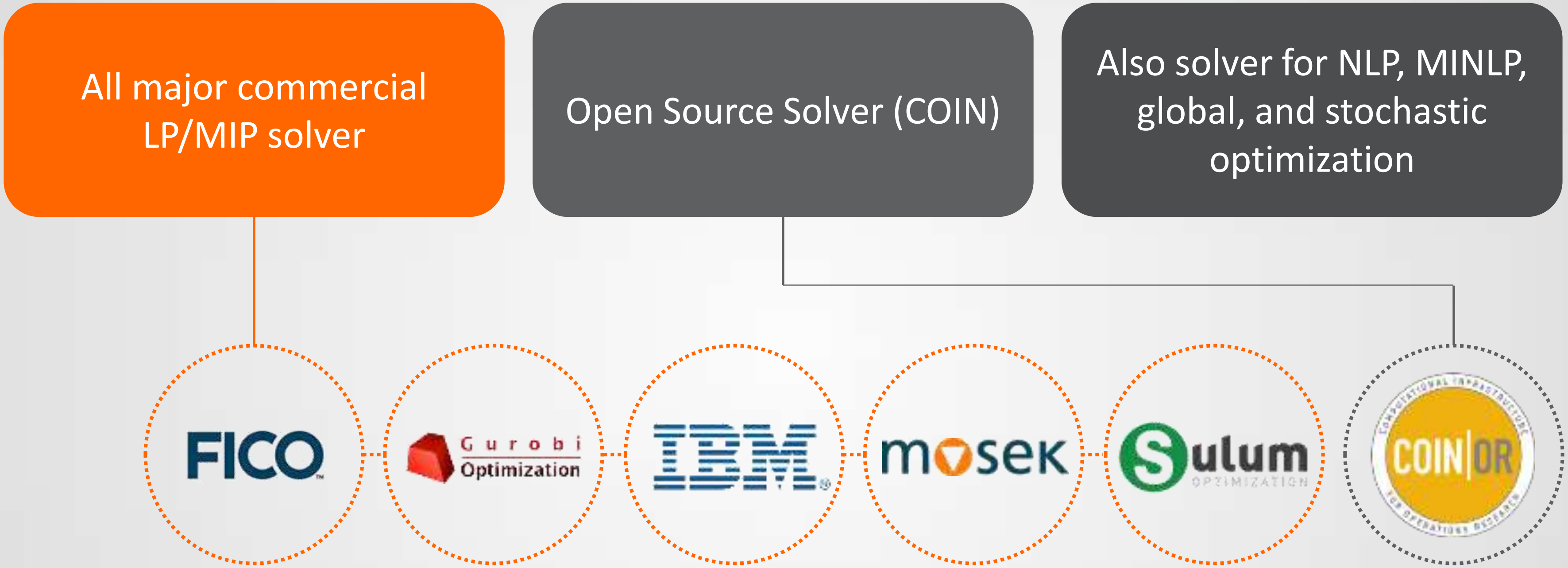
Microsoft Windows
Linux
X
Welcome to Mac OS X
solaris
AIX

- Models can be moved between platforms with ease!



Independence of Model and Solver

One environment for a wide range of model types and solvers



All major commercial LP/MIP solver

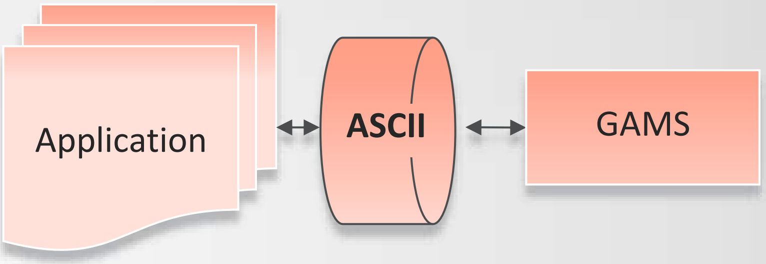
Open Source Solver (COIN)

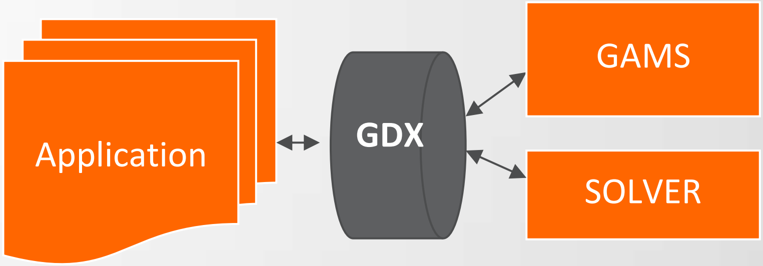
Also solver for NLP, MINLP, global, and stochastic optimization

- ➢ **Switching between solvers with one line of code!**

Independence of Model and Data

- **Declarative Modeling**: *x(j), j∈{1,…}*
  - ASCII: Initial model development
  - GDX: Binary Data layer ("contract") between GAMS and applications
    - Platform independent
    - Direct GDX interfaces and general API





Independence of Model and User Interface

1

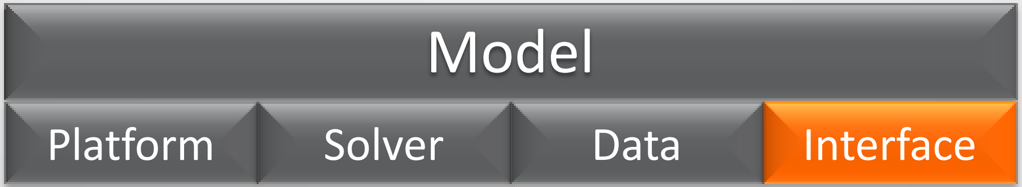
- **Open architecture and interfaces to other systems → No preference for a particular user interface**

2

- Application Programming Interfaces
  - *Low Level*
  - **Object Oriented**: .Net, Java, Python, …

3

- **Smart Links to popular environments**
  - **Excel, MATLAB, R, …**


Model

Platform

Solver

Data

Interface

Agenda

Algebraic Modeling Languages – A Success Story

GAMS – Highlights and Design Principles

Model Deployment

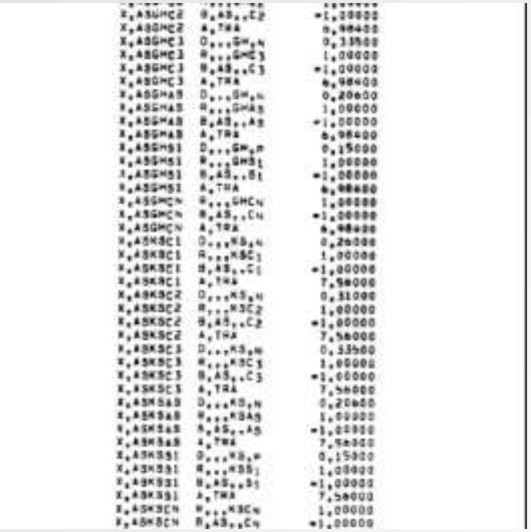
Is Optimization special?

Observation:

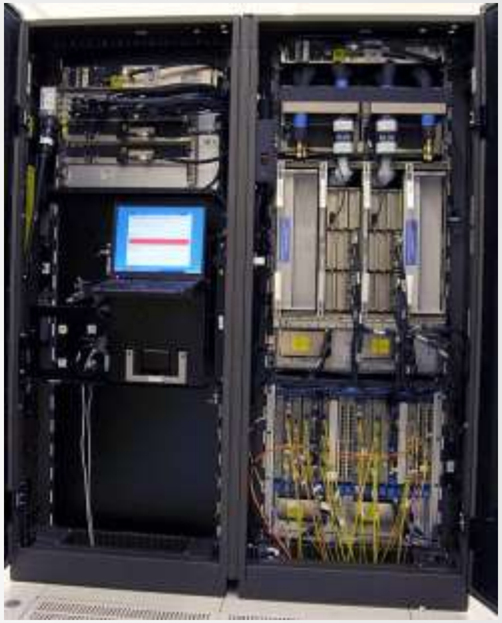
Optimization models
- are expensive to develop
- may have long a lifespan

Modeling Systems & Applications have to be adjusted
- New computer paradigms
- New solver technology and solution methods
- New graphical user interfaces and deployment environments

Change in Focus: Past



```
X,ASGMC2  B,AS..C2        -1,00000
X,ASGMC2  A,TRA            6,98400
X,ASGMC3  D...GM,N         0,33500
X,ASGMC3  R...GMC3         1,00000
X,ASGMC3  B,AS..C3        -1,00000
X,ASGMC3  A,TRA            6,98400
X,ASGMAS  D...GM,N         0,20600
X,ASGMAS  R...GMAS         1,00000
X,ASGMAS  B,AS..AS        -1,00000
X,ASGMAS  A,TRA            6,98400
X,ASGMS1  D...GM,M         0,15000
X,ASGMS1  R...GMS1         1,00000
X,ASGMS1  B,AS..S1        -1,00000
X,ASGMS1  A,TRA            6,98400
X,ASGMCN  R...GMCN         1,00000
X,ASGMCN  B,AS..CN        -1,00000
X,ASGMCN  A,TRA            6,98400
X,ASKSC1  D...KS,N         0,26000
X,ASKSC1  R...KSC1         1,00000
X,ASKSC1  B,AS..C1        -1,00000
X,ASKSC1  A,TRA            7,50000
X,ASKSC2  D...KS,N         0,31000
X,ASKSC2  R...KSC2         1,00000
X,ASKSC2  B,AS..C2        -1,00000
X,ASKSC2  A,TRA            7,56000
X,ASKSC3  D...KS,N         0,33500
X,ASKSC3  R...KSC3         1,00000
X,ASKSC3  B,AS..C3        -1,00000
X,ASKSC3  A,TRA            7,56000
X,ASKSAS  D...KS,N         0,20600
X,ASKSAS  R...KSAS         1,00000
X,ASKSAS  B,AS..AS        -1,00000
X,ASKSAS  A,TRA            7,56000
X,ASKSS1  D...KS,M         0,15000
X,ASKSS1  R...KSS1         1,00000
X,ASKSS1  B,AS..S1        -1,00000
X,ASKSS1  A,TRA            7,56000
X,ASKSCN  R...KSCN         1,00000
X,ASKSCN  B,AS..CN        -1,00000
```



Computation

Users:

→ Left out

Model

Users:

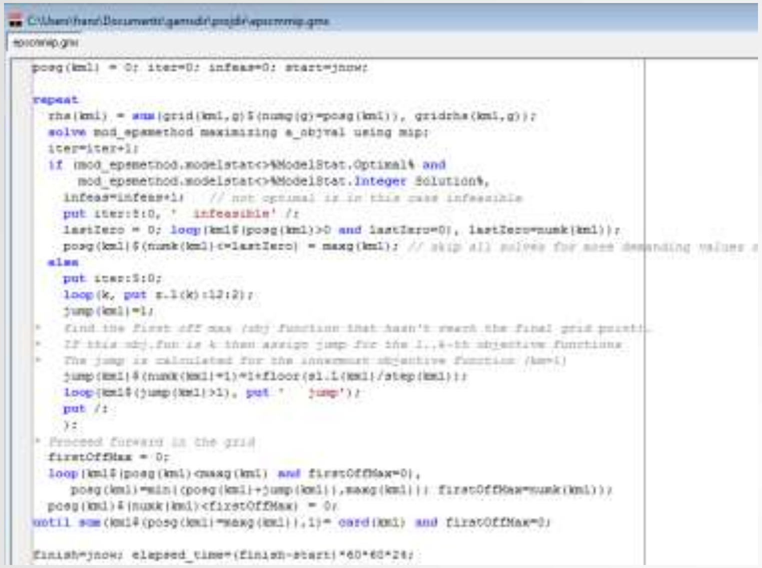
→ Involved

Application

Users:

→Not aware of model

Change in Focus: Now



```
posg(km1) = 0; iter=0; infeas=0; start=jnow;

repeat
  rhs(km1) = sum(grid(km1,g)$(numg(g)=posg(km1)), gridrhs(km1,g));
  solve mod_epsmethod maximizing a_objval using mip;
  iter=iter+1;
  if (mod_epsmethod.modelstat<>%ModelStat.Optimal% and
      mod_epsmethod.modelstat<>%ModelStat.Integer Solution%,
    infeas=infeas+1;   // not optimal is in this case infeasible
    put iter:5:0, '  infeasible' /;
    lastZero = 0; loop(km1$(posg(km1)>0 and lastZero=0), lastZero=numk(km1));
    posg(km1)$(numk(km1)<=lastZero) = maxg(km1); // skip all solves for more demanding values o
  else
    put iter:5:0;
    loop(k, put z.l(k):12:2);
    jump(km1)=1;
*   find the first off max (obj function that hasn't reach the final grid point).
*   If this obj.fun is k then assign jump for the 1..k-th objective functions
*   The jump is calculated for the innermost objective function (km=1)
    jump(km1)$(numk(km1)=1)=1+floor(sl.L(km1)/step(km1));
    loop(km1$(jump(km1)>1), put '   jump');
    put /;
    );
* Proceed forward in the grid
  firstOffMax = 0;
  loop(km1$(posg(km1)<maxg(km1) and firstOffMax=0),
     posg(km1)=min((posg(km1)+jump(km1)),maxg(km1)); firstOffMax=numk(km1));
  posg(km1)$(numk(km1)<firstOffMax) = 0;
until sum(km1$(posg(km1)=maxg(km1)),1)= card(km1) and firstOffMax=0;

finish=jnow; elapsed_time=(finish-start)*60*60*24;
```

Computation

Users:

→ Left out

Model

Users:

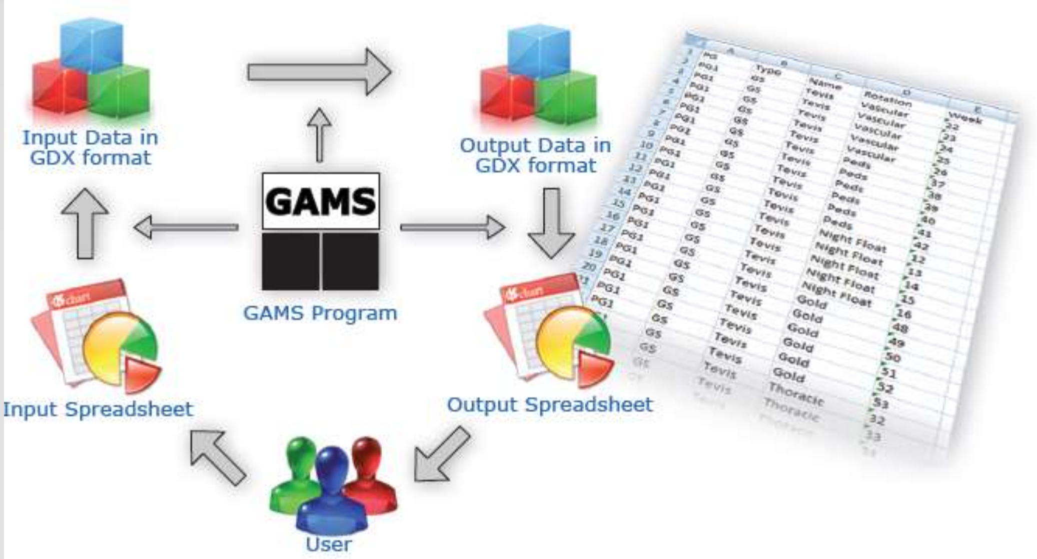
→ Involved

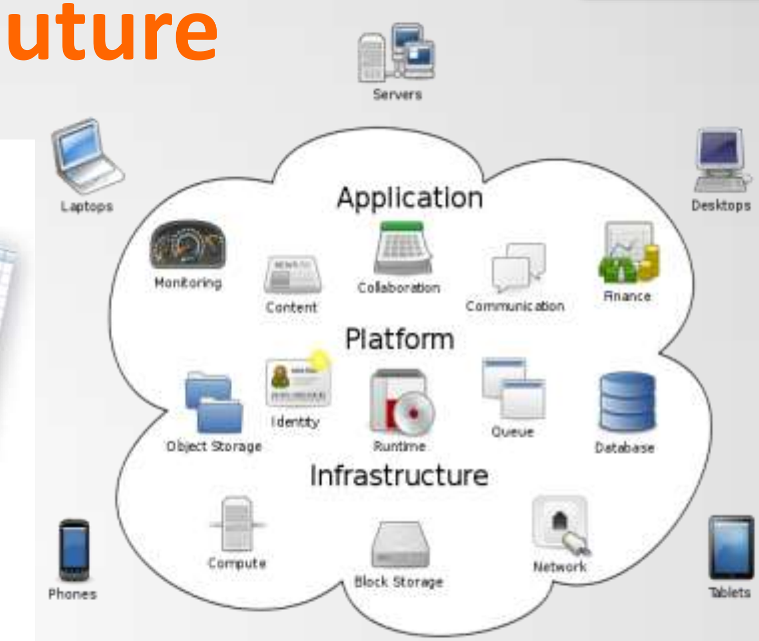
Application

Users:

→Not aware of model

Change in Focus: Now / Future





Cloud Computing

Computation

Users:

→ Left out

Model

Users:

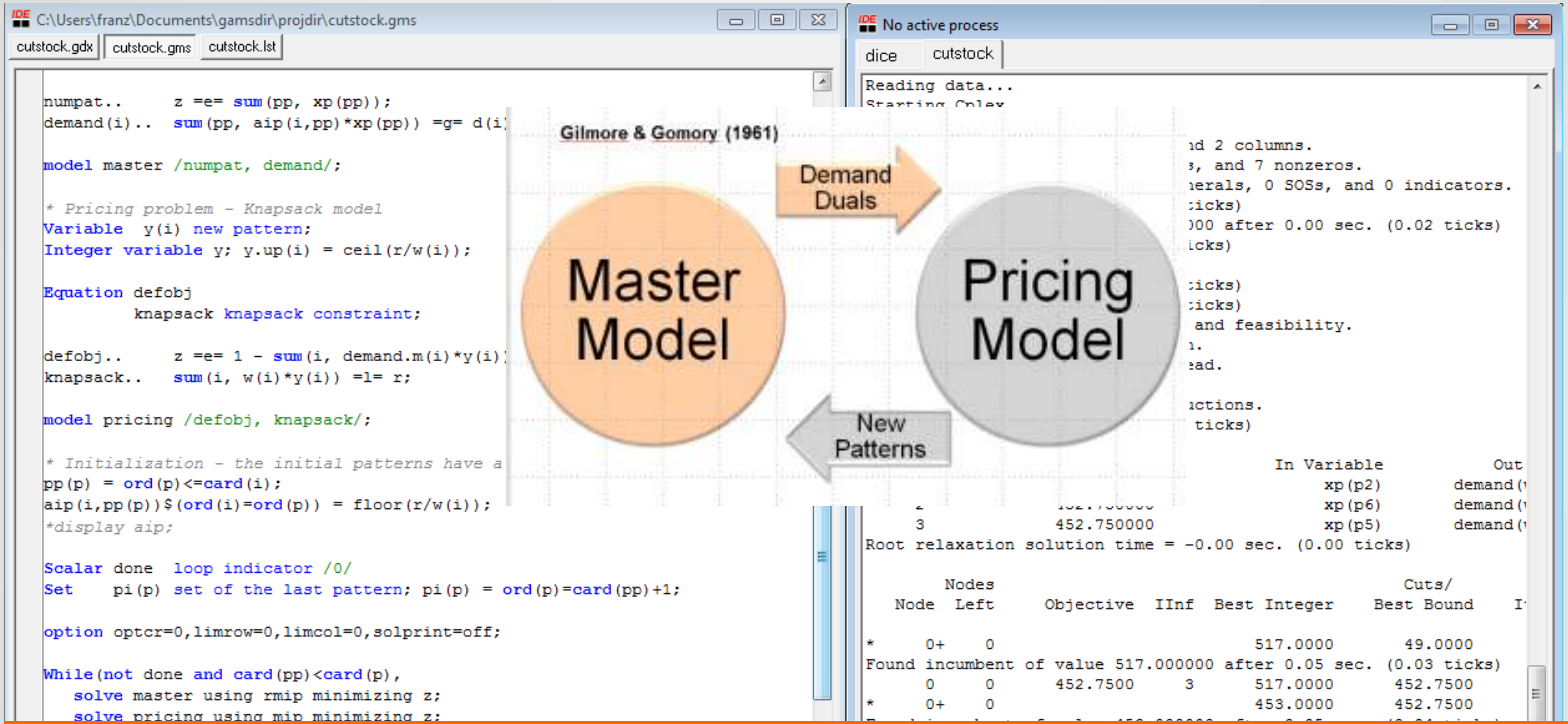
→ Involved

Application

Users:

→Not aware of model

Change in Focus: Modeler…

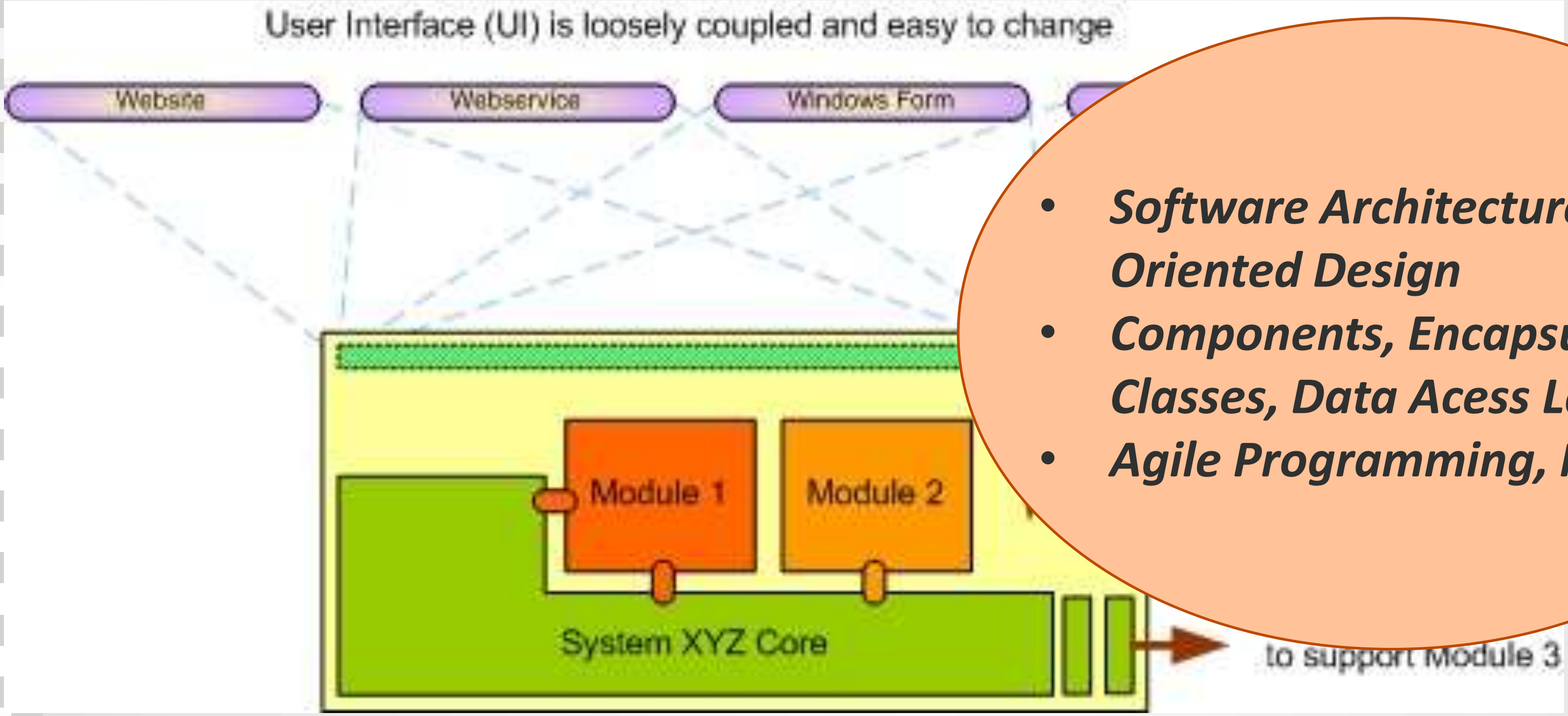


```
numpat..      z =e= sum(pp, xp(pp));
demand(i)..   sum(pp, aip(i,pp)*xp(pp)) =g= d(i

model master /numpat, demand/;

* Pricing problem - Knapsack model
Variable  y(i) new pattern;
Integer variable y; y.up(i) = ceil(r/w(i));

Equation defobj
         knapsack knapsack constraint;

defobj..      z =e= 1 - sum(i, demand.m(i)*y(i)
knapsack..    sum(i, w(i)*y(i)) =l= r;

model pricing /defobj, knapsack/;

* Initialization - the initial patterns have a
pp(p)  = ord(p)<=card(i);
aip(i,pp(p))$(ord(i)=ord(p))  = floor(r/w(i));
*display aip;

Scalar done  loop indicator /0/
Set    pi(p) set of the last pattern; pi(p) = ord(p)=card(pp)+1;

option optcr=0,limrow=0,limcol=0,solprint=off;

While(not done and card(pp)<card(p),
   solve master using rmip minimizing z;
   solve pricing using mip minimizing z;
```

```
Reading data...
Starting Cplex
                                     nd 2 columns.
                                     s, and 7 nonzeros.
                                     nerals, 0 SOSs, and 0 indicators.
                                     ticks)
                                     000 after 0.00 sec. (0.02 ticks)
                                     icks)

                                     ticks)
                                     ticks)
                                      and feasibility.
                                     n.
                                     ead.

                                     actions.
                                      ticks)

                                              In Variable          Out
                                                   xp(p2)        demand(
                                                   xp(p6)        demand(
      3              452.750000                    xp(p5)        demand(
Root relaxation solution time = -0.00 sec. (0.00 ticks)

        Nodes                                                   Cuts/
   Node  Left     Objective  IInf  Best Integer     Best Bound    I

*     0+    0                          517.0000        49.0000
Found incumbent of value 517.000000 after 0.05 sec. (0.03 ticks)
      0     0      452.7500     3      517.0000       452.7500
*     0+    0                          453.0000       452.7500
```

- Small Community: 2010 ~ 64,000 OR Analytic Professionals in the US

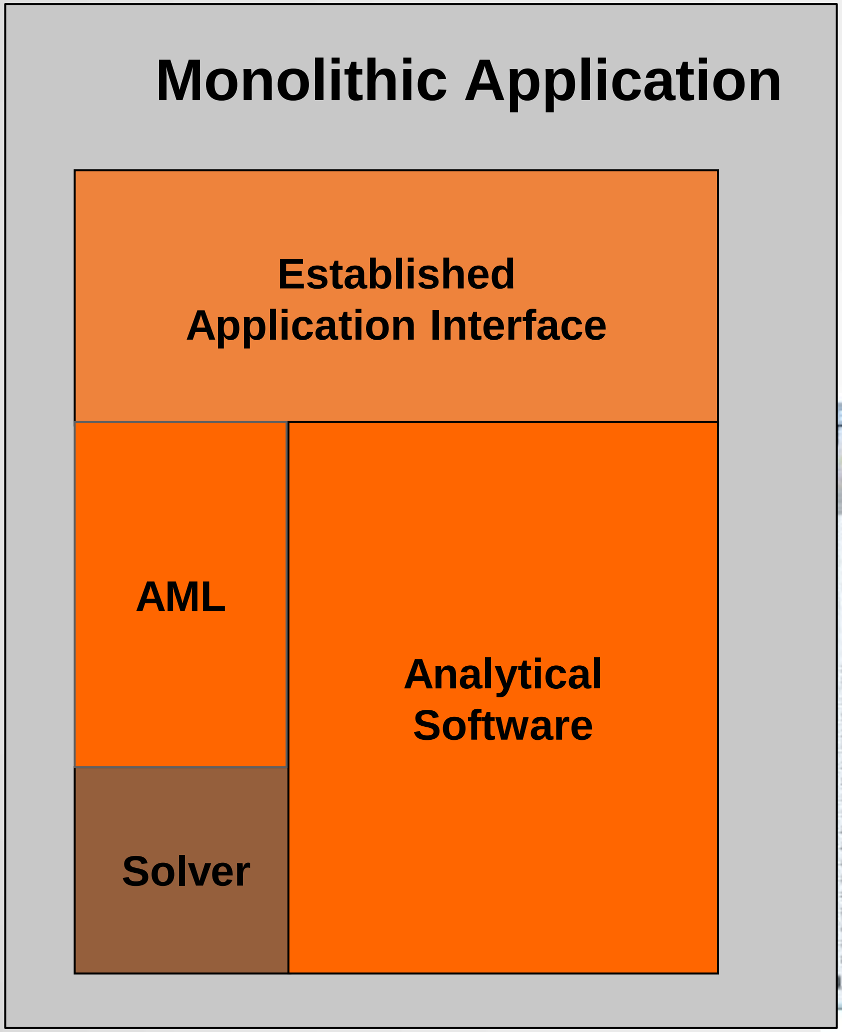
Change in Focus: Application Developer



- *Software Architecture, Object Oriented Design*
- *Components, Encapsulation, Classes, Data Acess Layer,..*
- *Agile Programming, Mesh, …*

- **Huge Community: 2006 ~ 3.3 Mill. IT Professionals in the US (2006)**
- **Rapidly changing IT environments**

Example – All in One – Top Down



Monolithic Application

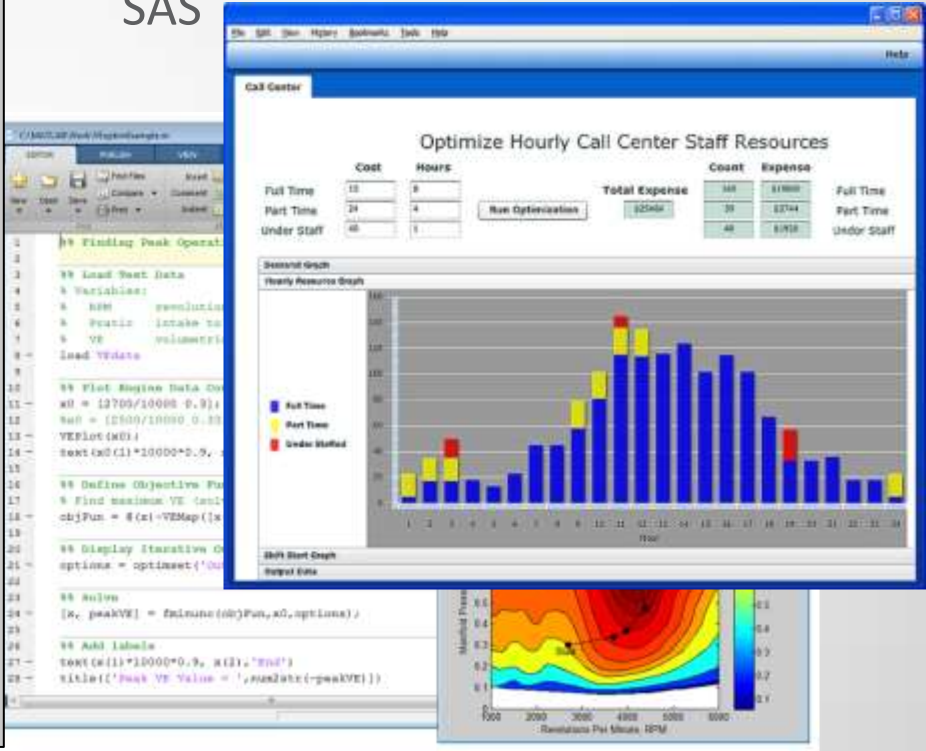
Established Application Interface

AML

Analytical Software

Solver

- Add “AML” to existing analytical software system
- “large” user base, e.g. MATLAB, or SAS



```
%% Finding Peak Operat
%% Load Test Data
% Variables:
%   RPM      revolution
%   Pratio   intake to
%   VE       volumetric
load VEdata

%% Plot Engine Data Co
x0 = [2700/10000 0.3];
%x0 = [2500/10000 0.33
VEPlot(x0);
text(x0(1)*10000*0.9,

%% Define Objective Fu
% Find maximum VE (onl
objFun = @(x)-VEMap([x

%% Display Iterative O
options = optimset('Di

%% Solve
[x, peakVE] = fminunc(objFun,x0,options);

%% Add labels
text(x(1)*10000*0.9, x(2),'End')
title(['Peak VE Value = ',num2str(-peakVE)])
```

Call Center

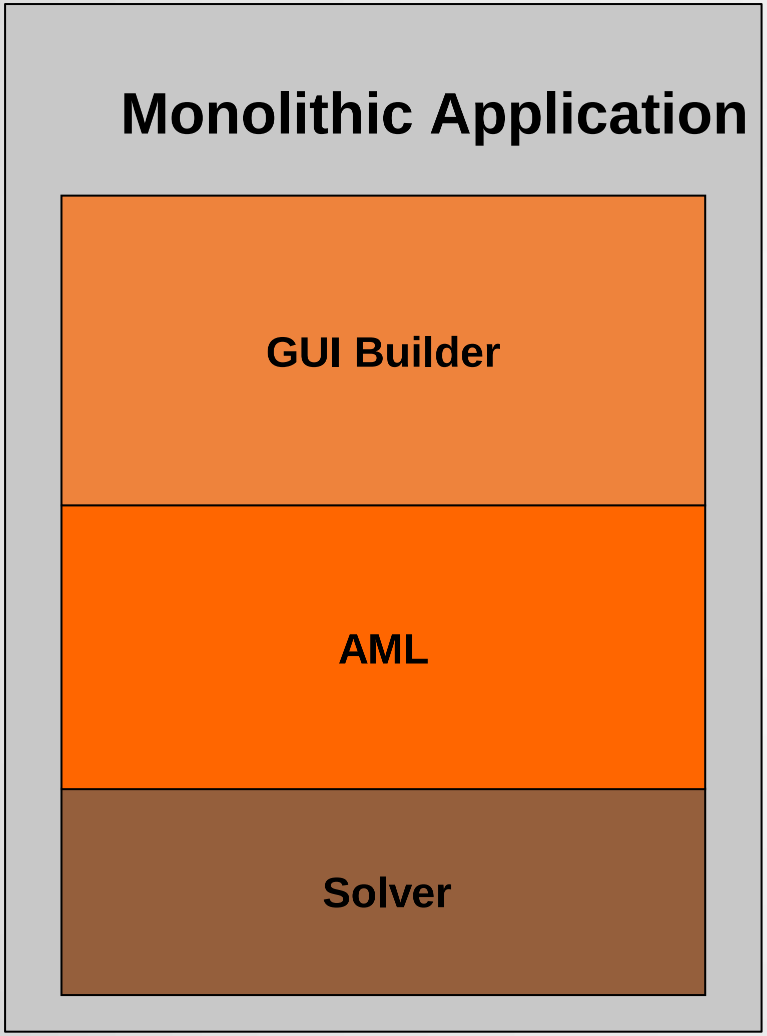
Optimize Hourly Call Center Staff Resources

	Cost	Hours
Full Time	15	8
Part Time	24	4
Under Staff	40	1

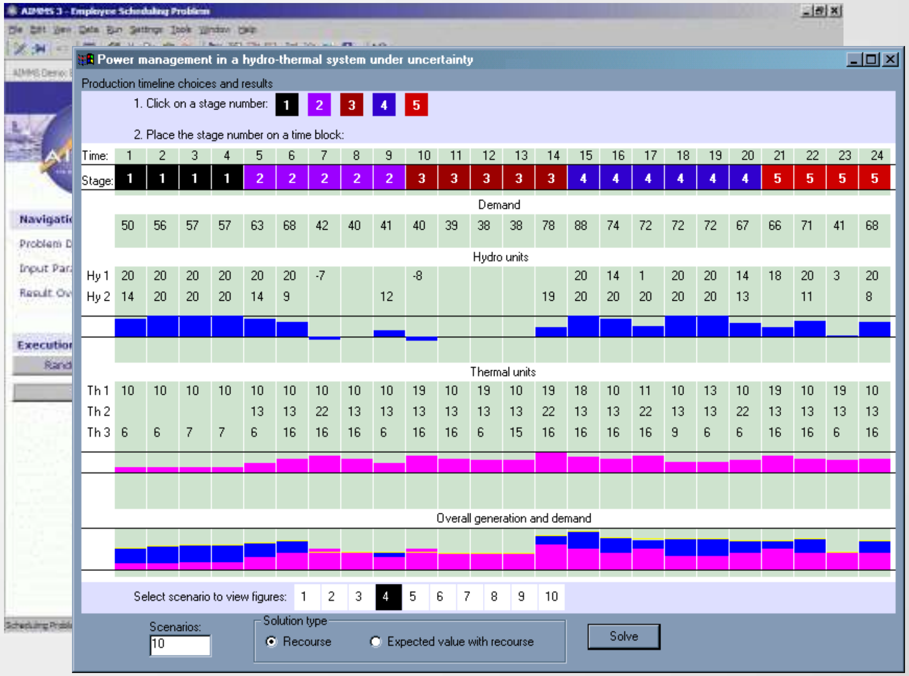
Total Expense

Count	Expense	
168	[illegible]	Full Time
39	[illegible]	Part Time
48	[illegible]	Under Staff

Example – All in One – Bottom Up



- Integrate GUI-builder into AML
- “small” user base, e.g. AIMMS (Pro) or FICO Xpress-Insight


AIMMS 3 - Employee Scheduling Problem
File Edit View Data Run Settings Tools Window Help
Navigation
Execution

Power management in a hydro-thermal system under uncertainty

Production timeline choices and results

1. Click on a stage number: 1 2 3 4 5

2. Place the stage number on a time block:

Time:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Stage:	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	4	5	5	5	5
Demand																								
	50	56	57	57	63	68	42	40	41	40	39	38	38	78	88	74	72	72	72	67	66	71	41	68
Hydro units																								
Hy 1	20	20	20	20	20	20	-7			-8					20	14	1	20	20	14	18	20	3	20
Hy 2	14	20	20	20	14	9			12					19	20	20	20	20	20	13		11		8
Thermal units																								
Th 1	10	10	10	10	10	10	10	10	10	19	10	19	10	19	18	10	11	10	13	10	19	10	19	10
Th 2					13	13	22	13	13	13	13	13	13	22	13	13	22	13	13	22	13	13	13	13
Th 3	6	6	7	7	6	16	16	16	6	16	16	6	15	16	16	16	16	9	6	6	16	16	6	16

Overall generation and demand

Select scenario to view figures: 1 2 3 4 5 6 7 8 9 10

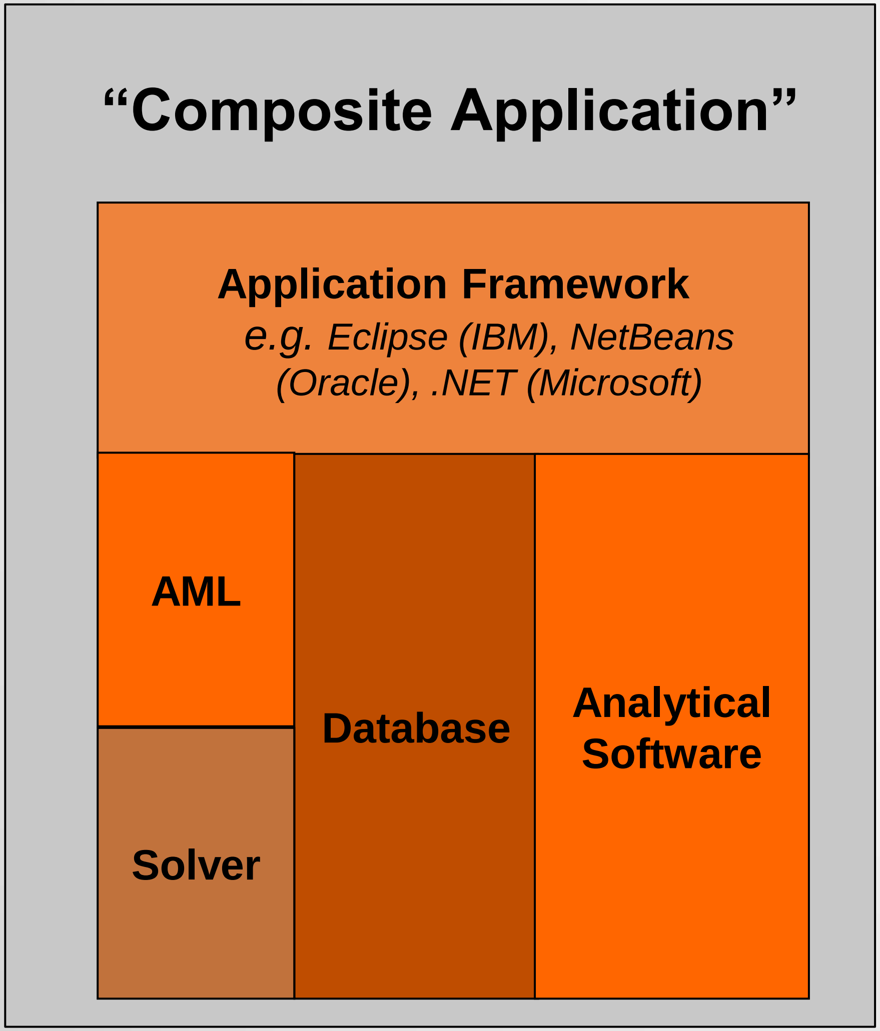
Scenarios: 10

Solution type

● Recourse ○ Expected value with recourse

Solve

Example – Composite Application



"Composite Application"

Application Framework
e.g. Eclipse (IBM), NetBeans (Oracle), .NET (Microsoft)

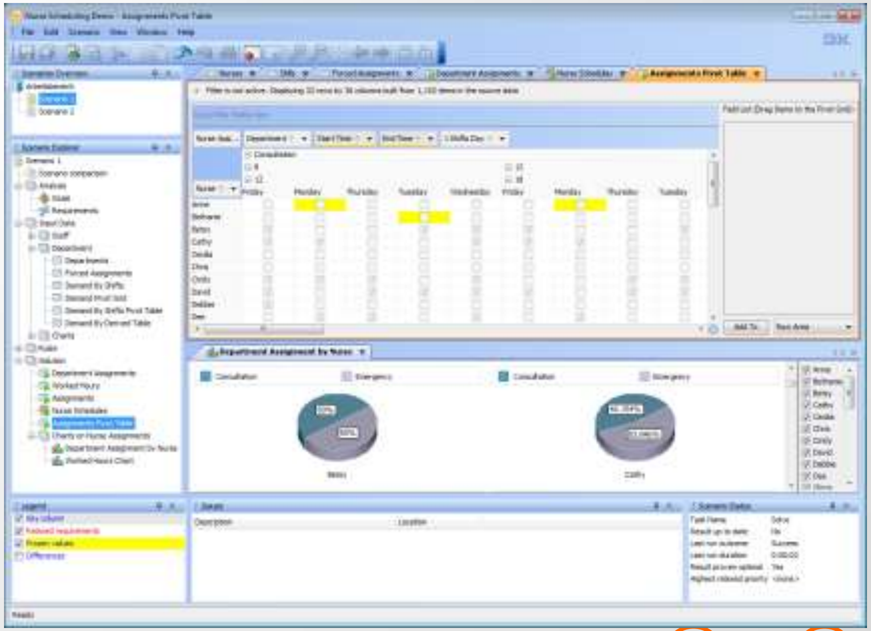
AML

Solver

Database

Analytical Software

- “Construction Kit” with different connected elements
- Use (open source) existing framework to build applications, e.g. IBM ODME



Summary

AML – A Success Story

Design Principles

- Simple, but powerful language
- Open interfaces
- Different layers

Model Deployment

- Is optimization special?
- Provide cutting edge technology
- Don’t lock developers and users into a certain environment

Thank You

USA

GAMS Development Corp. 1217 Potomac
Street, NW Washington, DC 20007
USA
Phone: +1 202 342 0180
Fax: +1 202 342 0181
sales@gams.com

Europe

GAMS Software GmbH
P.O. Box 40 59
50216 Frechen, Germany
Phone: +49 221 949 9170
Fax: +49 221 949 9171
info@gams.de