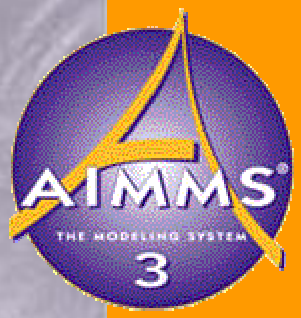


AIMMS

An All-Round Development Environment

Jan Bisschop

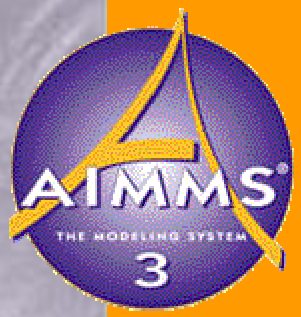
Paragon Decision Technology B.V.
Haarlem, The Netherlands

Presentations and Demos

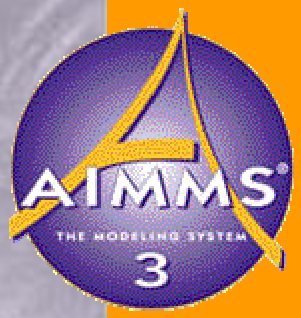
Outline :

- Paragon and AIMMS
- AIMMS Outer Approximation Approach
- Quantities and Units
- Database Interaction
- Multi-Language Support
- Excel Interface
- Multi-Agent Technology

Paragon Decision Technology B.V.



- Dutch company, founded in 1989
- Activities
  - AIMMS development
  - Modeling support and consulting
- AIMMS major releases
  - Version 2.x : 1993 - 1998
  - Version 3.x : 1999 - 2006
  - Version 4.x : 2007 -
- Web site: **www.aimms.com**



AIMMS Acronym

Advanced

Integrated

Multi-Dimensional

Modeling

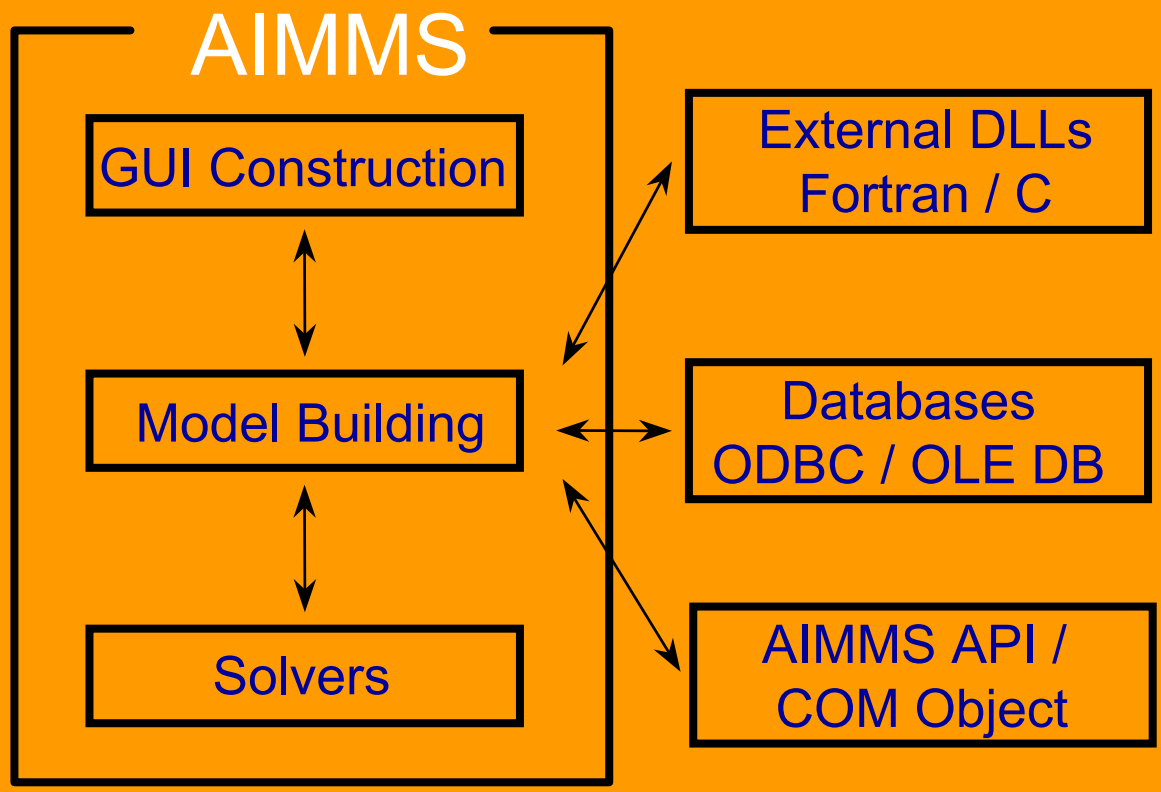
Software

AIMMS Overview

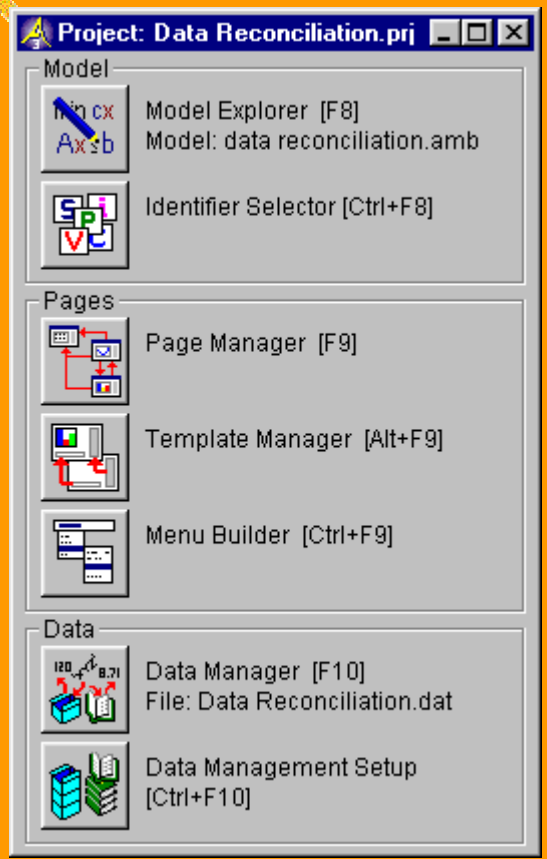
Point & Click

Modeling Language / Model Explorer

Cplex, Conopt, Xpress, etc.

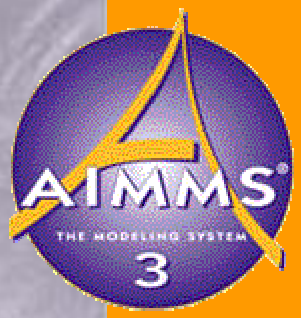


AIMMS Productivity Tools



- Model Explorer
  - tree-based model construction
- Identifier Selector
  - customizable model overviews
- Page Manager
  - navigational interface structure
- Template Manager
  - page and template construction
- Menu Builder
  - customizable menus
- Data Manager
  - case management
- Data Management Setup
  - advanced case management

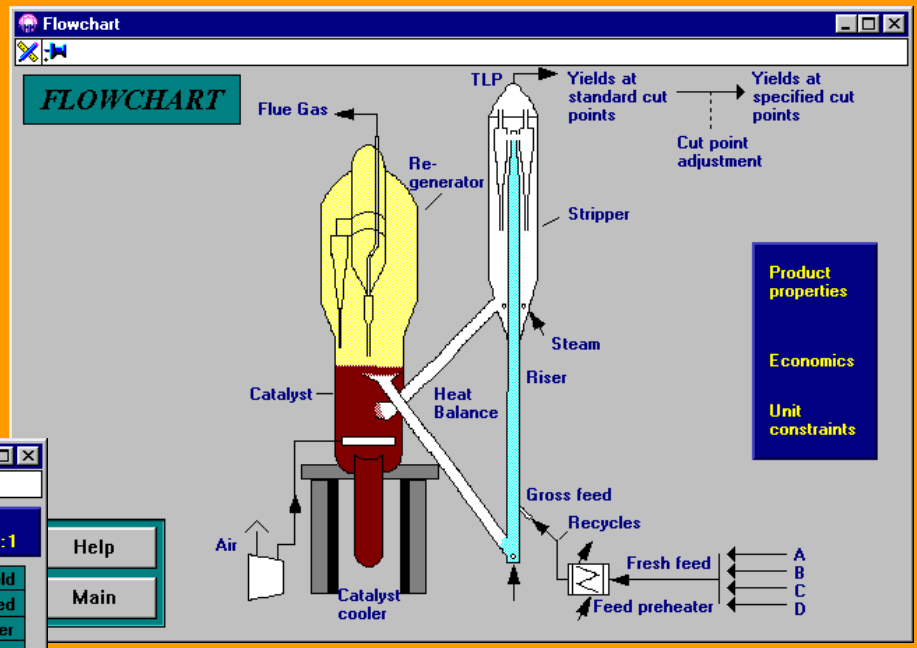
AIMMS-Based DSS Applications

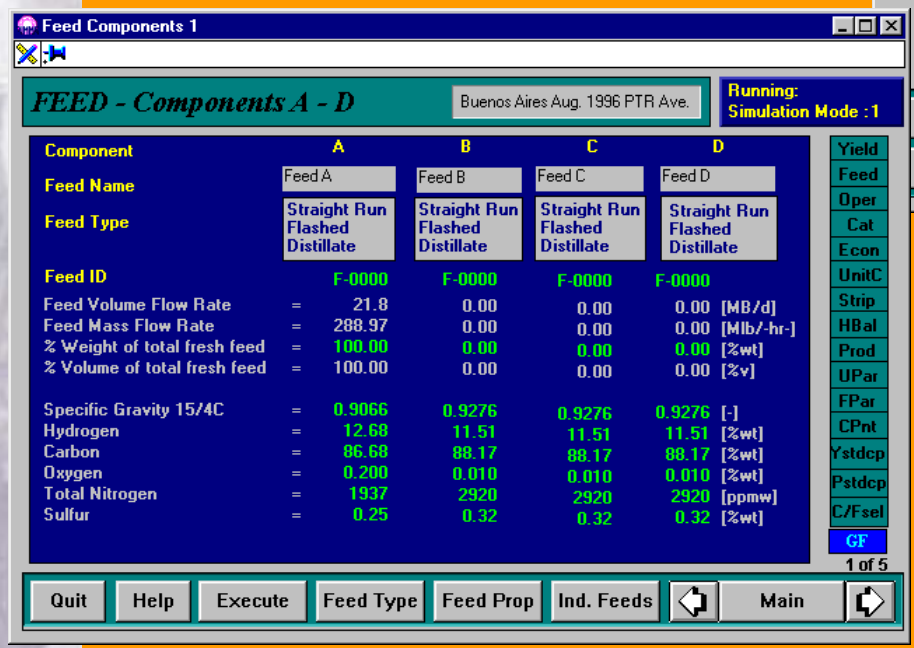


- Clearing Energy Markets (LP, ALSTOM ESCA)
- Cat Cracking Model (NLP, Shell Oil)
- Advanced Blending Module (LP+MIP+NLP, Shell)
- Data Reconciliation (NLP, Shell Chemicals)
- Crude Oil Scheduling (Discrete Event, Shell)
- Refinery Scheduling (NLP, Amoco)
- Forestry Management System (LP, Ontario)
- Structured Asset Management (LP+NLP, ABN-AMRO)
- Scheduling Beer Production (Heuristic, Heineken)
- Simulation Chemical Plant (Simulation, EC Germany)
- Cargo Revenue Management (Lufthansa Cargo)
- Liquidation Planning System (Brown-Forman)
- Management Chain Game (CentER Appl. Research)

AIMMS-based DSS Applications

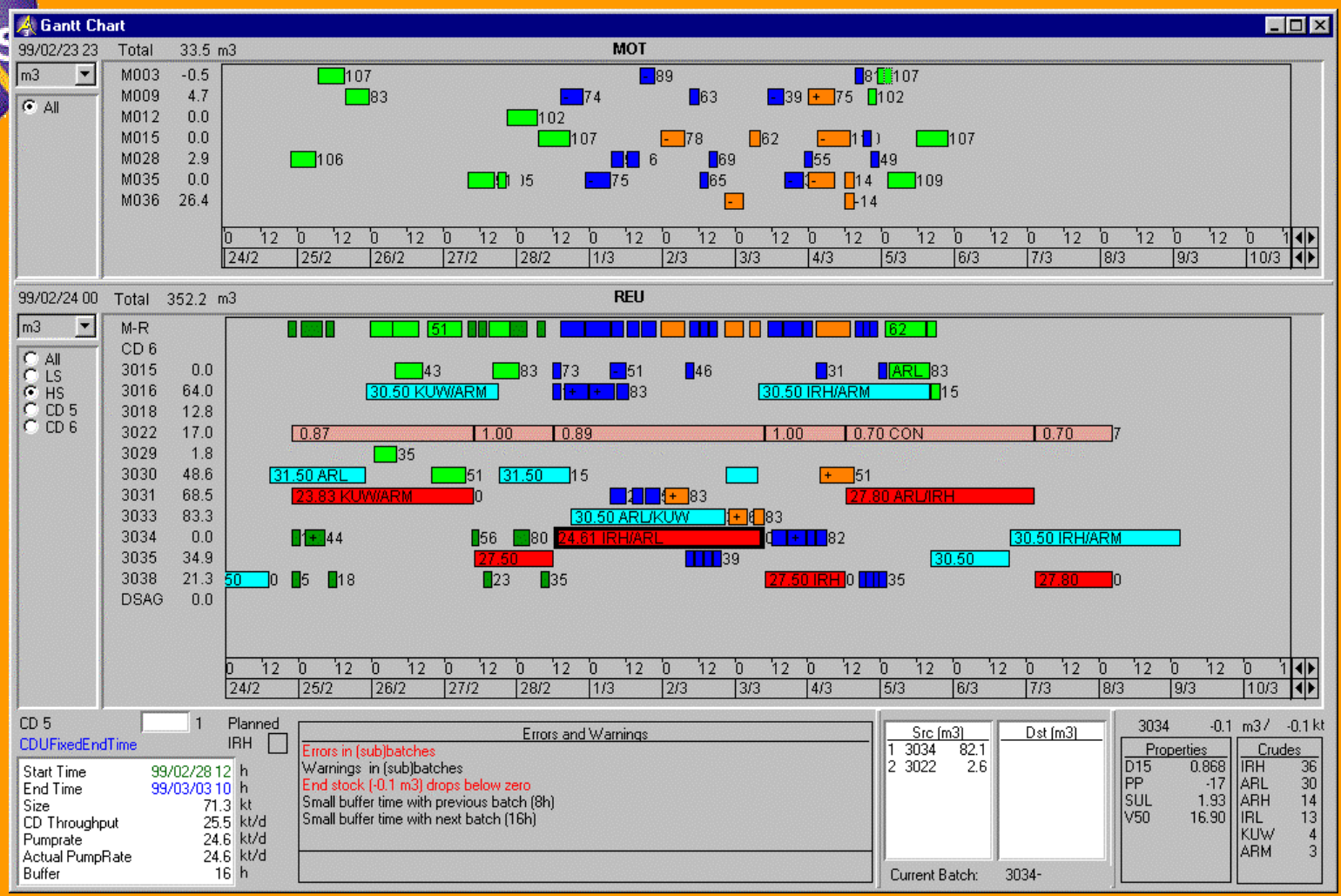
Interface Example (Cat Cracker Model)





AIMMS-based DSS Applications

Interface Example (Crude Oil Scheduling)



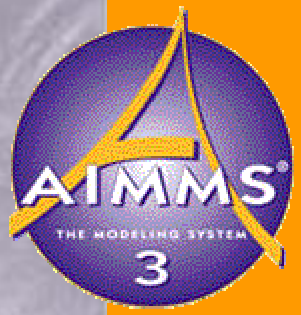
Standard Features

- (Finite) Index Sets
  - Compound
  - Indexed
  - Integer
- Parameters
  - Numerical-valued
  - Element-valued
  - String-valued
  - Unit-valued
- Variables and Constraints
- (Multiple) Mathematical Programs
  - Linear and Mixed-Integer Linear
  - Nonlinear and Mixed-Integer Nonlinear
  - Mixed Complimentarity

Standard Features Continued

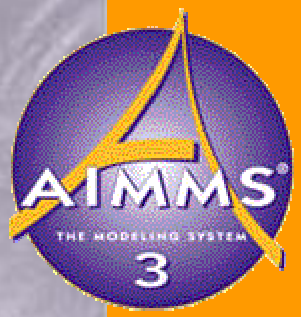
- Procedures and Functions
  - Intrinsic
  - Internal
  - External
- Flow Control
  - If-then-else
  - For / While / Repeat
  - Switch
  - Halt
- Optimization
  - Solve
- Data Communication
  - Read / Write

Key Features



- Sparse Execution and Data Storage
- Procedures and Definitions Combined
- Global Index Domain Propagation
- Database Communication
- Statistical / Financial / GUI Functions
- XML Support
- Solver Callbacks

Unique Features



- End-User GUI Builder
- Units of measurement
  - automatic unit checking in model
  - unit conventions for switching in end-user interface
- Modeling of time
  - horizons for period-based modeling
  - calendars for calendar-based interfacing
  - flexible support for aggregation and disaggregation
- White Box Outer Approximation
- Multi-Agent Technology

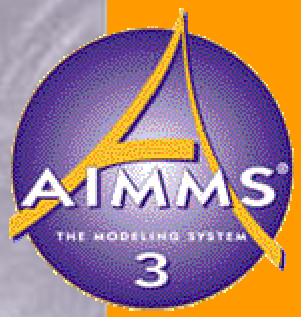
Transportation Demo

- Inspect Model Structure
- Extend the Model

- Introduce Template Page
- Copy Existing Page
- Adjust Graphical Network Object
- Introduce Page Navigation Buttons
- Specify Page Resize Behavior

DEMO

AIMMS Outer Approximation for MINLP



$$Min\ z = c^T y + f(x)$$

s.t.

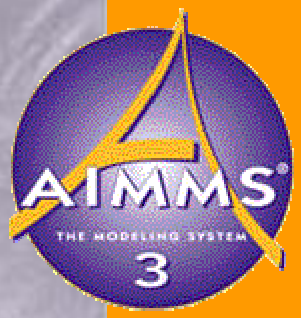
$$A\ y + h(x) = 0$$

$$B\ y + g(x) \leq 0$$

$$C\ y + D\ x \leq d$$

$$x \in X = \{\ x \in R^n \mid x^L \leq x \leq x^U\ \}$$

$$y \in Y = \{\ 0, 1\ \}^m$$



Outer Approximation Solution Procedure

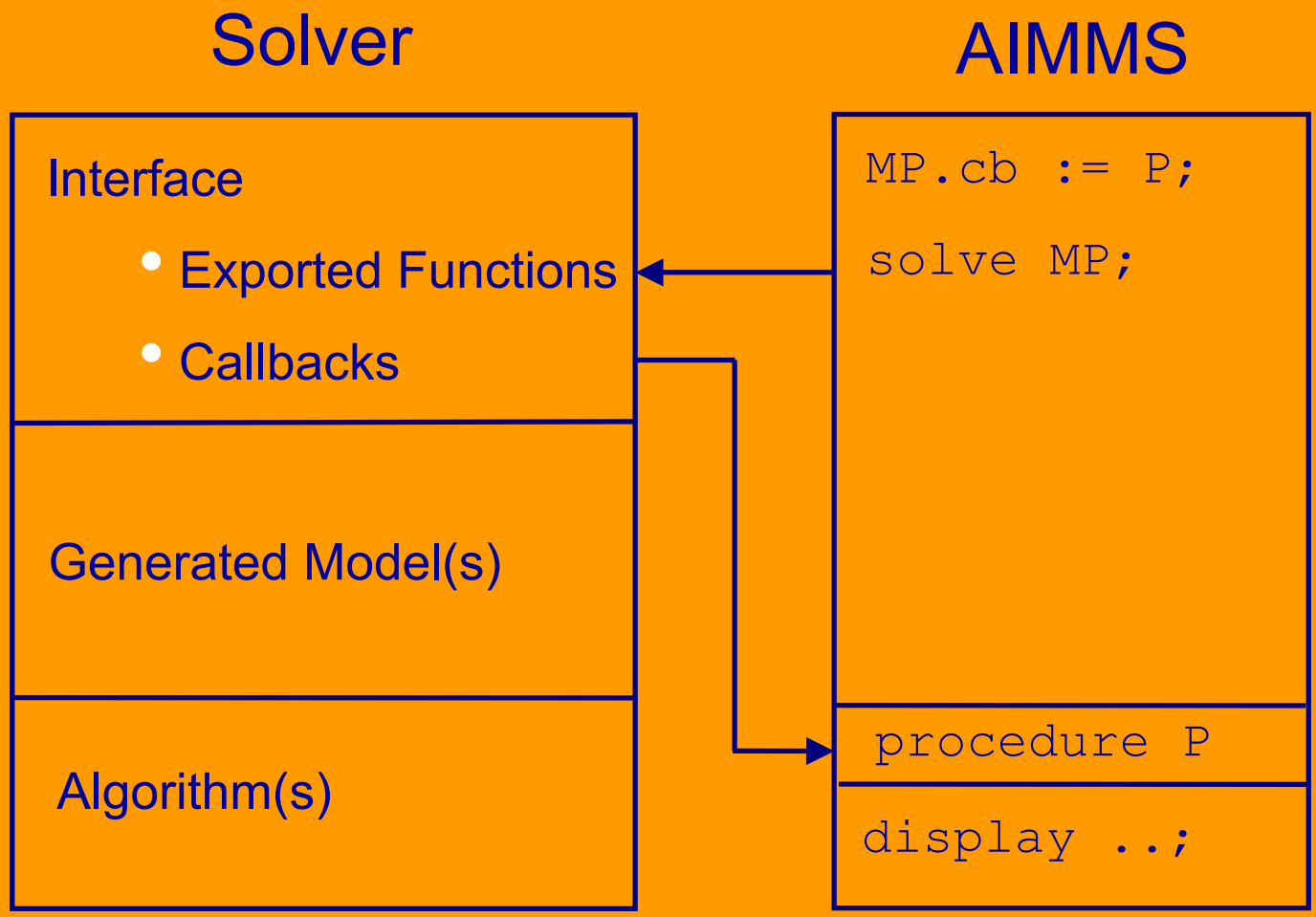
Solve the initial (relaxed) NLP model

Repeat

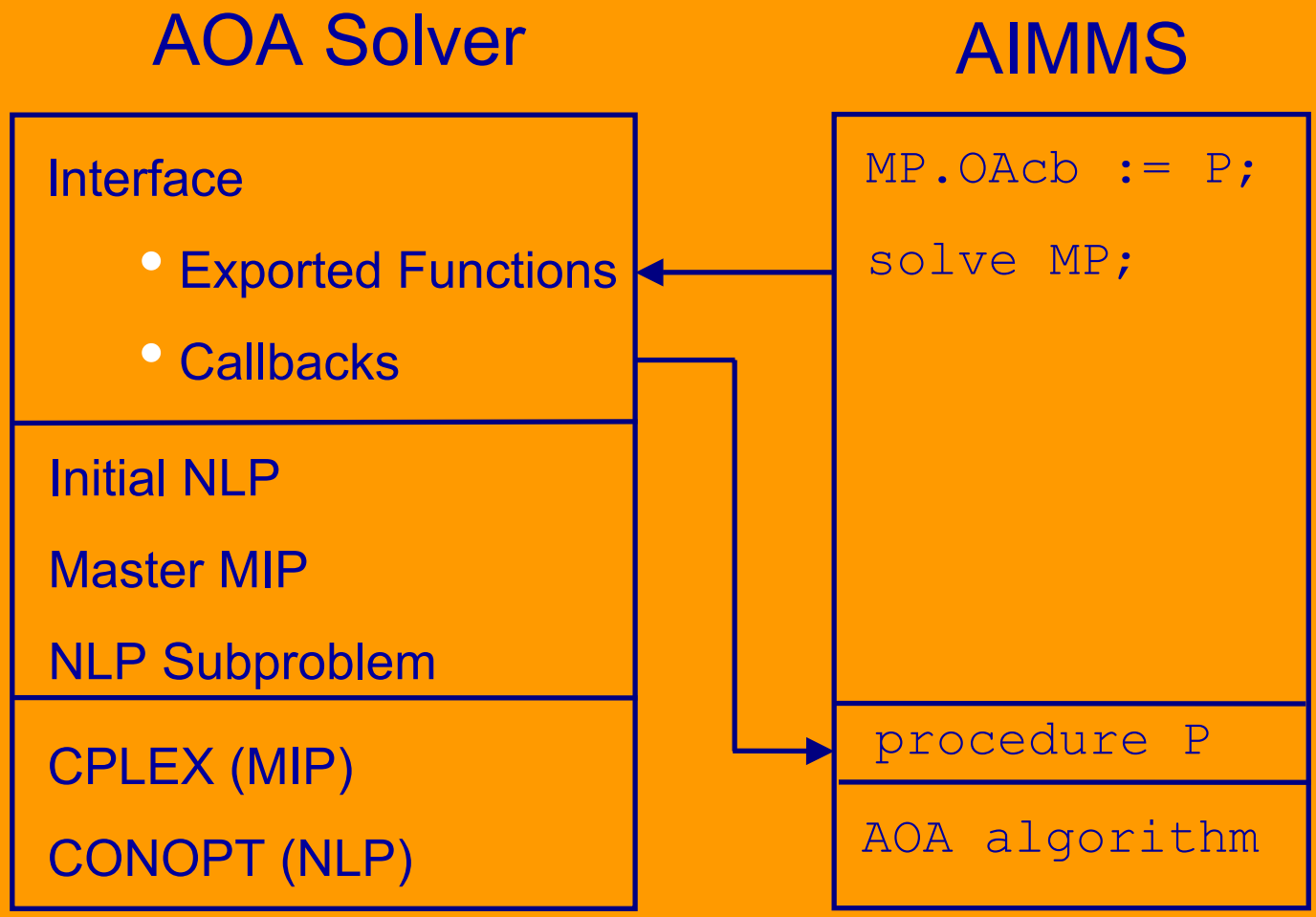
- Generate and solve master MIP model
- Solve the NLP sub-model
- Check termination criteria
- Prepare for next iteration

End Repeat

Solver Communication



Open AOA Solver Communication



Outer Approximation Solution Procedure in AIMMS

Reset all state parameters

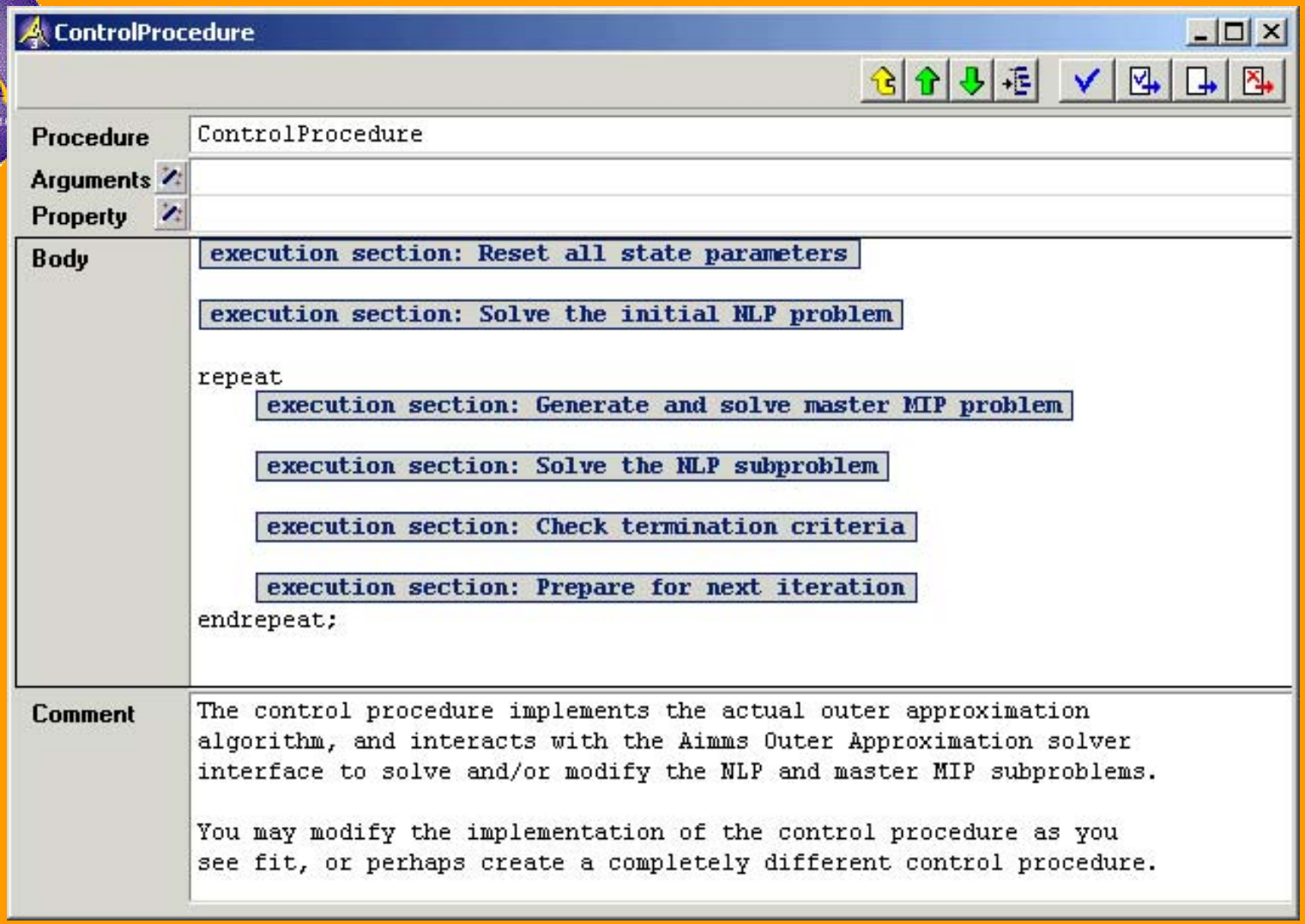
Solve the initial NLP problem

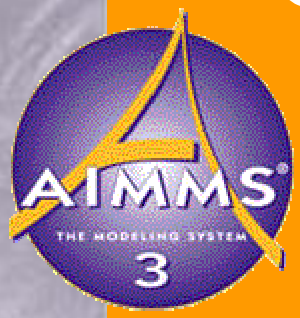
Repeat

- Generate and solve master MIP problem
- Solve the NLP subproblem
- Check termination criteria
- Prepare for next iteration

Endrepeat

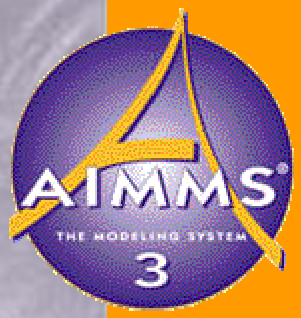
Outer Approximation Solution Procedure in AIMMS





Outer Approximation Interface Exported Functions

- Selected NLP Functions
  - NLPSolve
  - NLPIsFeasible
  - NLPSolutionIsInteger
- Selected Master MIP Functions
  - MasterMIPSolve
  - MasterMIPSolveToNextIntegerNode
  - MasterMIPIsFeasible
  - MasterMIPAddIntegerSolutionEliminationCut
  - MasterMIPAddLinearizations
  - MasterMIPSetCallback
- Selected Solution Manipulation Functions
  - SolutionSave
  - SolutionRetrieve
  - SolutionReplace
  - SolutionDelete



Outer Approximation Possible Uses of Openness

- User-specified switch between NLP solvers
- User-specified control (through MIP callback) over intermediate integer solutions for later evaluation
- User-specified points for adding linearizations
- User-specified procedures to adjust the penalties associated with linearizations in the Master MIP objective function

AIMMS Outer Approximation for Extended MINLP

$$Min\ z = f(x,y)$$

s.t.

$$h(x,y) = 0$$

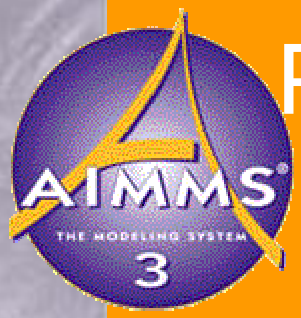
$$g(x,y) \leq 0$$

$$C\,y + D\,x \leq d$$

$$x \in X = \{\, x \in R^{n} \mid x^{L} \leq x \leq x^{U} \,\}$$

$$y \in Y = I^{m}$$

AIMMS Outer Approximation Reactor Production Scheduling

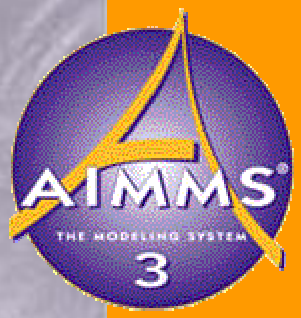


20 – Product Scenario

- AIMMS NLP Branch & Bound
  - Solver did not terminate
  - First integer solution 32.53 after 25 seconds
  - Best integer solution 32.13 after 250 seconds
  - Optimal integer solution not found
- AIMMS Outer Approximation
  - 20 integer solutions in 500 seconds
  - Best integer solution 32.01 after 85 seconds
  - Optimal integer solution not found

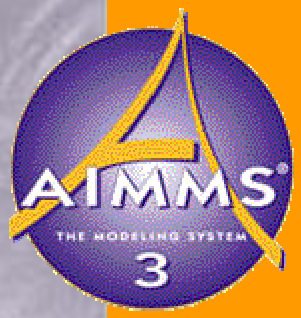
DEMO

AIMMS Measurement Concepts



- Quantities (Expresses what is measured)
  - Basic (e.g. Length, Time)
  - Derived (e.g. Velocity)
- Units (Expresses a reference measurement)
  - Base Units (e.g. [m], [s], [m / s])
  - Compound Units (e.g. [Hz] = [1 / s])
  - Derived Units (e.g. [mile] -> [m] : # -> # *1609)
- Unit-valued Parameters
  - NutrientUnit(n) := data { Energy : [KJ], Protein : [mg], Iron : [%RDA] }

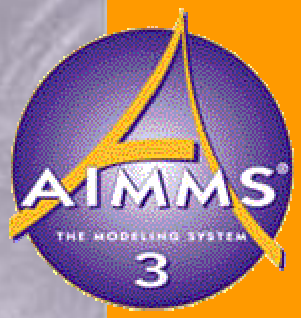
AIMMS Measurement Concepts



- Unit-valued Parameters
  - NutrientUnit(n) := data { Energy : [KJ],
    Protein : [mg], Iron : [%RDA] }
  - FoodUnit(f) := data { Cola : [liter],
    Carrots : [pounds], Burgers : [-] }
- Numerical Parameter

  identifier : NutrientValue
  index domain : (f,n)
  unit : NutrientUnit(n) / FoodUnit(f)

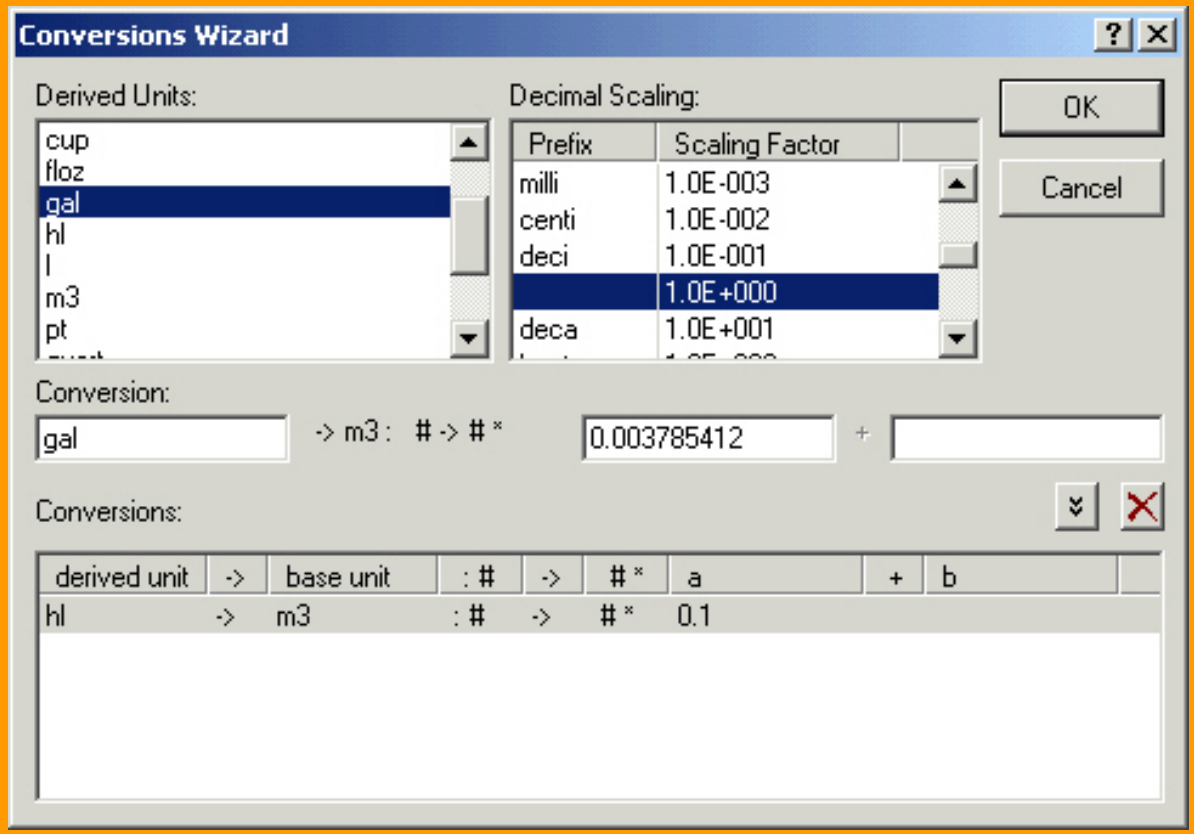
Unit Consistency



- Parameter
  - identifier : A
  - unit : [m / s]
- Parameter
  - identifier : B
  - unit : [km / hour]
- Parameter
  - identifier : C
  - unit : [mile / hour]
  - definition : A + B + 3 [km / s]

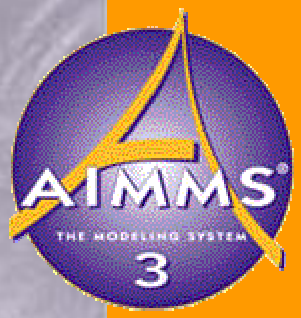
Unit Support

- Wizard that uses database with
  - SI Quantities
  - Common Units
  - Decimal Scaling



DEMO

AIMMS Link with Databases



- AIMMS connects to ODBC / OLE DB-compliant databases (Access, SQL Server, Oracle, etc.)
- AIMMS supports the use of
  - Database tables,
  - Database queries
  - Database procedures
  - Direct execution of SQL statements

Link Specification

- A Data Source
- A Table Name
- A Mapping of
  - primary key columns to AIMMS index domain
  - data columns to AIMMS identifiers

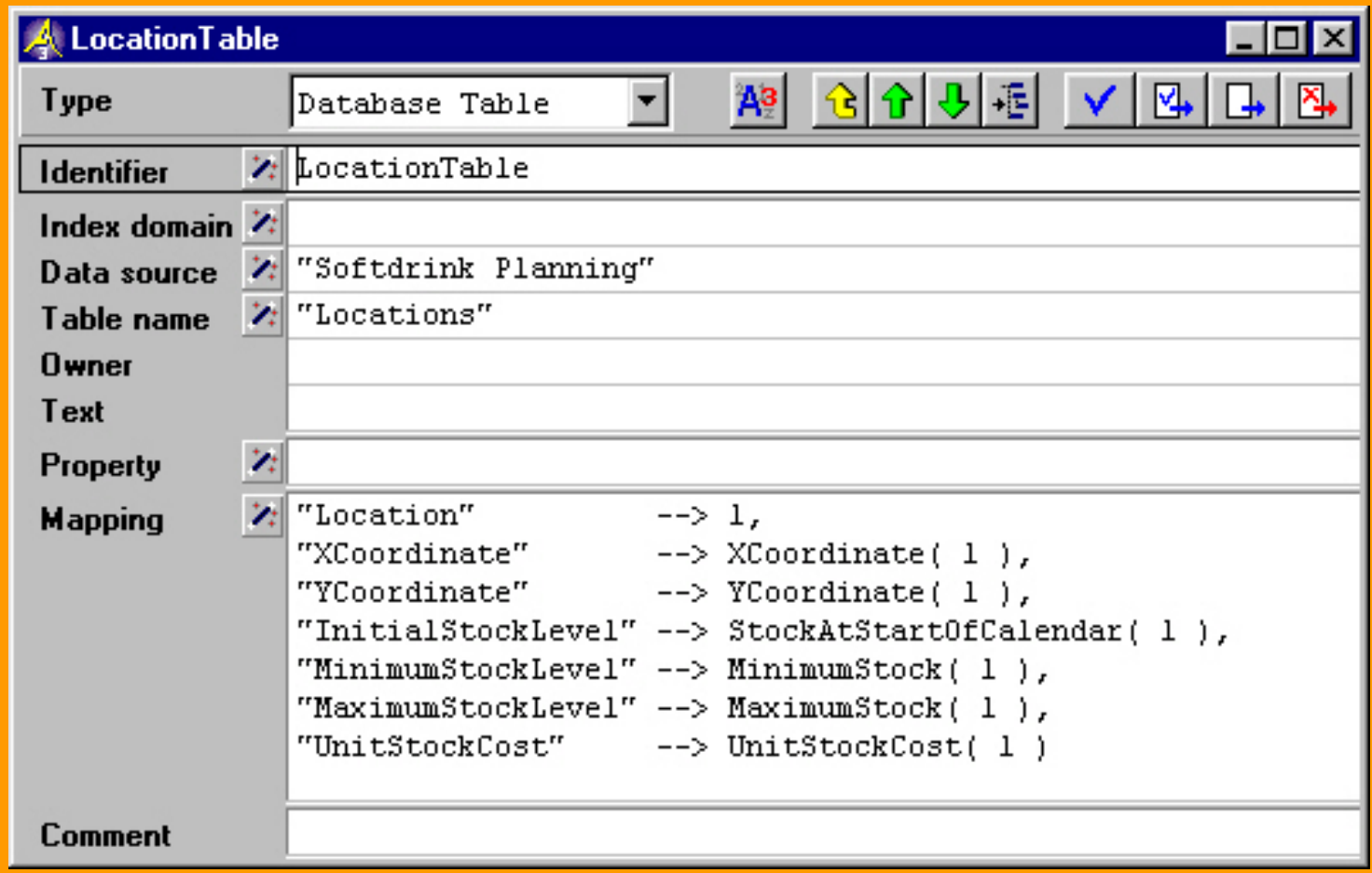
Database Table

Locations : Table

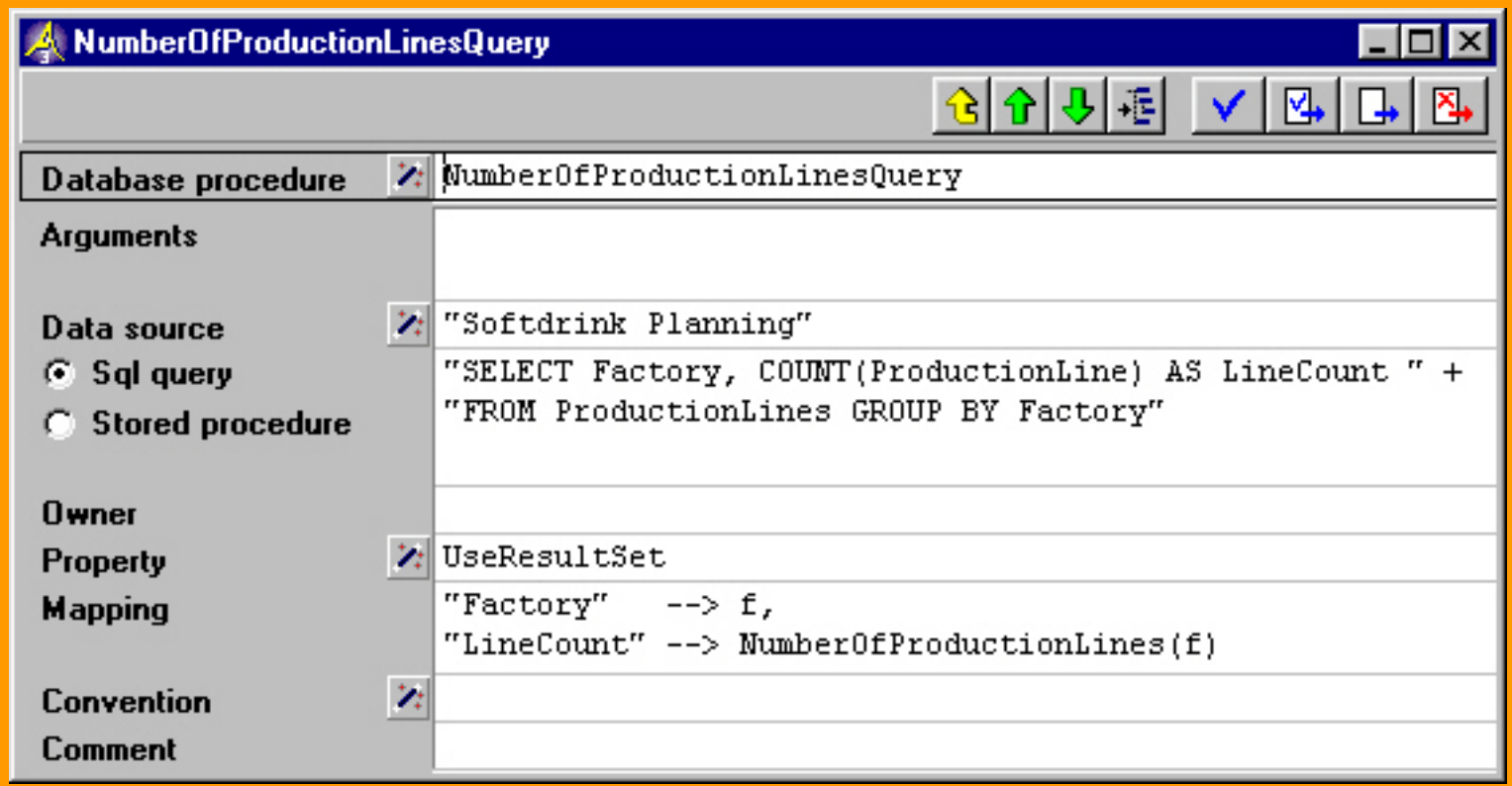
	Location	XCoordina	YCoordinat	UnitStockCos	InitialStockLe	MinimumStockLe	MaximumStockL	NumberOfInhabitants
▶	Amersfoort	5.377	52.158	0.07	2208	460	2760	123367
	Amsterdam	4.88	52.376	0.07	14016	2920	17520	727053
	Apeldoorn	5.953	52.213	0.07	2736	570	3420	152860
	Arnhem	5.917	51.982	0.07	2352	490	2940	137222
	Assen	6.561	53.011	0.07	1056	220	1320	57376
	Breda	4.778	51.588	0.07	2928	610	3660	159042
	Den Bosch	5.308	51.701	0.07	2208	460	2760	128009
	Den Haag	4.303	52.079	0.07	8064	1680	10080	440743
	Den Helder	4.754	52.958	0.07	1056	220	1320	59590
	Deventer	6.158	52.263	0.07	1488	310	1860	82621
	Dordrecht	4.679	51.794	0.07	2160	450	2700	119462
	Eindhoven	5.461	51.432	0.105	8330	8330	33320	199877
	Emmen	6.885	52.788	0.07	1920	400	2400	105497
	Enschede	6.89	52.22	0.07	2784	580	3480	148814
	Groningen	6.574	53.226	0.07	3024	630	3780	171193
	Haarlem	4.618	52.382	0.105	4310	4310	17240	148262
	Leeuwarden	5.782	53.212	0.07	1632	340	2040	88762
	Maastricht	5.696	50.857	0.07	2256	470	2820	121479
	Nijmegen	5.845	51.84	0.07	2496	520	3120	151864
	Rotterdam	4.482	51.929	0.07	10272	2140	12840	592665
	Tilburg	5.071	51.568	0.07	3312	690	4140	190559
	Utrecht	5.118	52.107	0.07	4080	850	5100	232718
	Venlo	6.158	51.374	0.07	1200	250	1500	64580
	Vlissingen	3.571	51.458	0.07	768	160	960	44530
	Zwolle	6.09	52.522	0.105	2790	2790	11160	104431

Record: 1 of 25

Mapping Columns to Identifiers



Database Procedure



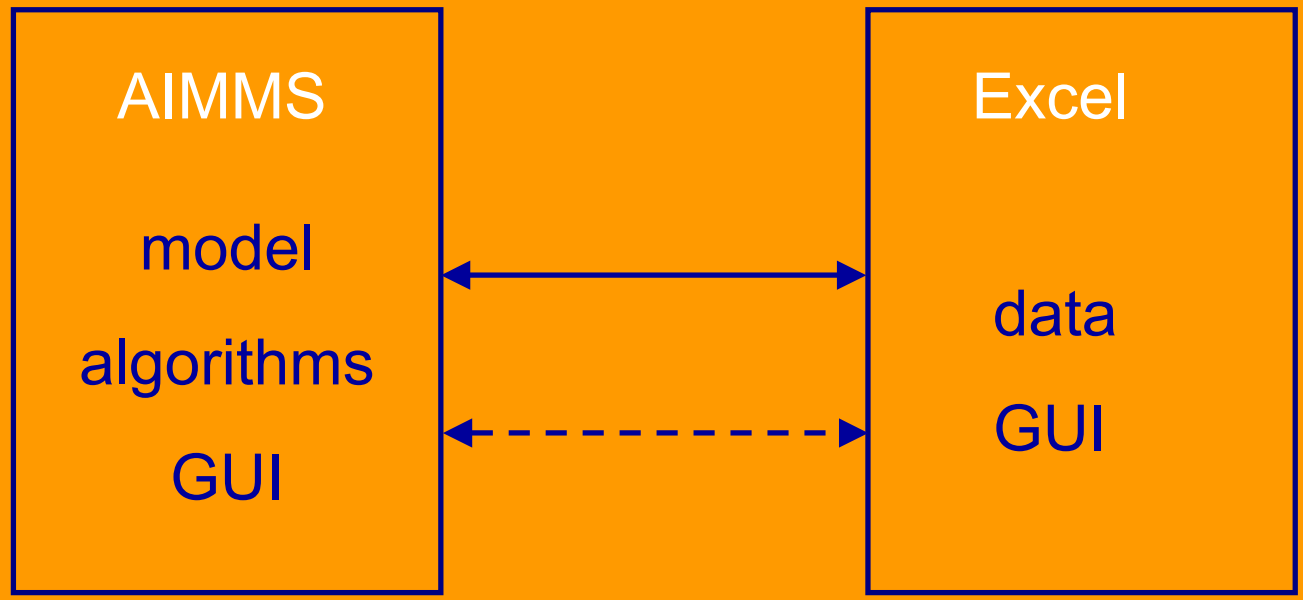
DEMO

AIMMS Multi-language Support

- AIMMS Unicode : support for double-byte (i.e. Asian, Russian) character sets
- Easy selection of language in end-user interface
- Support for automatic extension of existing models with language database

DEMO

AIMMS and Spreadsheets Combined



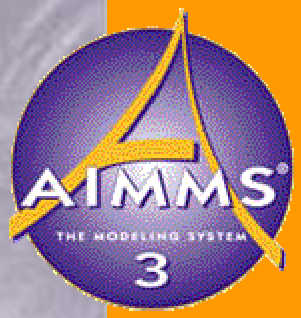
——— AIMMS in control : Excel Interfacing Functions

- - - - Excel in control : AIMMS Excel Add-In

Excel Add-In Features

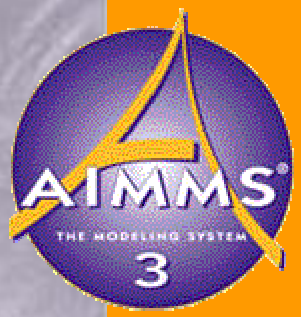
- Data Transfer between AIMMS and Excel
  - Retrieve Set/Array/Table Data
  - Assign Set/Array/Table Data
- Empty AIMMS Identifiers
- Update AIMMS Identifiers
- Run AIMMS Procedures
- Run Execution Sequence
- Run Excel Macro

DEMO



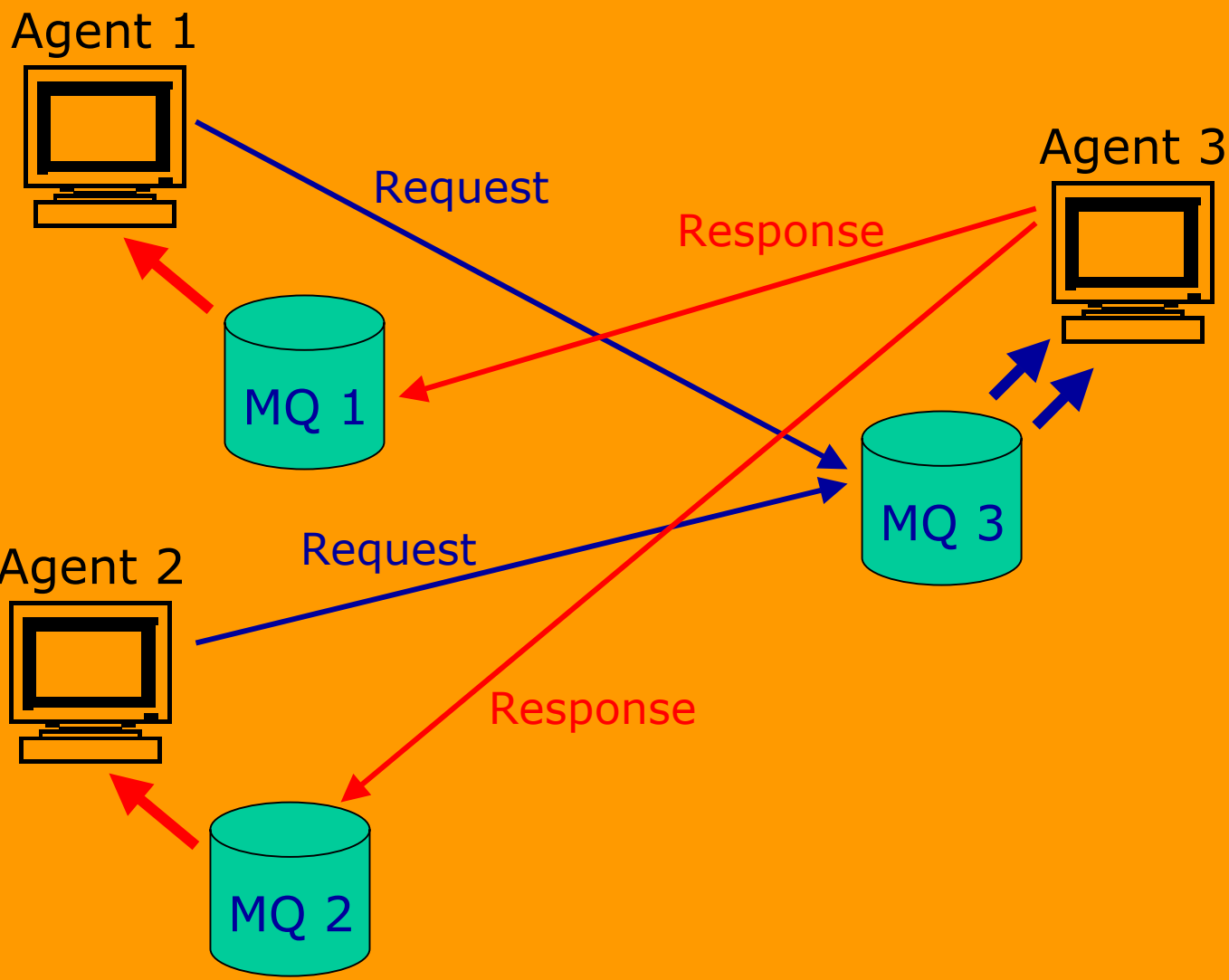
The AIMMS Multi-Agent Technology

What is a Multi-Agent System?

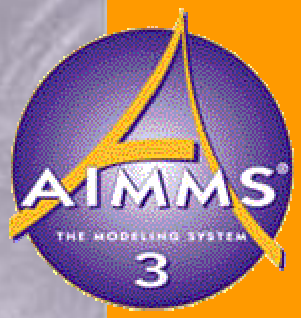


- An *agent* is a software program that independently performs its tasks on behalf of its user.
- A *multi-agent system* is a community of agents that communicate together to accomplish individual and/or common goals.

How do Agents Communicate?



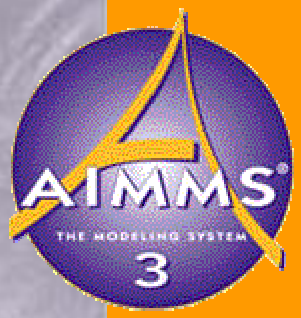
What are Agent Characteristics?



Agents are …

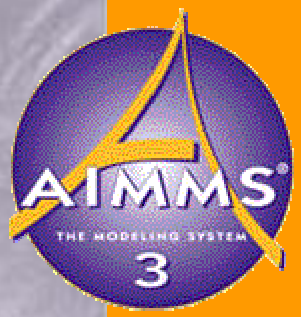
- *reactive*: they react to messages
- *proactive*: they initiate messages
- *intelligent*: they can make decisions
- *autonomous*: they control their own actions and internal states

Computing Paradigm



- *Parallel computing*: agents can reside on different computers.
- *Dynamic computing*: agents can enter and leave the community dynamically.
- *Non-deterministic computing*: the order of execution is not known a priori.

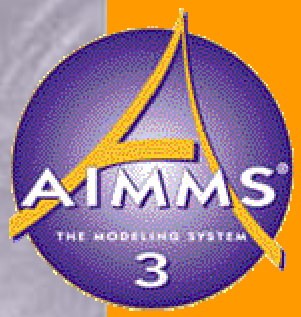
Example: A Purchasing Agent



A purchasing agent independently:

- checks inventories at regular moments
- determines order quantities
- requests price information from suppliers
- places orders
- updates the administration
- etc.

Example: Auctions (B2B)



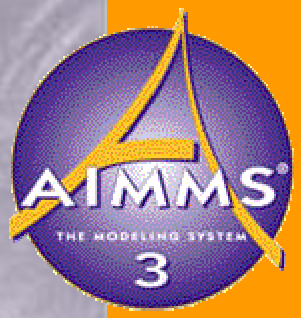
Bidding agents:

- supply bids each round
- decide on how to modify the bids
- know when to stop

Market clearing agent:

- receives the bids each round
- informs the bidding agents
- determines final allocation

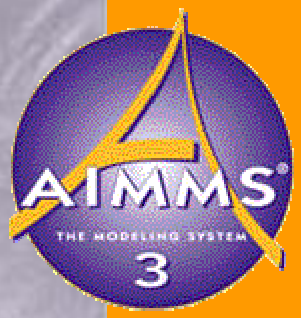
Example: Control Systems



Different specialized agents:

- to observe and register plant measurements
- to analyze these measurements
- to decide when and what to communicate with other agents
- to exercise control when needed

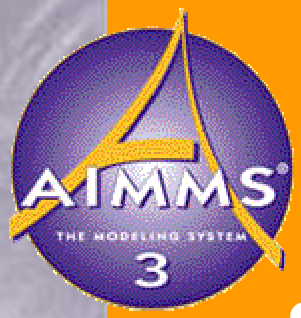
Example: Algorithms



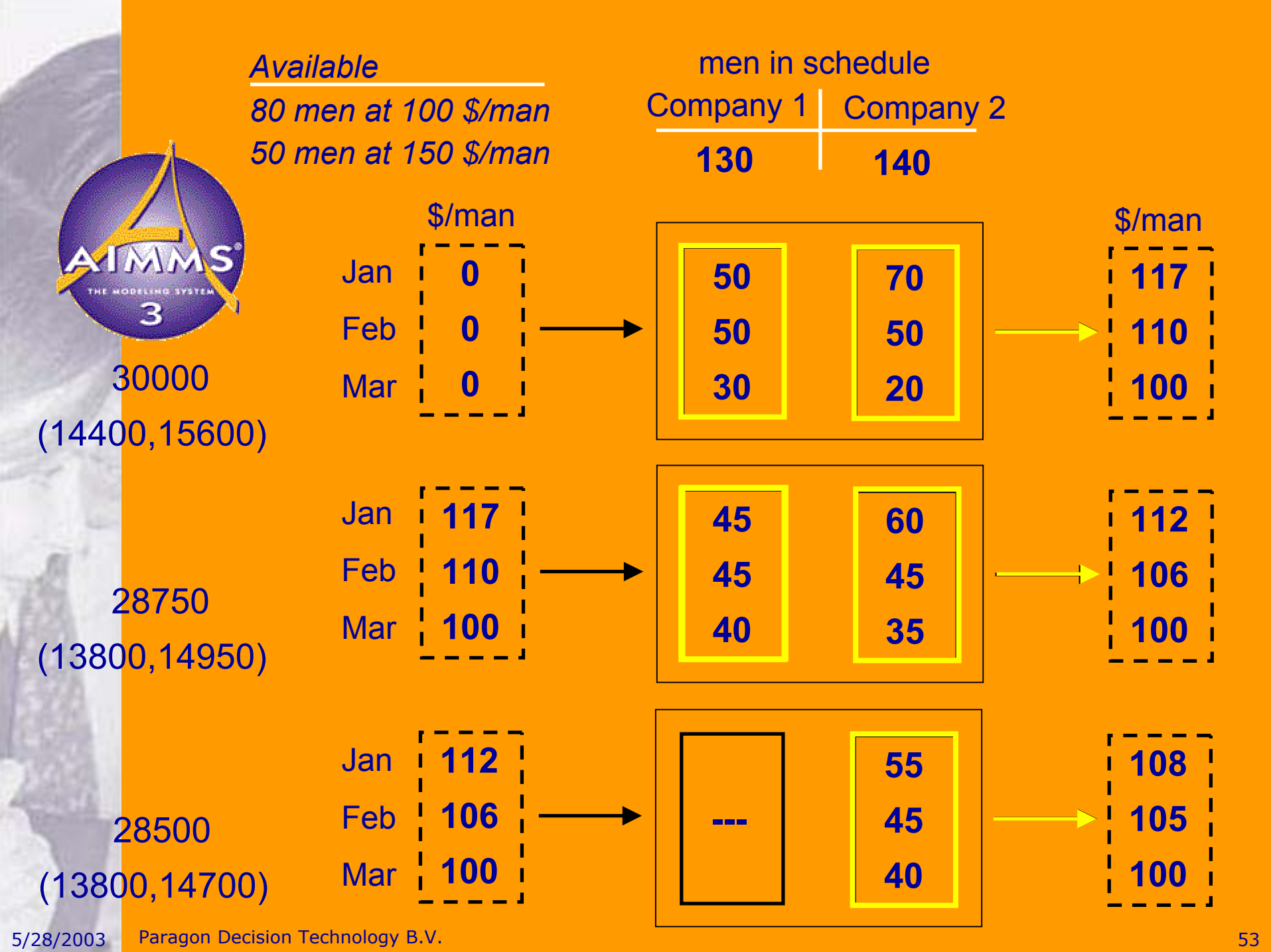
Worker agents to solve:

- sub-models in branch-and-bound
- experiments in Monte Carlo optimization
- scenario's in stochastic programming

Example: Pooled Resource Allocation

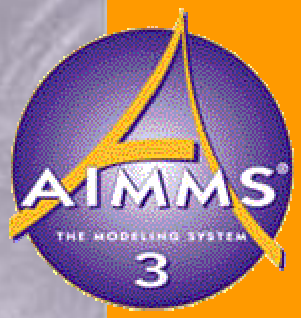


- Independent contractors (agents)
  - generate improved resource use schedules
- Pooled resource coordinator (agent)
  - selects one schedule per contractor
  - minimizes total resource cost
  - updates resource price


Available
80 men at 100 $/man
50 men at 150 $/man
men in schedule
Company 1
Company 2
130
140
$/man
Jan
Feb
Mar
0
0
0
50
50
30
70
50
20
$/man
117
110
100
30000
(14400,15600)
Jan
Feb
Mar
117
110
100
45
45
40
60
45
35
112
106
100
28750
(13800,14950)
Jan
Feb
Mar
112
106
100

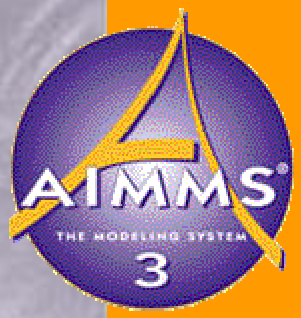
55
45
40
108
105
100
28500
(13800,14700)

AIMMS Agent Projects

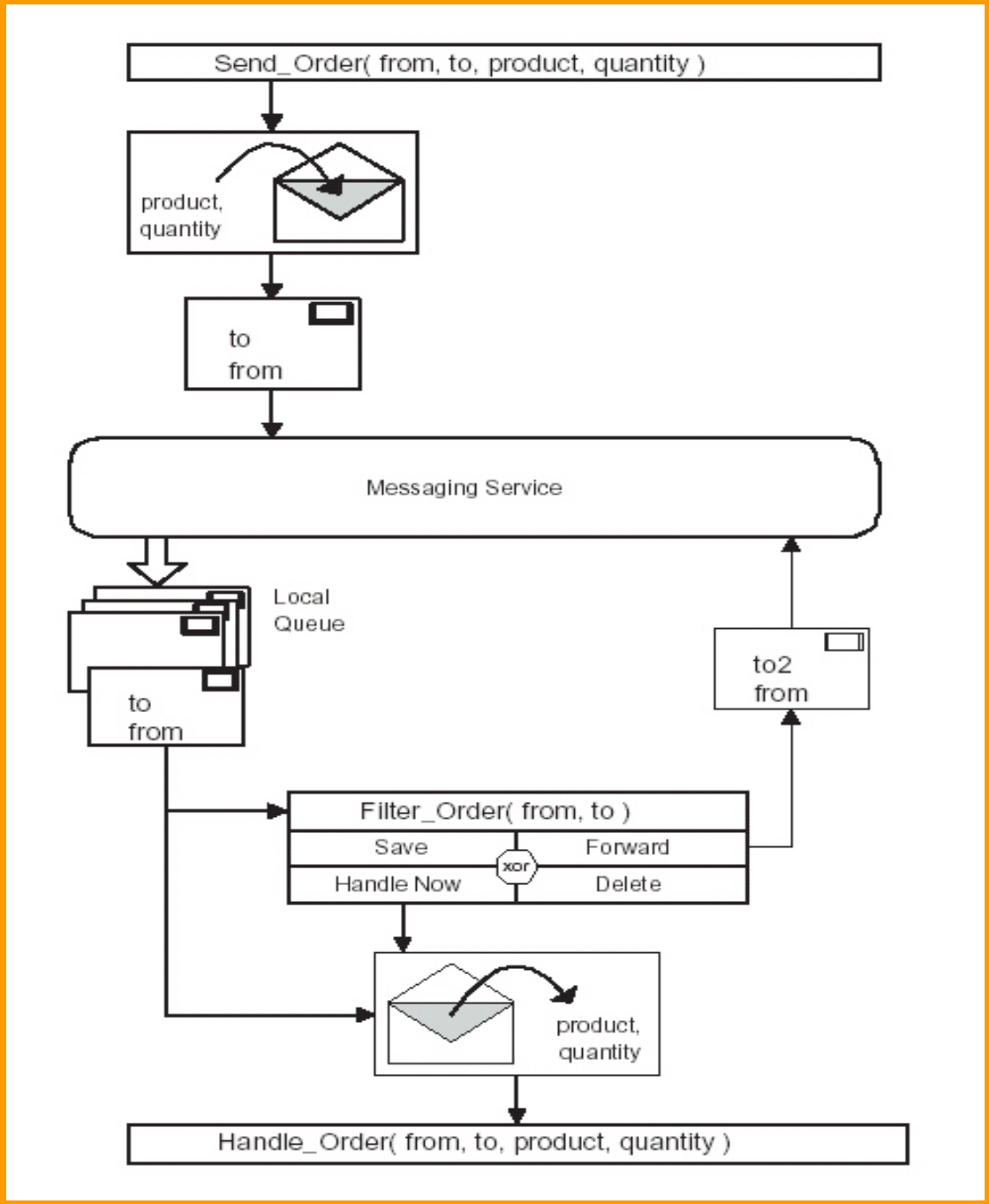


- Each AIMMS project contains one or more individual agents.
- Projects have a physical location in a local-area network.
- Each project has its own associated message queue.
- There is a hierarchy between projects to start and stop other projects and their associated queues.

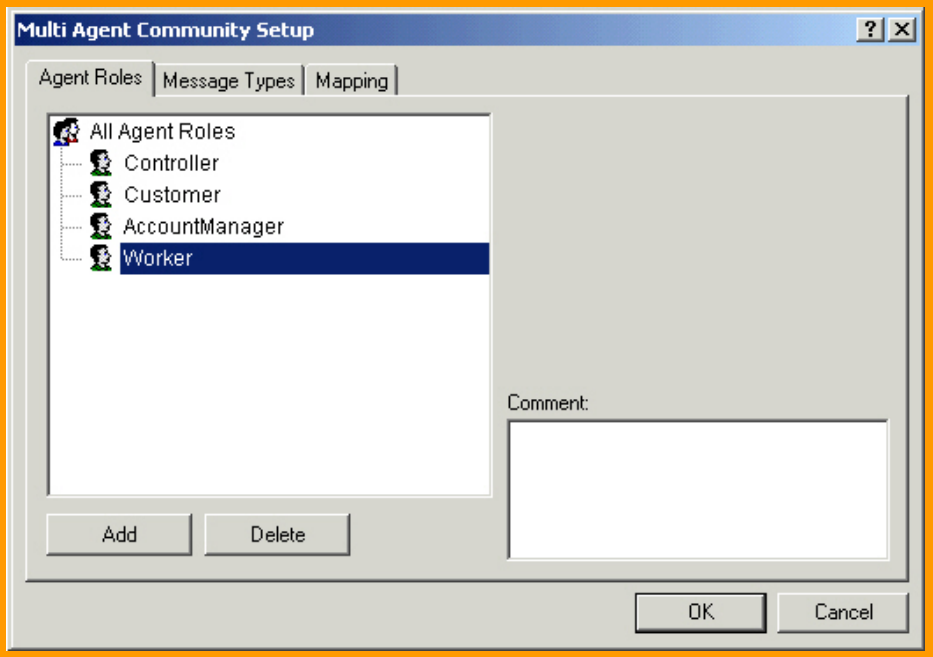
Individual AIMMS Agents

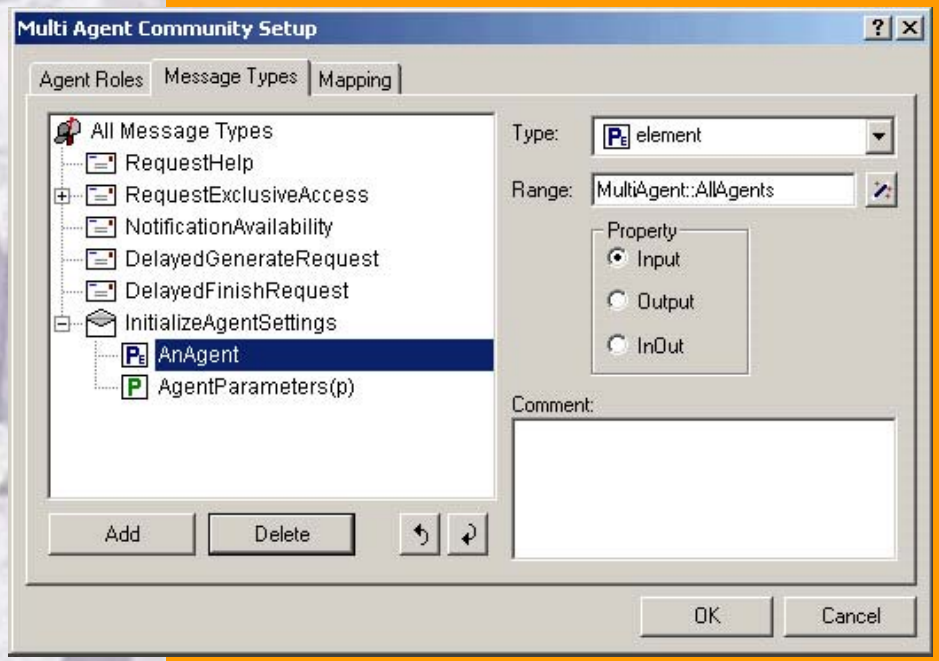


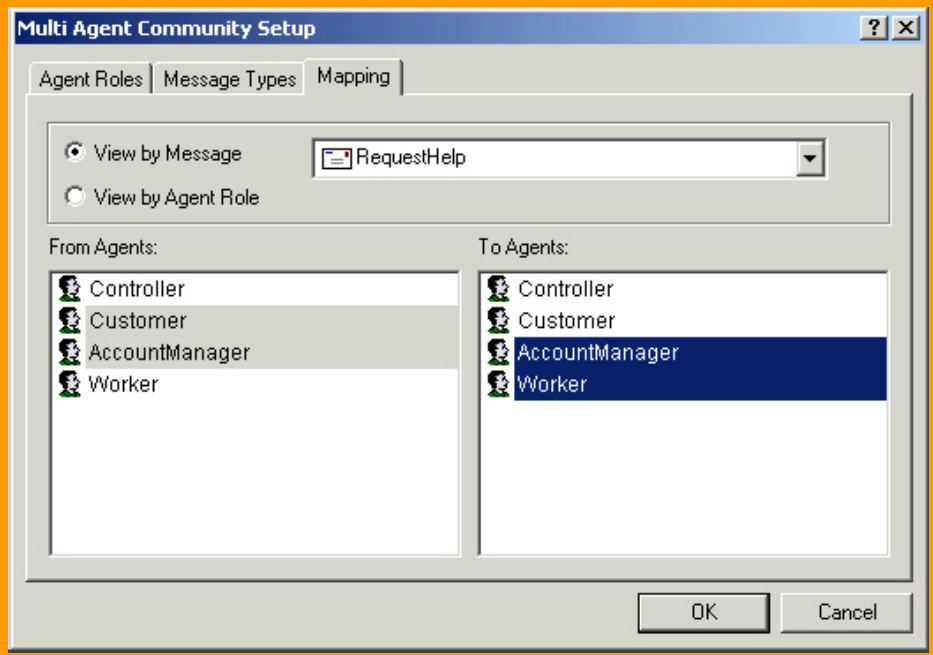
- Each individual AIMMS agent has its own *role* in an agent-based application.
- Several agents can have the same role.
- For each role there is an associated set of types of messages to be handled.
- For each type of message received, there is a separate AIMMS procedure to handle that particular type of message.


Send_Order(from, to, product, quantity)
product,
quantity
to
from
Messaging Service
Local
Queue
to
from
to2
from
Filter_Order(from, to)
Save
Forward
xor
Handle Now
Delete
product,
quantity
Handle_Order(from, to, product, quantity)

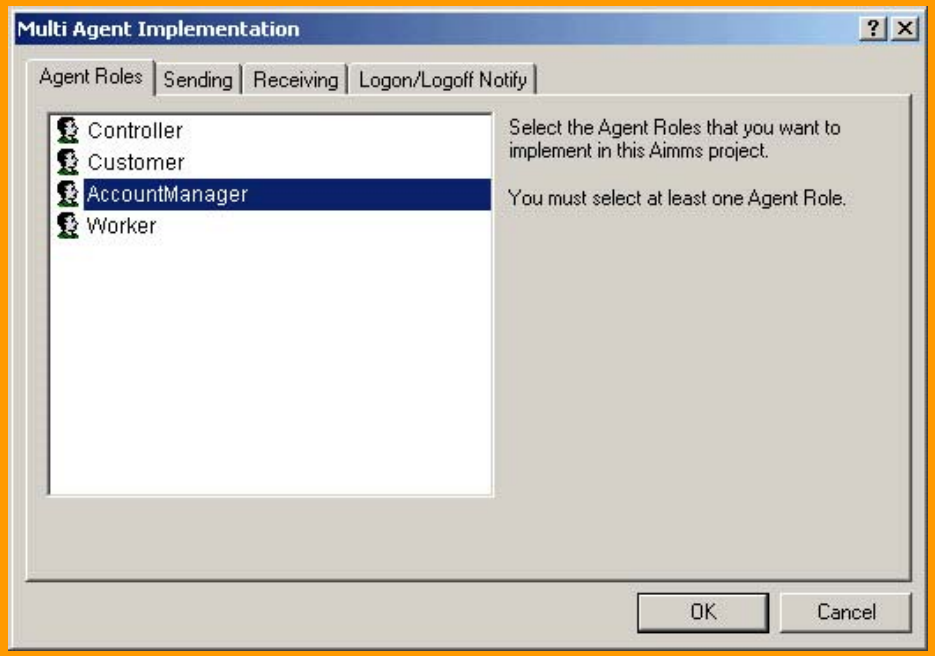
Agent Community Setup

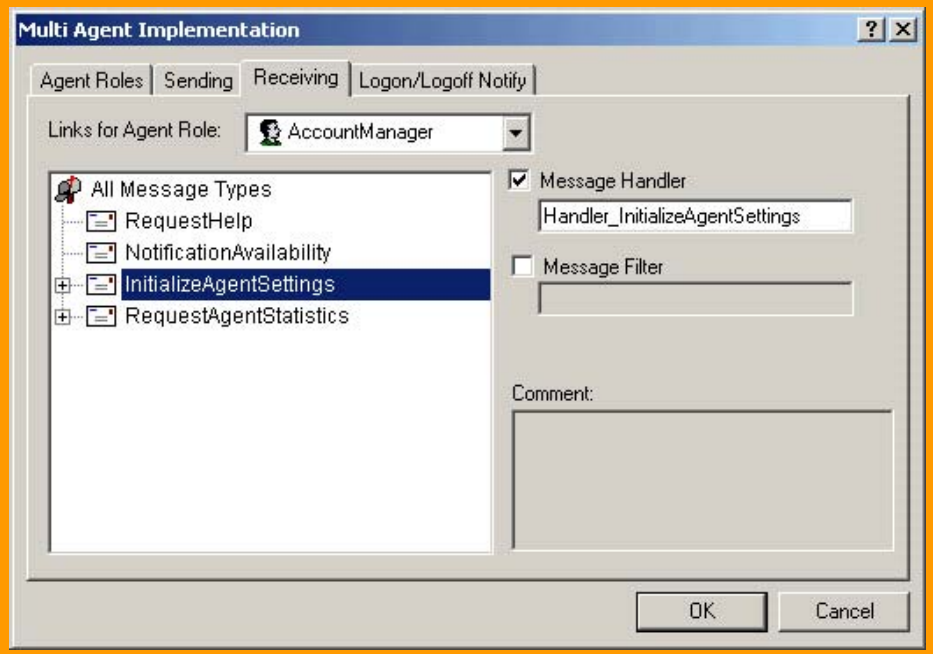






Specific Agent Setup





DEMO