Building a Startup with Standard ML

Steve Sims, Reactive Systems, Inc.

Reactive Systems, Inc.
Tomorrow's Software Today®

Outline

1. About Reactive Systems
2. Overview of Reactis

Using Standard ML (SML) to:

3. Build Reactis
4. Test Reactis
5. Build the RSI Intranet
6. Build the RSI website

7. Intangible benefits of SML use

Reactive Systems, inc

About Reactive Systems, Inc. (RSI)

1999 - Founded; based in Falls Church, VA

Small Business Innovation Research (SBIR) grants from National Science Foundation

2000 – Phase I ⎫
2001 – Phase II ⎬ Develop technologies for validating embedded control software

2002 – Launch of Reactis, tool supporting model-based testing and validation for Simulink/Stateflow.

2006 – Customer landscape:

- 29 automotive, 5 aerospace, 1 heavy equipment

- Australia, Canada, France, Germany, Great Britain, India, Japan, Taiwan, US, South Korea, Romania

Outline

1. About Reactive Systems

2. **Overview of Reactis**

3. Build Reactis

4. Test Reactis

5. Build the RSI Intranet

6. Build the RSI website

7. Intangible benefits of SML use

Using Standard ML (SML) to:

Reactive Systems, inc
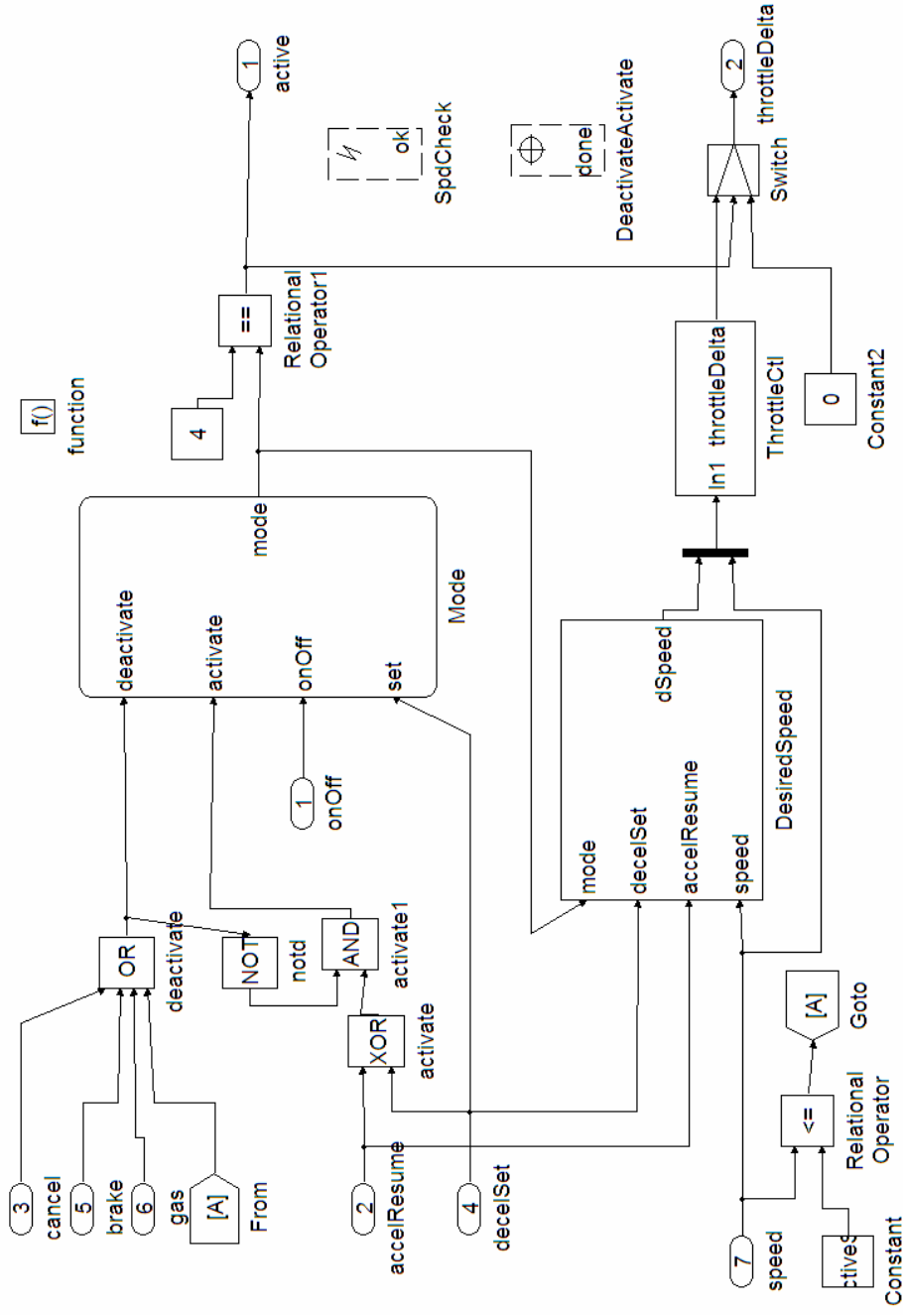
Interacting with Reactis

Users:

- create Simulink®/Stateflow® models outside Reactis

- load these models into Reactis

Reactis

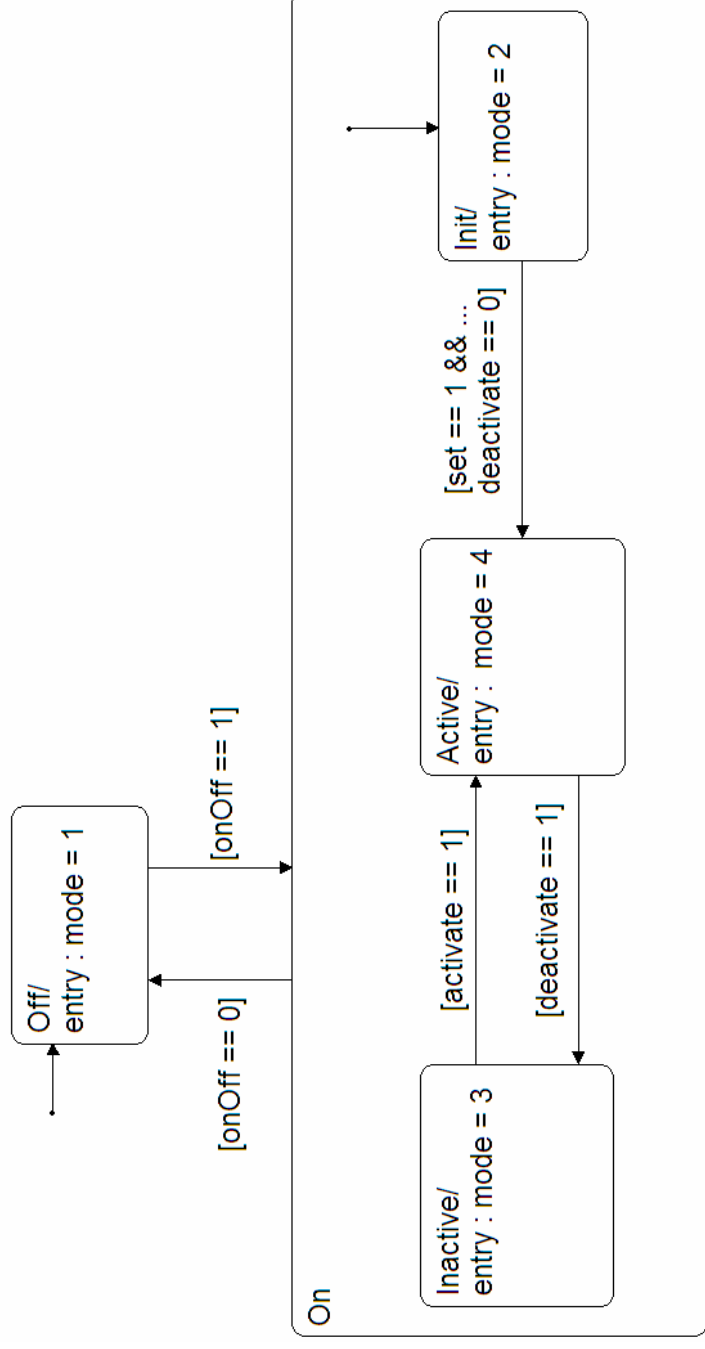**.mdl file**
Model

Simulink /
Stateflow
Modeling
Environment
The MathWorks

**Reactive
Systems, inc**
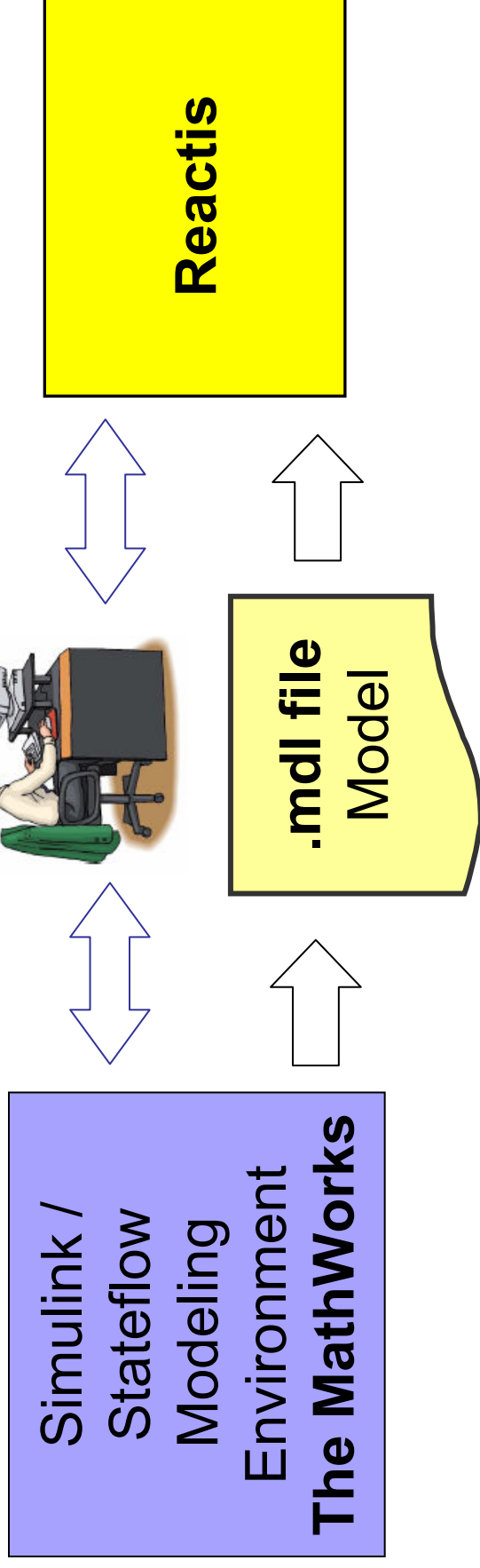
Simulink
Synchronous block-diagram language

©2006 Reactive Systems, Inc

Reactive
Systems, inc

Stateflow

Hierarchical state machines based on Statecharts

Reactive
Systems, inc

C

- C code can be embedded in models
- Version of Reactis to support this is in development

Reactive Systems, inc

Reactis Tester
Automatic test generation from models

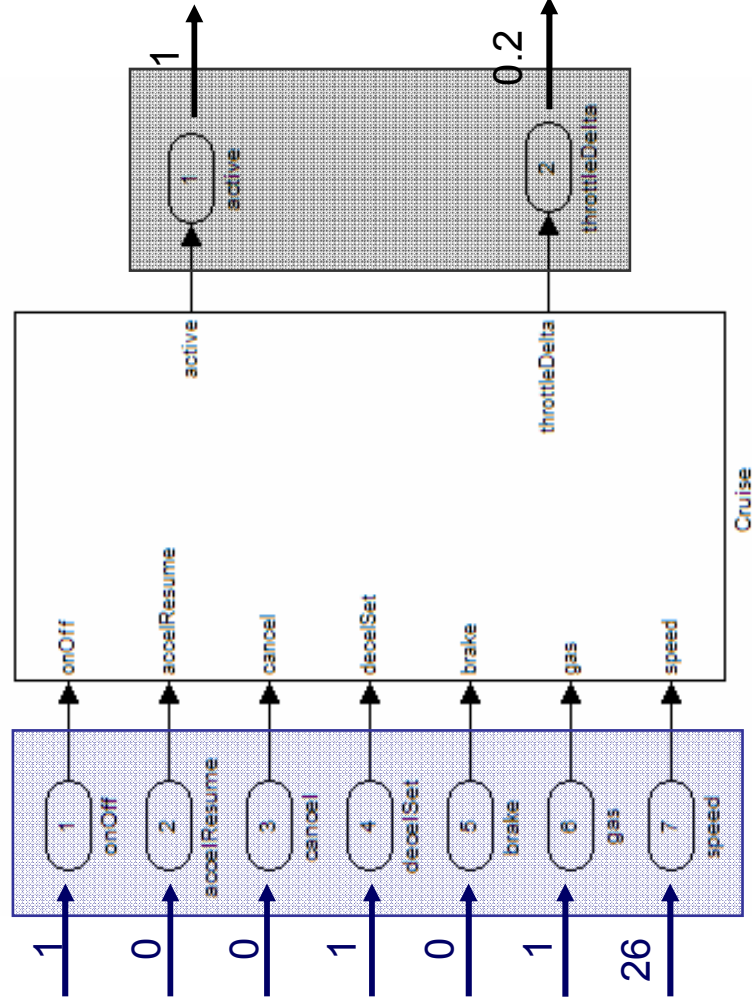Simulink/Stateflow Model → Reactis Tester → Test Suite
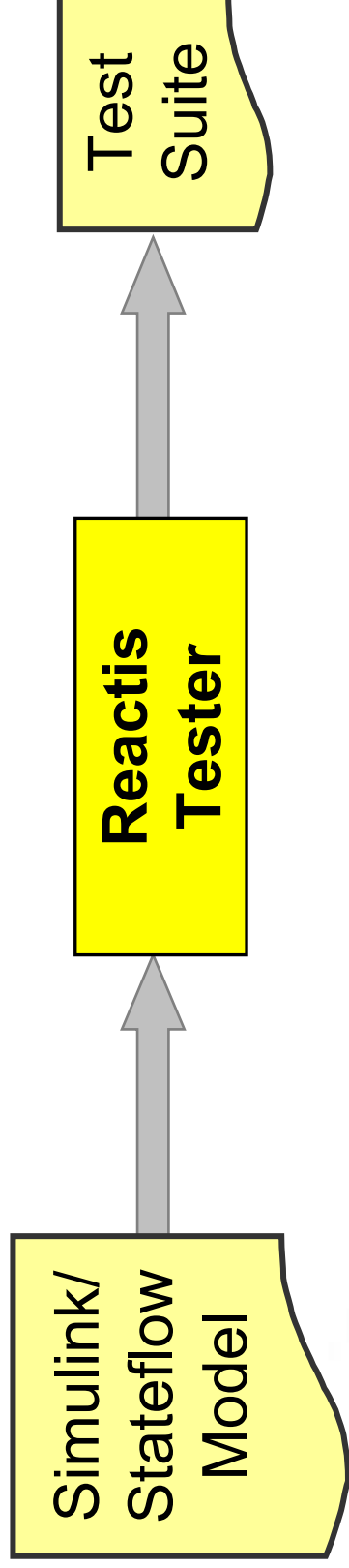
Goals
- Maximize coverage
- Avoid redundancy
- Store model-generated outputs

Uses
- Source code testing
- Model debugging
- Model comparison

Reactive Systems, Inc

Reactis Tests

Simulink/Stateflow Model

Test

Step		1	2	3
Inputs	onOff	1	1	0
	accelResume	1	0	1
	Cancel	0	0	0
	decelSet	0	1	0
	Brake	0	0	0
	gas	1	1	0
	Speed	25	26	27
Outputs	active	1	1	0
	throttleDelta	0.1	0.2	0.0

Cruise

onOff
accelResume
cancel
decelSet
brake
gas
speed

active   1
throttleDelta   0.2

1
0
0
1
0
1
26

Reactive Systems, inc

Reactis Simulator: *Model Execution Environment*

Read input values from test, generate them randomly, or query the user.

See current values of Stateflow variables, Simulink blocks/signals

Take forward or reverse Simulation steps

Set breakpoint to halt simulation when entering state

Visualize unexercised parts of model

Dynamically open scopes To view values of Stateflow variables, Simulink blocks/signals

Reactis: cruise.mdl - cruise.rsi - cruise.rst

File   Edit   View   Simulate   Test Suite   Validate   Coverage   Window   Help

#6 (67): Test 6

67

cruise.CruiseMain.CruiseMDL...

cruise
SL CruiseMain
SL CruiseMDL
SF Mode
DesiredSpeed
Off
Active
Inactive
Init
ThrottleCtl
DeactivateActivate
SpdCheck
Plant
LowSpeedOn
Speed

Input: onOff (double) = 1.0 (double)

[onOff == 1]

[set == 1 && ...   Init/   mode = 2

Add To Watched
Open Scope
Open Distribution Scope

Active,
entry : mode

[activate == 1]   [deactivate == 1]

Inactive/
entry : mode = 3

t=25.0

Test #6, Step 26/67

Reactive Systems, inc

Reactis Validator

Requirement-Checking on Models

Simulink/ Stateflow Model → **Reactis Validator** → Test Suite

Requirements → **Reactis Validator**

Automatic search for violations of requirements
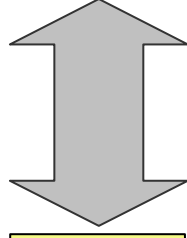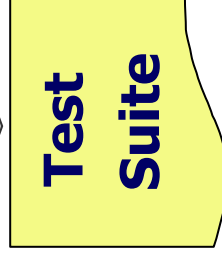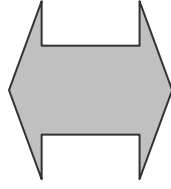
Tests leading to violations
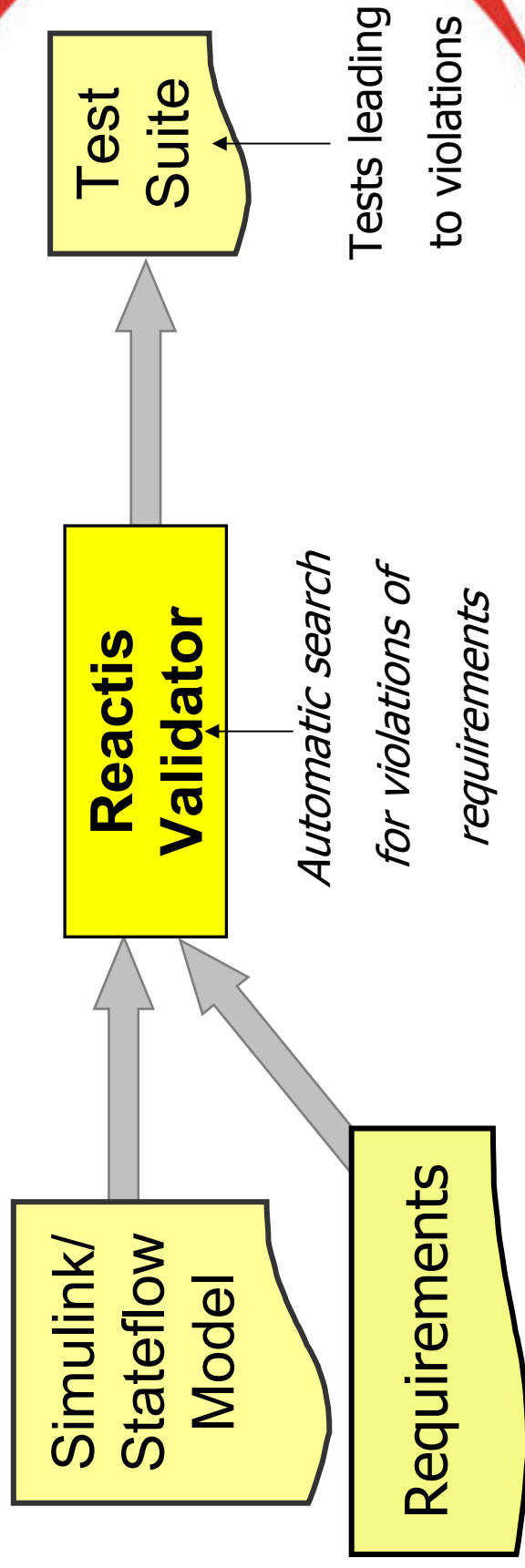
Requirements specified as:

– C-like assertions

– Simulink/Stateflow observer diagrams

Reactis is Testing-Based Solution

Reactis Tester
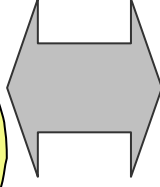Generate test suites from models

Reactis Validator
Generate tests from requirements

Test Suite

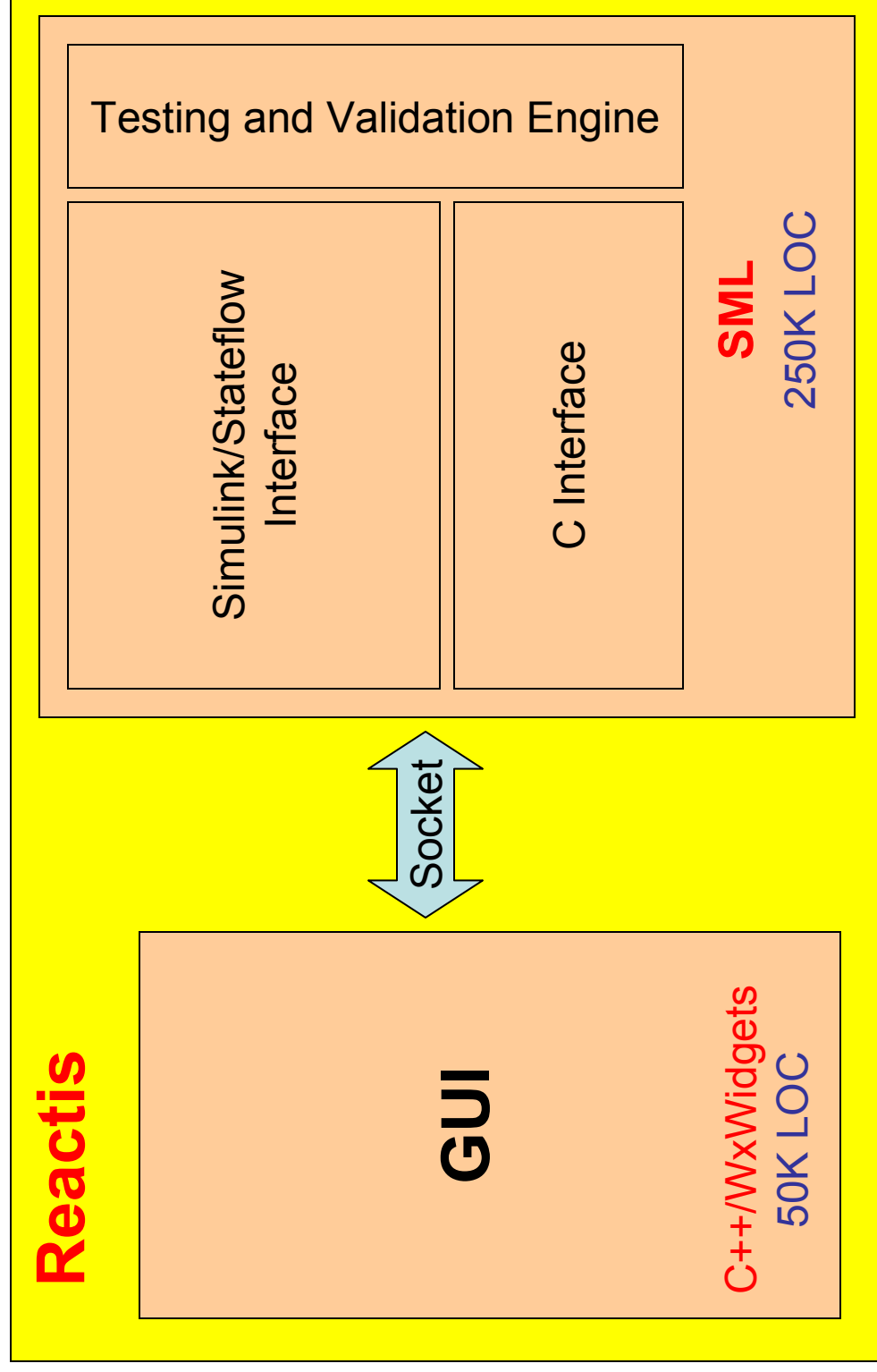**Reactis Simulator**
Run, fine-tune tests

©2006 Reactive Systems, Inc

Reactive Systems, inc

Outline

1. About Reactive Systems

2. Overview of Reactis

3. **Build Reactis**

   Using Standard ML (SML) to:

4. Test Reactis

5. Build the RSI Intranet

6. Build the RSI website

7. Intangible benefits of SML use

Reactive Systems, inc

Reactis Architecture

Reactis

Testing and Validation Engine

Simulink/Stateflow Interface

C Interface

SML
250K LOC

Socket

GUI

C++/WxWidgets
50K LOC

©2006 Reactive Systems, Inc

Reactive Systems, inc

Reactis Architecture

SML especially well-suited for ...

...language processing

...state-space exploration algorithms

Key Technology: *Guided Simulation*

To generate test suite
- execute model
- record inputs/outputs
- at each step select inputs to exercise new parts of model

Testing and Validation Engine

Reactis Backend

Simulink/Stateflow Interface

C Interface

SML

250K LOC

Interpreter for Simulink/ Stateflow models

Interpreter for C programs

17

©2006 Reactive Systems, Inc

Reactive Systems, inc

Key Benefits of SML

- Strong type checking catches many errors at compile time.

- Garbage collection removes major burden from programmer.

- Recursive data types and pattern matching especially useful for building interpreters (for Simulink/Stateflow, C).

- Modules system helps organize big programs.

- Exceptions. Almost all bug reports from customers come with some information.

- Functions as first class values.

- Imperative features available when needed.
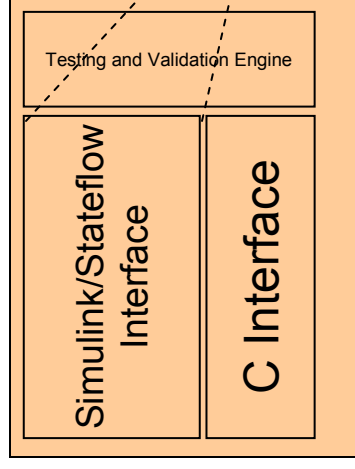
Reactive Systems, inc

Productivity

- 250K lines of SML
- To date approximately 10 man-years of programmer time invested
- 25K SML lines per man-year
- Assume 1 line SML = 2-5 lines of C
- 50K-125K lines of C code per man-year

**Reactive
Systems, inc**

Productivity: SML vs C/C++

The MathWorks, Inc.

- Builds Simulink/Stateflow modeling environment, including simulation engine similar to Reactis Simulink/Stateflow interface

- 1400 employees

- Dozens of programmers have worked on Simulink/Stateflow for 10 years or more

Testing and Validation Engine

Simulink/Stateflow Interface	C Interface

- Implementation language: C/C++

- Conservative estimate would be 50-100 man-year investment

Reactive Systems, Inc.

- Implementation language: SML

- 10 man-year investment

Suggests factor 5-10 productivity improvement from SML!

©2006 Reactive Systems, Inc

Reactive Systems, inc

Outline

1. About Reactive Systems

2. Overview of Reactis

3. Build Reactis

Using Standard ML (SML) to:

4. **Test Reactis**

5. Build the RSI Intranet

6. Build the RSI website

7. Intangible benefits of SML use

Reactive Systems, inc

Unit Testing

- For each SML structure, aim to have unit tests implemented to test functionality.

- Approach:
  - define SML function that implements test
  - create list of test cases with expected results
  - apply test to each element of list, log fails

©2006 Reactive Systems, Inc

Reactive Systems, inc

Unit Testing Example

$2^3 = 8$

$3^2 = 9$

...

```
val powTestL = [ (2,3,8), (3,2,9), (4,2,16), (2,0,1)]

fun powTest(m,n,expected) =
  let val found = RsiLib.intPow(m,n)
  in
```

Function under test

```
    if found = expected
    then T.pass()
    else T.fail("pow "^(Int.toString m)^
                " "^(Int.toString n),
                SOME (Int.toString expected),
                SOME (Int.toString found))

  end

...
fun runTests() = app powTest powTestL
```

Reactive Systems, inc

Integration Testing

Key check: *does Reactis conform to MathWorks simulation engine?*

Approach:

- Maintain test bed of models

  Implemented by 3K line SML program which runs nightly.

  - For each model:
    - generate test suite in Reactis
    - run tests in MathWorks
    - compare results

Reactive Systems, inc

Outline

1. About Reactive Systems
2. Overview of Reactis
3. Build Reactis
4. Test Reactis
5. **Build the RSI Intranet**
6. Build the RSI website
7. Intangible benefits of SML use

Using Standard ML (SML) to:

Reactive Systems, inc

RSI Intranet

Web-based application automating business activities

Reactive Systems, inc

RSI Intranet

Steve Sims logged in.
Log out | Intranet Home

Accounting

Time Cards
Daily Effort Report
Bimonthly Effort Report
Effort Summary
Correct Error

Accounting
Standard Expense Report
View Reports

Miscellaneous
Security Policy
Change your password
Board Information
RSI Intranet Usage Statistics
RSI Public Web Site Usage Statistics
RSI CVS Repository
Database Tables

Communications
RSI Mail Server
Generate Fax Cover Sheet
Sending a Fax Over the Web
Generate Mailing Labels
Generate Form Letters

Sales
Price List
Generate Quote
View Quotes
Generate Invoice
View Invoices
Packing Slips
View Licenses
View Support Contracts
Generate Demo License File
Generate Paid License File
View License Files

Reactis
Documentation
Wish List

Quoting, Invoicing, License management

Contact Management

People:
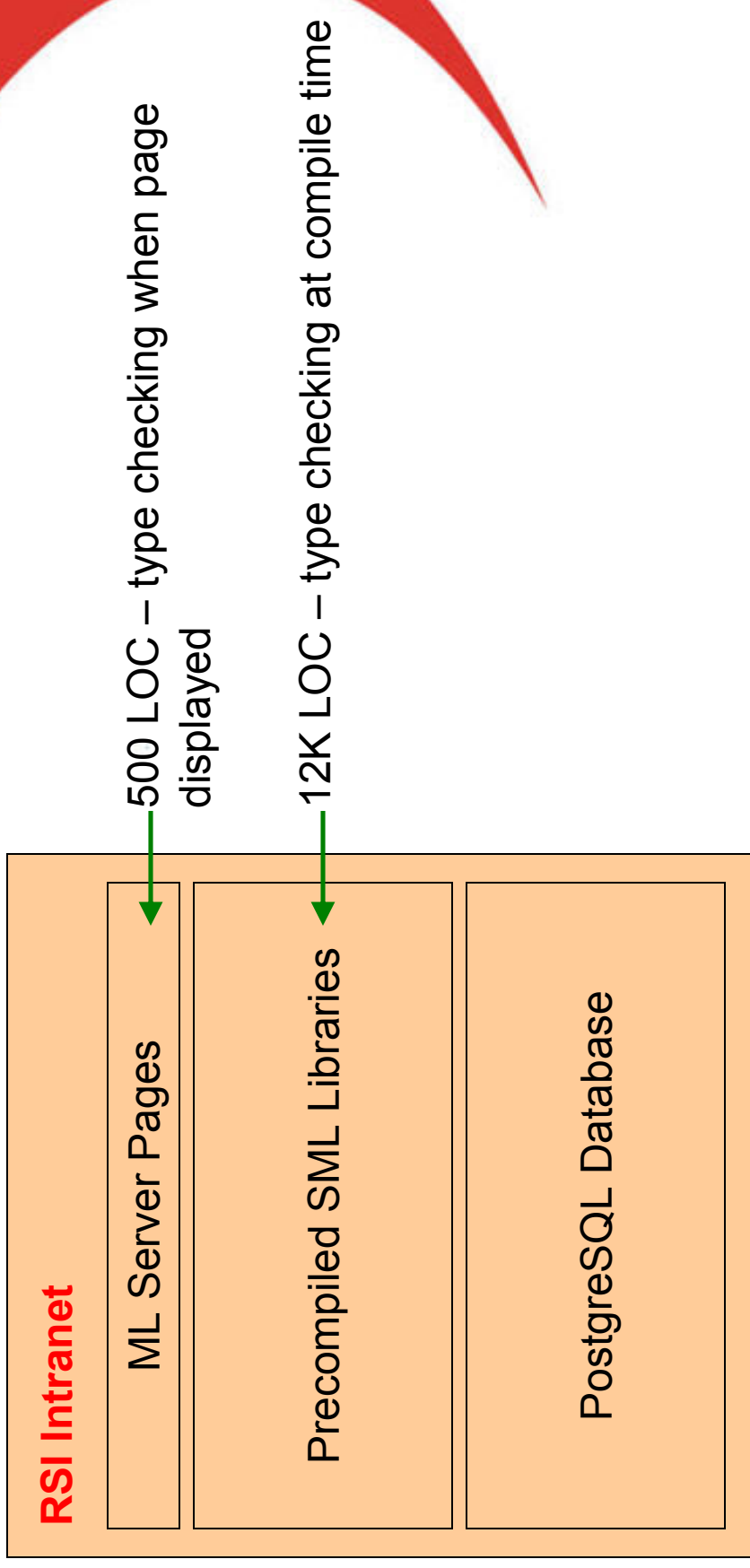A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

Leads: Warm, Hot | Evaluators: Warm, Hot | Customers | Active Customers

Employees | Partners | Investors | Suppliers/Service Providers

Track leads, customers

©2006 Reactive Systems, Inc

Reactive Systems, inc

RSI Intranet Architecture

500 LOC – type checking when page displayed

12K LOC – type checking at compile time

RSI Intranet

ML Server Pages

Precompiled SML Libraries

PostgreSQL Database

Reactive Systems, inc

Outline

1. About Reactive Systems

2. Overview of Reactis

Using Standard ML (SML) to:

3. Build Reactis

4. Test Reactis

5. Build the RSI Intranet

6. **Build the RSI website**

7. Intangible benefits of SML use

©2006 Reactive Systems, Inc

Reactive Systems, inc

RSI Website

Reactive Systems, inc

Login | Site Map

Google™ [Search]

Reactis®
Model-Based
Testing and Validation

Home | About Us | Contact | Products | Services | News

Model-Based
Testing and Validation
with
Reactis®

Quick Tour

Test and validate Simulink®
and Stateflow® models
automatically. Obtain
machine-generated test data
from models for system testing.
Save time and money on
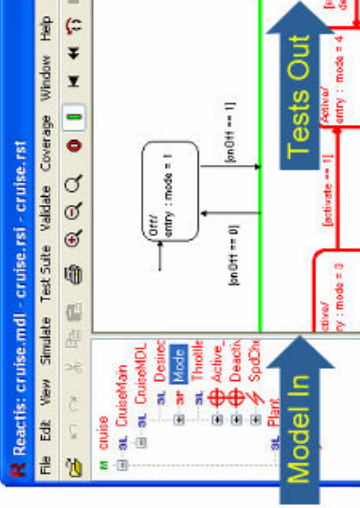verifying embedded controllers.
To learn more, take the tour...

- What Reactis Does
- Using Reactis
- Generating Tests
- Validating Models
- Enhancing Tests
- Try Reactis!

Reduce Testing Costs with Reactis...

- Quick Tour | FAQs
- Extended Tour: Tester, Simulator, Validator
- User's Guide: PDF | HTML
- Download Trial
- Latest Updates: Reactis V2006.1.5

News
- Reactive Systems to Exhibit at Aerospace and Defense Conference
- Reactive Systems to Exhibit at International Automotive Conference
- Reactis V2006 Now Available
- More News...

White Papers
- Model-Based Testing and Validation of Control Software with Reactis

Partners
- The MathWorks
- Europe: IT Power Consultants
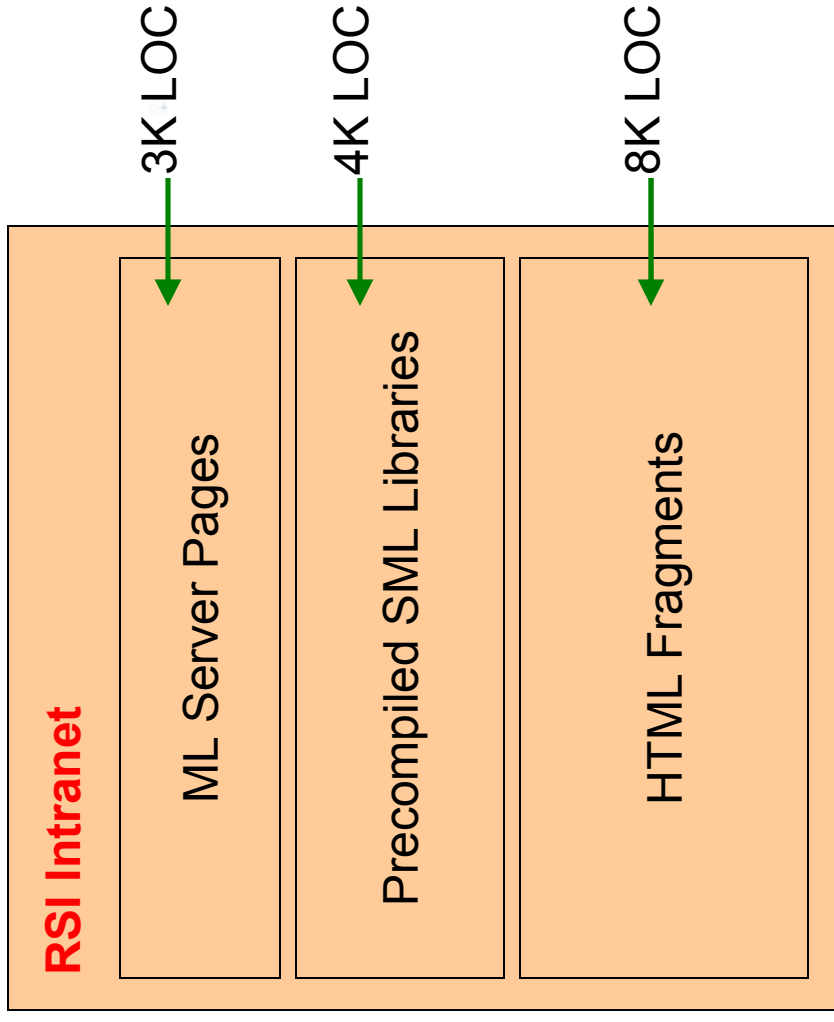- Japan: Cybernet Systems

Newsletters
- Reactis Communicator
- Reactis Patches

Tomorrow's Software Today®

Reactis: cruise.mdl - cruise.rsi - cruise.rst
File Edit View Simulate Test Suite Validate Coverage Window Help

Model In

Tests Out

RSI Website

RSI Intranet

- Same SML function generates each page to give uniform look and feel (same headers, footers, …)

- When adding content, need not worry with page layout, just add html fragment.

Component	LOC
ML Server Pages	3K LOC
Precompiled SML Libraries	4K LOC
HTML Fragments	8K LOC

©2006 Reactive Systems, Inc

**Reactive
Systems, inc**

RSI Website

Common design pattern

SML list	val navBarL = [("/index.msp","Home"), ("/about.msp","About Us"), …]
Recursive function	fun mkNavBar(l:(string * string) list) = …

Navigation bar

List of news items

Reactive Systems, inc

Login | Site Map
Search

Google™

Reactis®
Model-Based
Testing and Validation

Home | About Us | Contact | Products | Services | News

News Room

Jun 09, 2006 Reactive Systems to Exhibit at Aerospace and Defense Conference
May 03, 2006 Reactive Systems to Exhibit at International Automotive Conference
Jan 25, 2006 Reactis V2006 Now Available
Oct 20, 2005 Reactive Systems to Exhibit at J-MAAB Open Conference
Jul 22, 2005 Reactive Systems to Exhibit at Control Design Automation Seminar
May 18, 2005 Reactis 2005 Now Available
Apr 11, 2005 IT Power Consultants and Reactive Systems Enter Joint Marketing Agreement
Mar 22, 2005 Reactis Beta Supports Fixed-Point Data Types in Simulink

Reactive
Systems, inc

Outline

1. About Reactive Systems
2. Overview of Reactis

Using Standard ML (SML) to:

3. Build Reactis
4. Test Reactis
5. Build the RSI Intranet
6. Build the RSI website
7. Intangible benefits of SML

Reactive Systems, inc
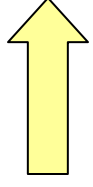
Intangible Benefits of SML

Compiler does dirty work: type inference, memory management, missing cases when pattern matching, …
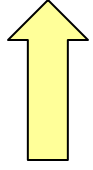
…programmer focuses on more interesting aspects of problem

Programming in SML is fun!

Key factor in attracting and retaining highly productive employees.

- No programmer has left RSI, all have had other offers, all state that programming in SML is factor in staying.

- Fun work makes productive people even more productive

Reactive Systems, inc

Thank you!

**Reactive
Systems, inc**