

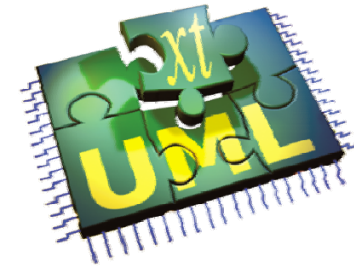
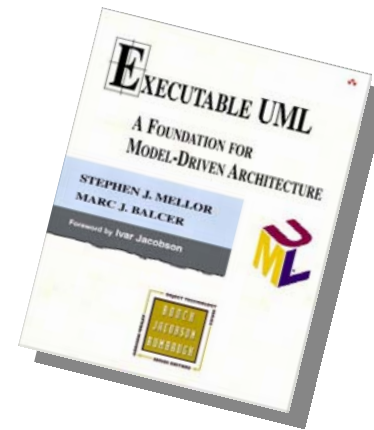
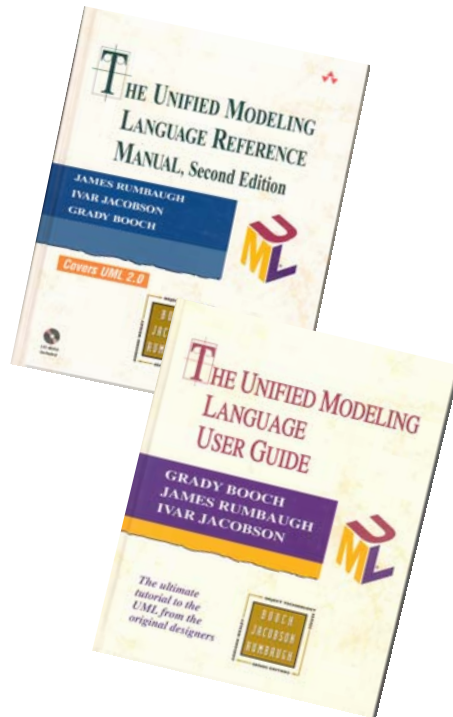
# Model Driven Development with xtUML and BridgePoint®

**Mentor  
Graphics®**

# xtUML – Executable and Translatable UML

## Unified Modeling Language

- Industry standard *notation*
- *Family* of languages

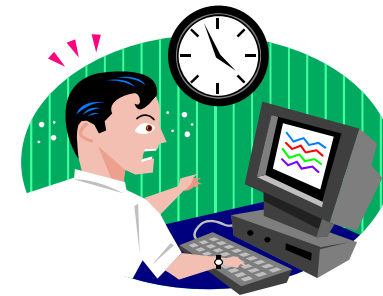
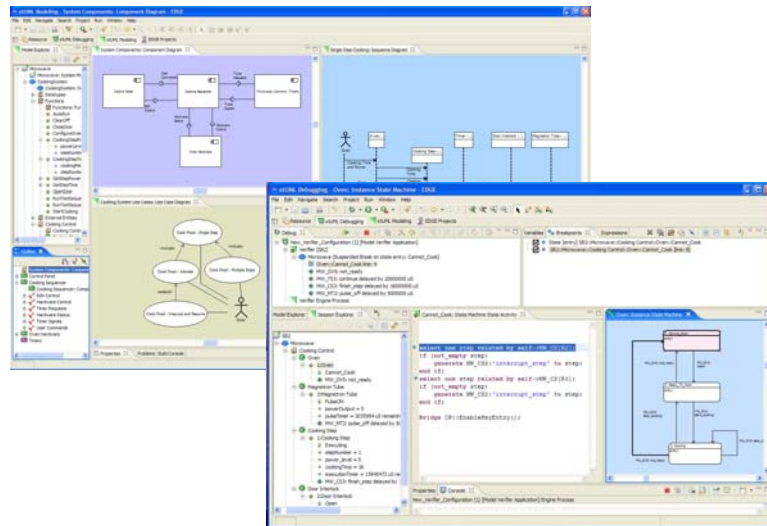


## “Executable UML”

- Defines a method, including:
  - Semantics of diagrams
  - Relationship between diagrams
  - Execution rules
  - Order of construction
  - Path to implementation

# Executable Models, Not Just Diagrams

- Test application design before coding it
- Result: Verified executable specification

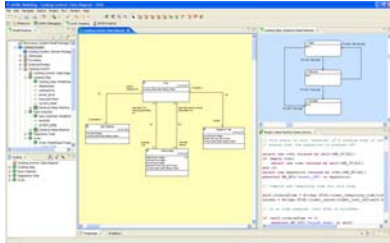


Drive existing design flows

Compile into optimized code...

# Compiling Models: It's not magic

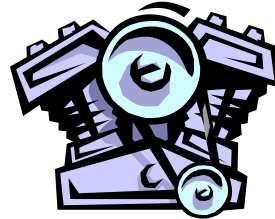
Application Models



Marking Data



Rules & Templates



Translation Engine

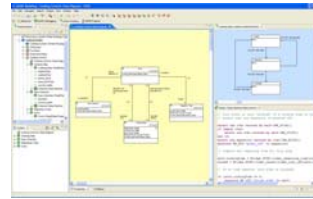
Optimized Code

```
// Now index is object's address (C recommended) current state.  
// Now keep in the hashtable index is 0, for opening event transitions.  
// Column index is (C recommended) state machine events.  
// =====  
void get_state_name(  
    Microworld_M0_01StateMachine * S ) {  
    // Row 0: (this table)  
    { EVENT_CANT_HAPPEN, EVENT_CANT_HAPPEN },  
    // Row 1: Microworld_M0_0_STATE_0 "Open"  
    { EVENT_IS_LOCKED, Microworld_M0_0_STATE_1 },  
    // Row 2: Microworld_M0_0_STATE_1 "Close"  
    { Microworld_M0_0_STATE_1, EVENT_IS_LOCKED }  
};  
  
// =====  
// Array of pointers to the object's state action methods.  
// Index is the (C recommended) number of the state action method to execute.  
// =====  
void get_state_action(  
    Microworld_M0_01StateMachine * S ) {  
    (ptr_state_action_t) 0,  
    (ptr_state_action_t) Microworld_M0_01StateAction_1, // "Open"  
    (ptr_state_action_t) Microworld_M0_01StateAction_2, // "Close"  
};
```

# Separation of Application from Implementation

- Subject-matter experts focus on application
  - Features and capabilities
  - Intricacies of the application

## Application Models



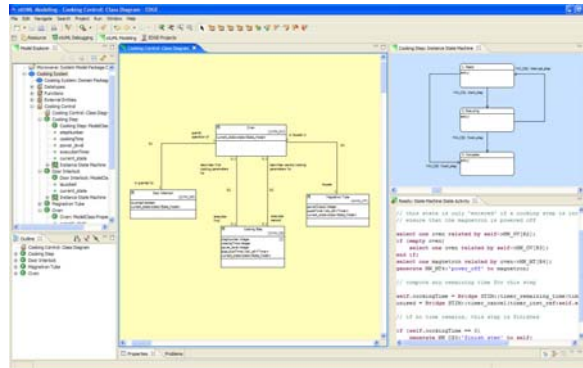
- Implementation experts focus on optimization
  - Faster, smaller
  - Less power
  - Lower cost

## Model Compiler



# Reusable IP(1): Application Models

Platform-independent  
Application Models

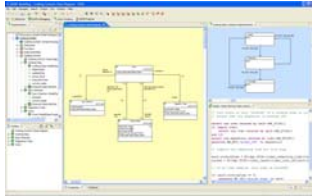


Reuse application models  
across platforms and  
product variants.

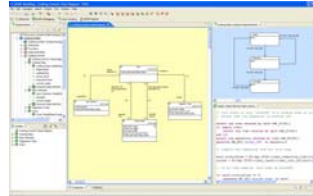


# Reusable IP(2): Model Compilers

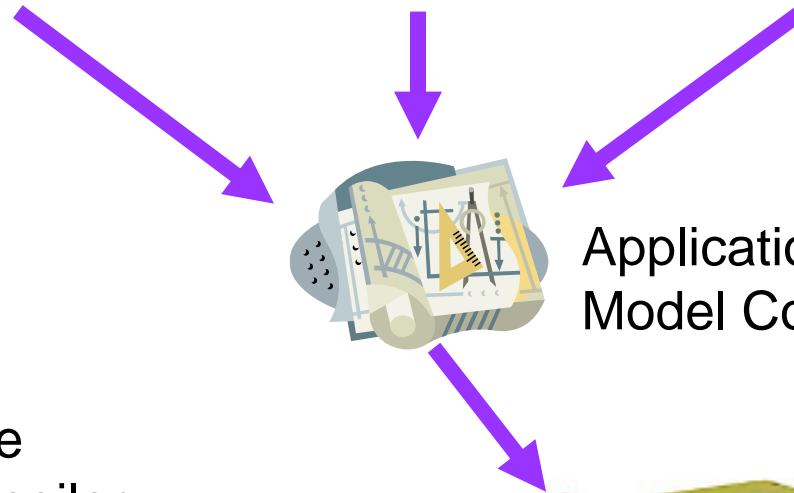
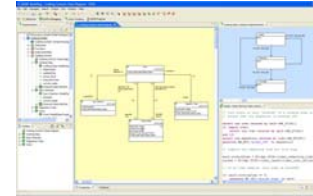
Site Link



Channel Controller



MME

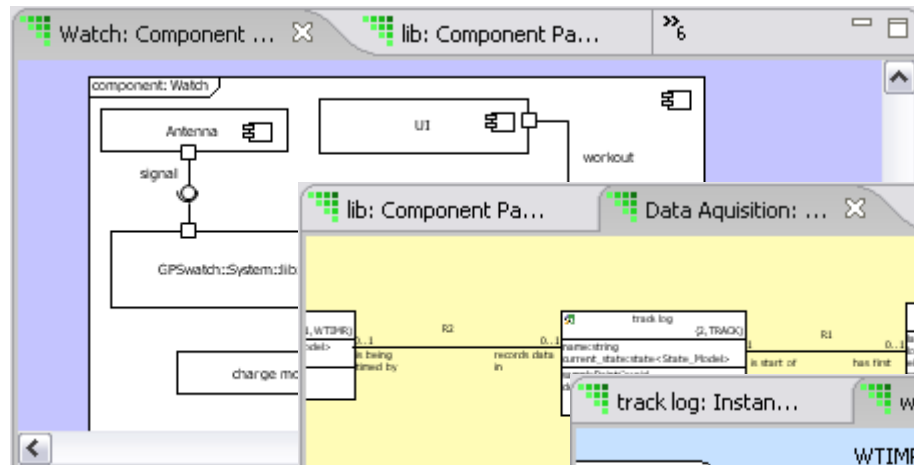


Application-independent  
Model Compiler

Reuse one  
model compiler  
across many  
applications.

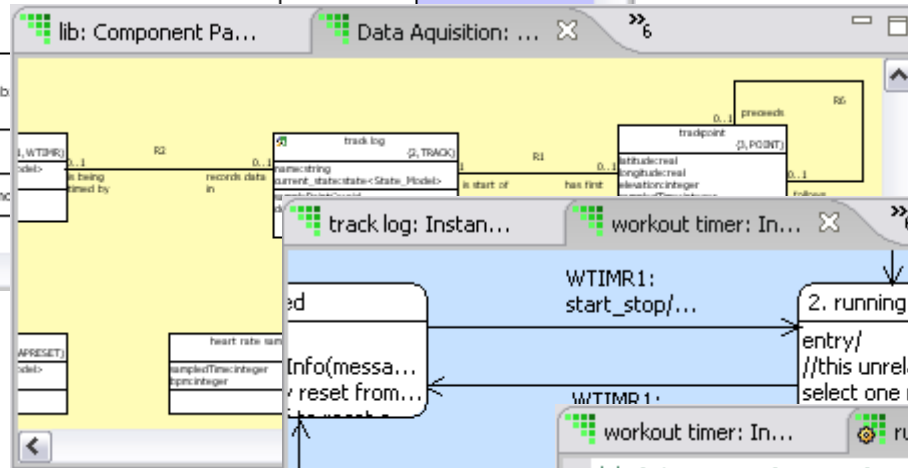


# xtUML Design flow



## Component Diagram

- Decompose the application
- Define Interfaces

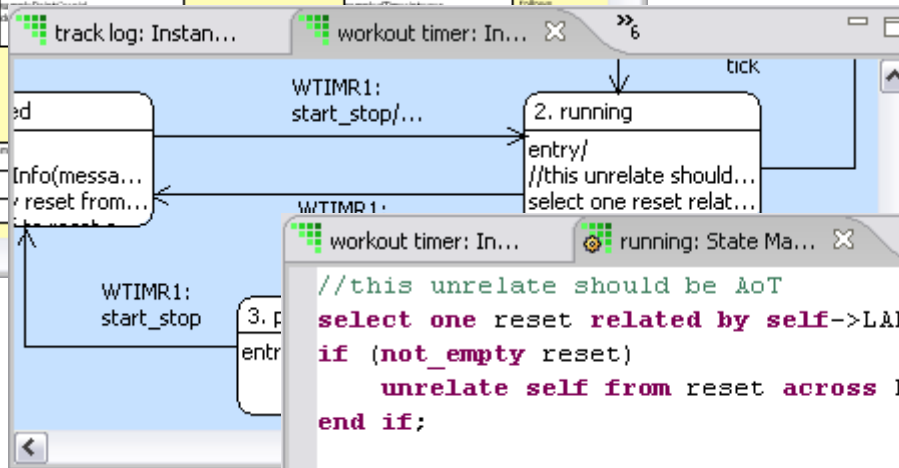


## Class Diagram

- Abstractions, associations
- Operations

## State Diagram

- Functional lifecycle
- Event handling



## Action Specification

- Processing

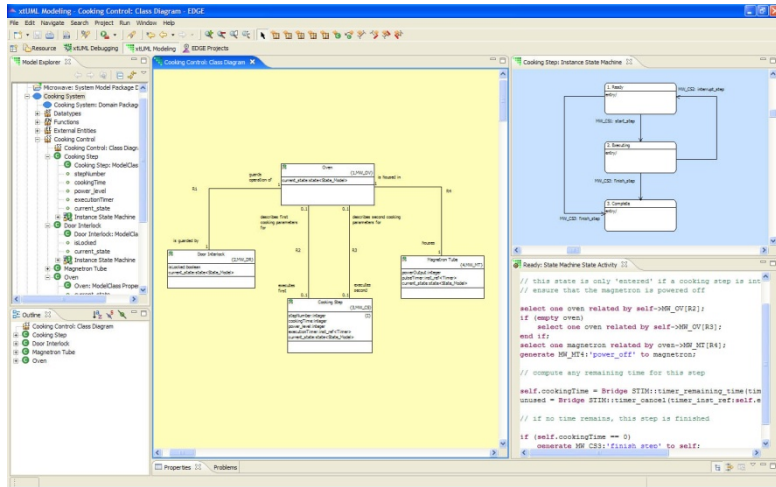
```
//this unrelate should be &oT
select one reset related by self->LAPRESET[R4];
if (not_empty reset)
    unrelate self from reset across R4;
end if;

self.seconds = self.seconds + 1;
create event instance tick of WTIMR2:'tick' to self;
t = TIM::timer_start( microseconds:1000000, event_
LOG::LogInfo(message:"timer tick");
```

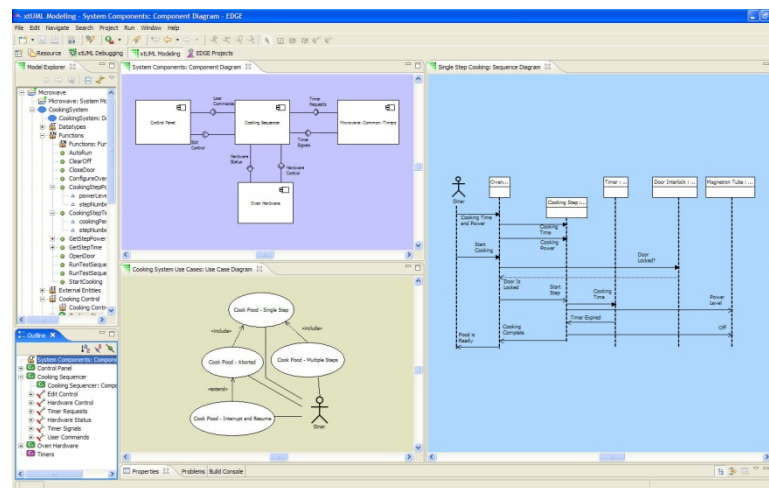
**Executable**  
**Translatable**



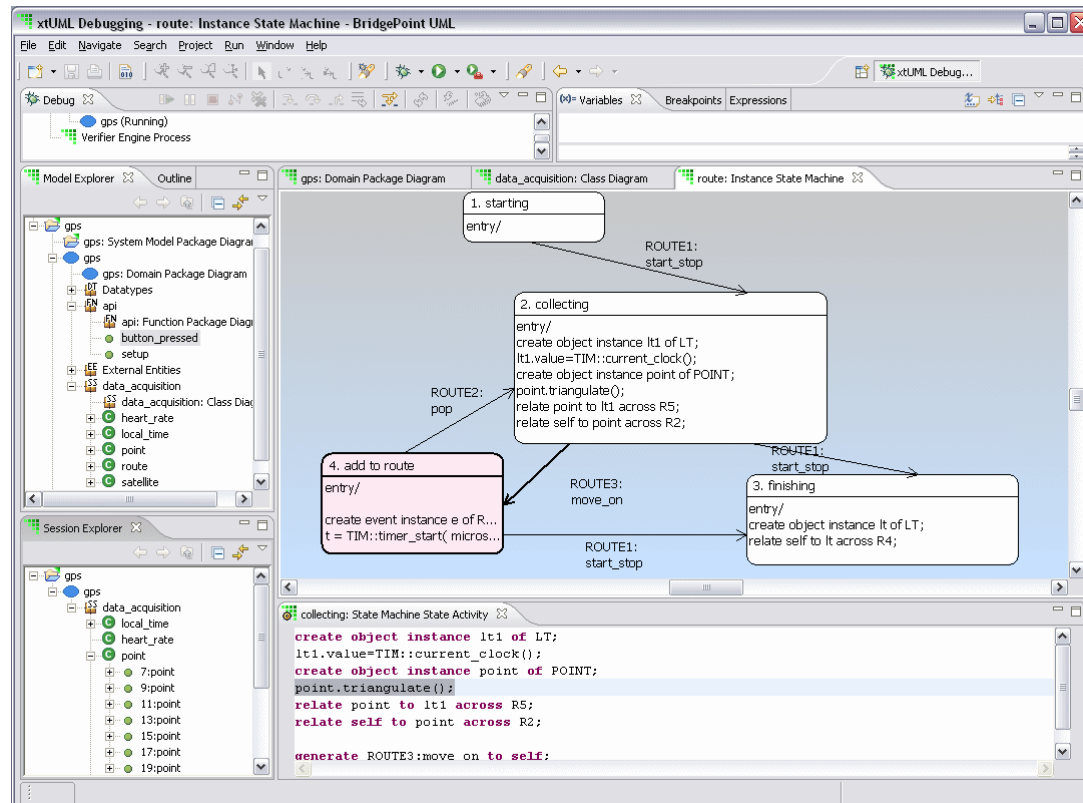
# BridgePoint Model Builder



- Intelligent model entry
- Intuitive navigation
- Syntax-aware OAL editor
- Flexible CM
- Eclipse framework



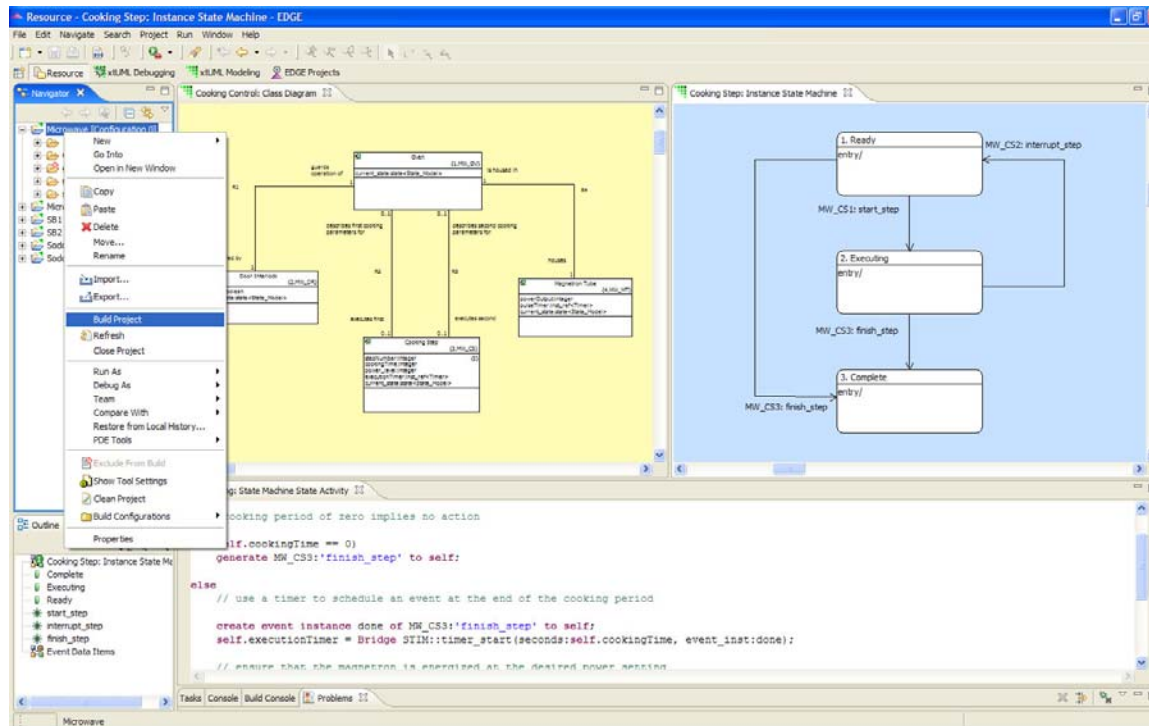
# BridgePoint Model Verifier



- Interpretive execution
- Interactive debugging
- Animation of models
- Connect to legacy

- Execute models before you generate or write code
- Run models immediately and incrementally
- Remove defects early

# BridgePoint Model Compilers



- Design patterns
  - Templates
- Translation rules
  - Query
  - Transform
  - Populate templates
- Marking
  - Like compiler flags
  - Determine which patterns are applied to each model element

- One-button build
  - Customize tool chain and build parameters
  - Generate build script for CM and batch
  - Buy or build model compiler

# BridgePoint xtUML is Proven Technology

- Wireless network controllers, base stations
- Telecom multi-processor cross-connect switches
- PBX, call processing
- Office equipment – copier/printer/fax/scanner
- Surgical X-Ray, ultrasound imaging
- Automotive HVAC, Audio, Navigation, Instruments
- Satellite launch control, navigation control, avionics
- Railroad control system
- Manufacturing robots, data collection
- Training simulations, mission planning
- Implanted Medical Devices

# Results

- 10x productivity increase over traditional approaches
  - Very large US defense contractor
  - Medium projects, 5-20 engineers
- 8x productivity increase over traditional approaches
  - Medium-sized Australian defense contractor
  - Small team, 3-5 engineers
- 30% productivity increase over traditional approaches
  - Very large US aerospace company
  - Large projects, 50-200 engineers
- European wireless telecom equipment maker
  - 40% reduction in code size
  - 9% increase in execution speed
  - Replacement of hand-written module with model-generated code

The background is a vibrant blue with a complex digital aesthetic. It features a network of thin white lines, some forming a grid and others connecting nodes. There are several circular patterns, some resembling orbits or data paths. Binary code (0s and 1s) is scattered throughout, appearing as if floating or being processed. The overall effect is one of high-tech connectivity and data flow.

**Mentor  
Graphics®**