
# *Modelling Security in UML/OCL for C2IS*

Robert Charpentier & Martin Salois

R et D pour la défense Canada
Defence R&D Canada
Canada

# Plan

- Motivation & Objectives
- Software Certification Techniques
- Modelling Security in UML and OCL
- Summary and Perspectives

# Motivations for High-Confidence Software

- Increasing numbers of critical infrastructures:
  - Banking, medical instruments, emergency services, power distribution, telecommunications, transportation, government archives, etc
- Real-time and embedded systems:
  - Pacemakers, power plants, avionics, cellular phones, etc
- *Shielding C2IS against attacks and subversive exploits*

# Impact of Unreliable Software

*Inadequate software testing is currently estimated to cost the US* ***between 22 and 60 billion dollars***

- *± 50 % due to development and design flaws*
- *± 50 % due to user behaviour*

Ref : - *The economic Impacts of Inadequate Infrastructure for Software Testing* RTI Planning Report 02-3 for NIST and DOC , May 2002
- Standish Group estimate is 200 billion dollars

# Quantitative Assessment

- 45 e-business applications analyzed
  - *10 types of defects studied*
  - *Business Impact (1-5) x Risk of Exploit (1-5)*

    *= Business Risk (1-25)*
  - *Best Quartile scored* ***4.8 / 25***
  - *Worst Quartile scored* ***23.0 / 25***
- 70 % of defects were associated with design flaws
- 47 % of defects were exploitable

Ref : Jaquith A, 'The security of Applications; Not All Are created Equal', Research Report @stake, February 2002 www.atstake.com/research

# Impact of Unreliable C2IS

- Failure of an important mission
- Significant loss or damage to property
- Serious environmental damage
- Injury or illness
- Loss of life

Ref : Rodrique J. P., *'Market Study – Critical Software Certification',* GeoAlliance International, May 2003

# Long –Term Vision

Craig Mundie (Microsoft) has bet Eric Schmidt (Google) :

*by 2030, passengers will routinely board commercial airline flights without a pilot…*

*… flights will be flown entirely by computers !*

Ref : Wired Magazine – May 2002

# High-Confidence Software -- *R&D*

- Reliability and Security enforcement tools
  - *MaliCOTS (1997-2001)*
  - *SOCLe (2002-2005)*
- Modelling Security
  - *PoliSEC (~ 2003-2006)*
  - *CompoSEC (~ 2004-2007)*

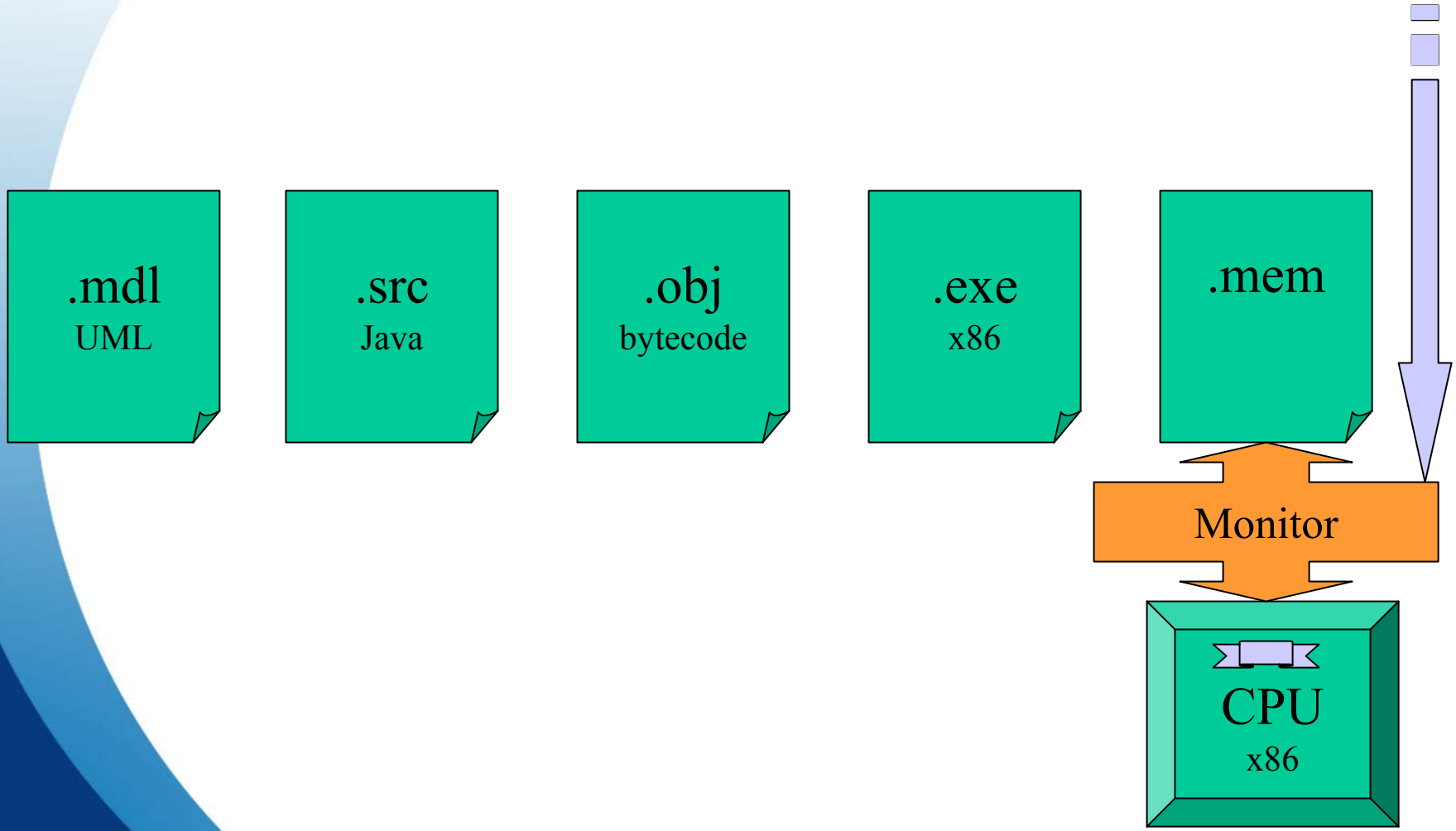
# Software Attacks: *Threat and Consequences*

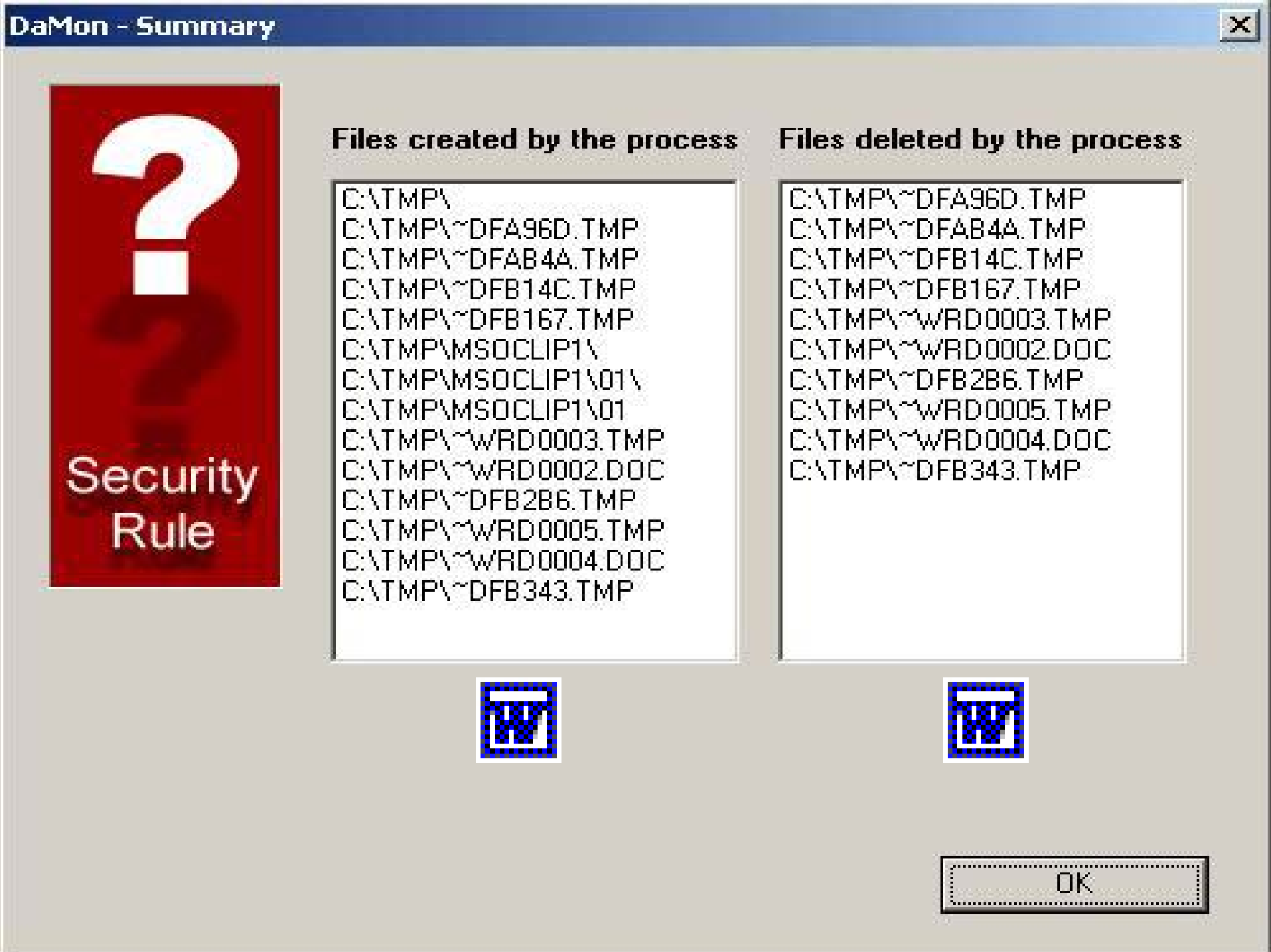
1. Survey of exploitations
2. Hacker community
3. Countries and other organisations
4. Consequences of software vulnerabilities
5. Conclusion
6. Bibliography of 121 references
7. Insider threat assessment
8. Cracking tools survey, and more

Ref : Salois M. *« Software Attacks: Threat and Consequences»,* DREV ECR 2002-150, presented at the WP-11 Quadripartite Annual Conference on Information Technologies, Fall 2002, Unclassified or Classified versions

# Reliability & Security Policy

Wide Spectrum ⟷ Narrow Spectrum



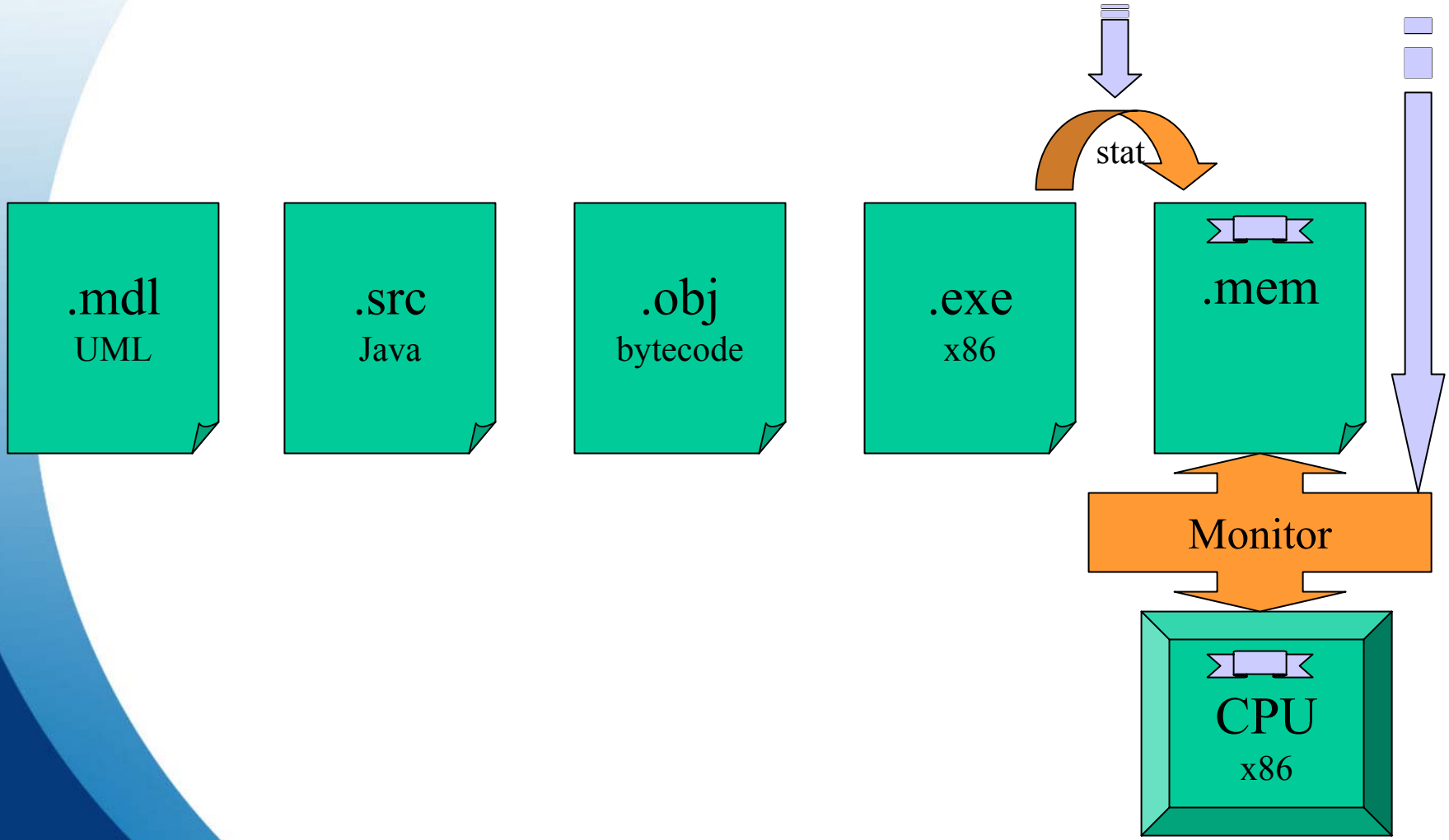

DaMon - Summary
Security Rule
Files created by the process
C:\TMP\
C:\TMP\~DFA96D.TMP
C:\TMP\~DFAB4A.TMP
C:\TMP\~DFB14C.TMP
C:\TMP\~DFB167.TMP
C:\TMP\MSOCLIP1\
C:\TMP\MSOCLIP1\01\
C:\TMP\MSOCLIP1\01
C:\TMP\~WRD0003.TMP
C:\TMP\~WRD0002.DOC
C:\TMP\~DFB2B6.TMP
C:\TMP\~WRD0005.TMP
C:\TMP\~WRD0004.DOC
C:\TMP\~DFB343.TMP
Files deleted by the process
C:\TMP\~DFA96D.TMP
C:\TMP\~DFAB4A.TMP
C:\TMP\~DFB14C.TMP
C:\TMP\~DFB167.TMP
C:\TMP\~WRD0003.TMP
C:\TMP\~WRD0002.DOC
C:\TMP\~DFB2B6.TMP
C:\TMP\~WRD0005.TMP
C:\TMP\~WRD0004.DOC
C:\TMP\~DFB343.TMP
OK

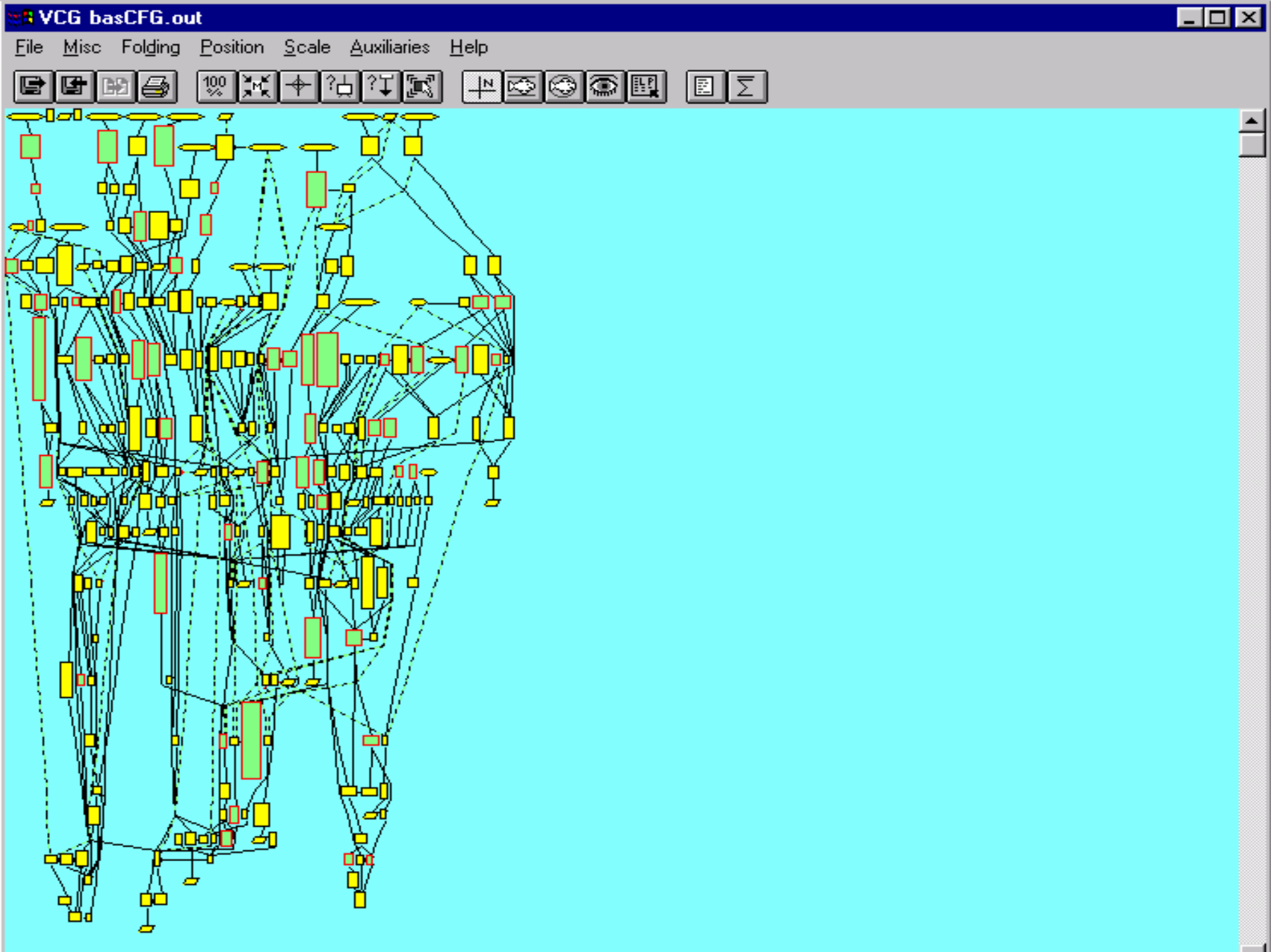
# Monitoring at Runtime

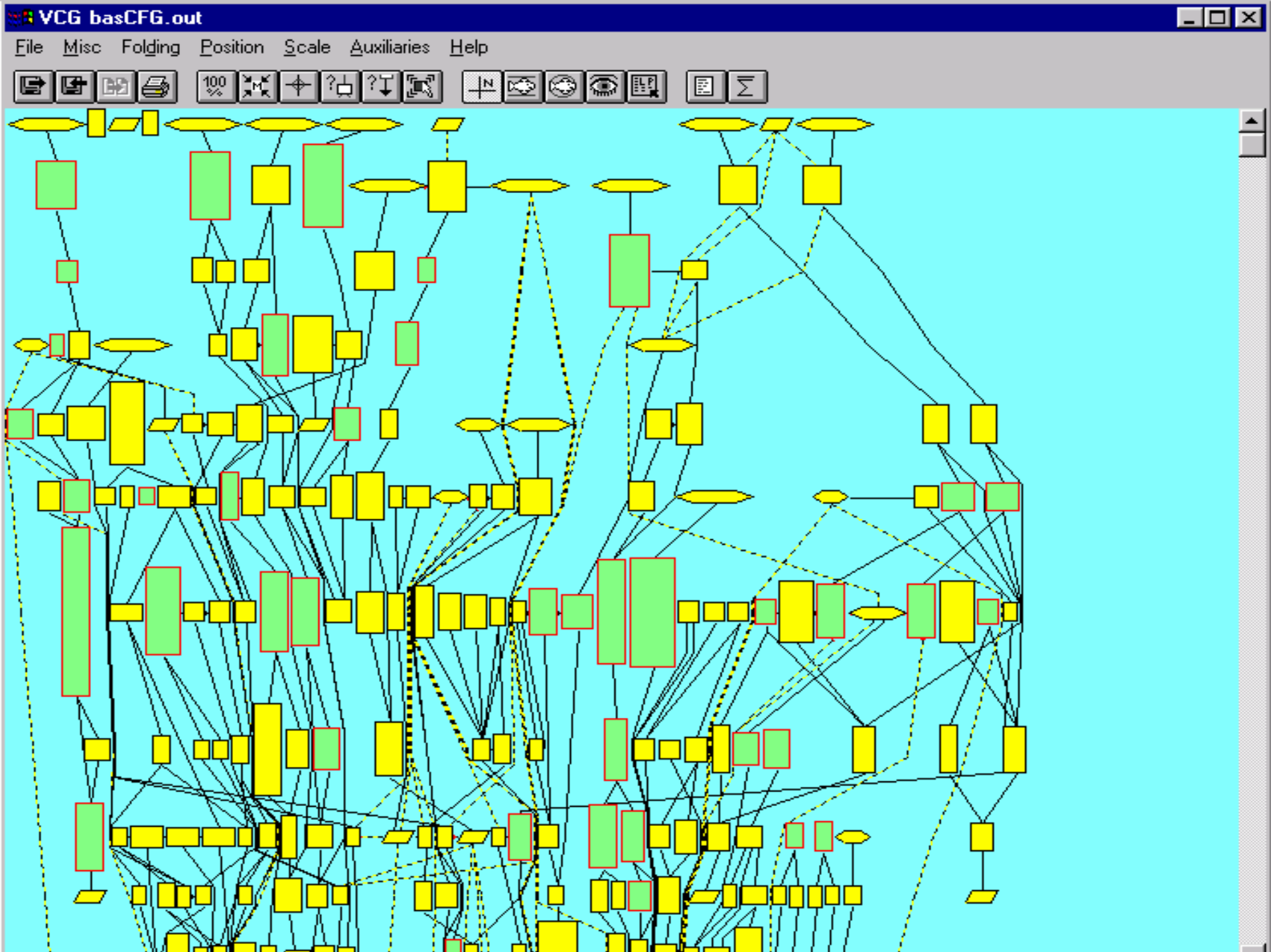
- PROS :
  - Exploits the knowledge that can be gained by running the program
  - Best technique for user surveillance
  - Acceptable for software vendors
- CONS :
  - Significant overhead in run-time performance
  - “Infinite number” of possibilities and conditions
  - Too much information to manage

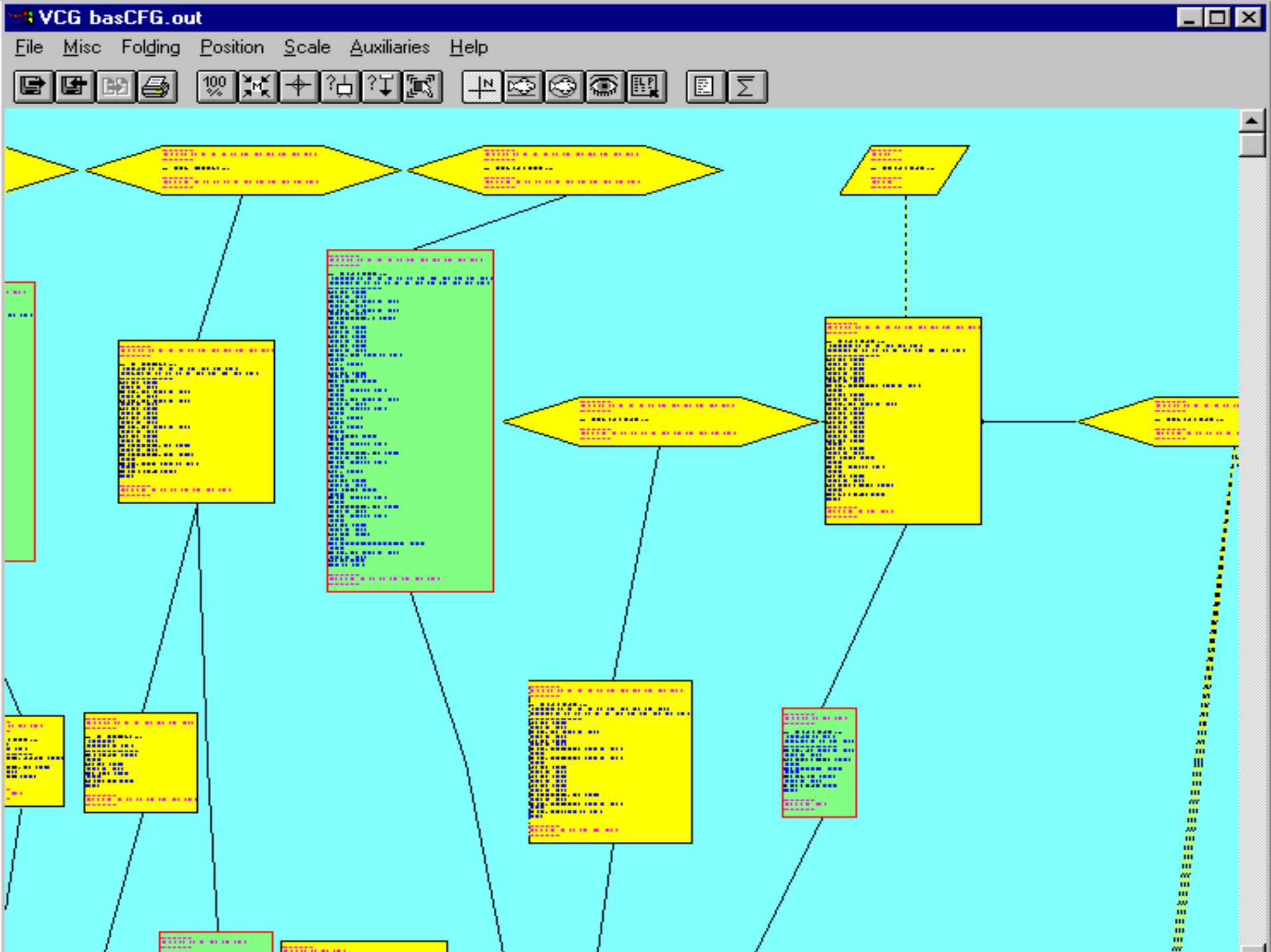
# Reliability & Security Policy

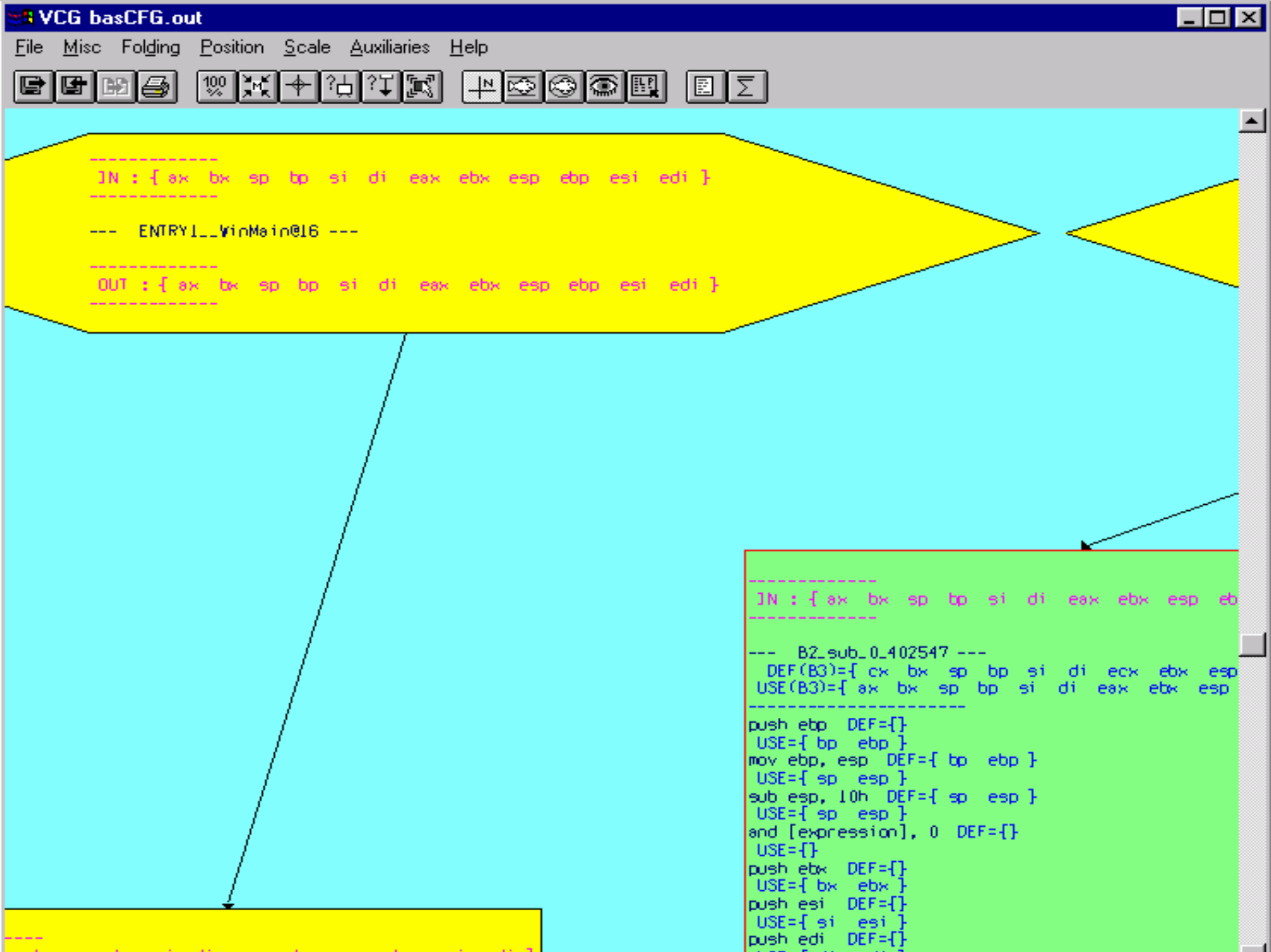
Wide Spectrum ⟷ Narrow Spectrum

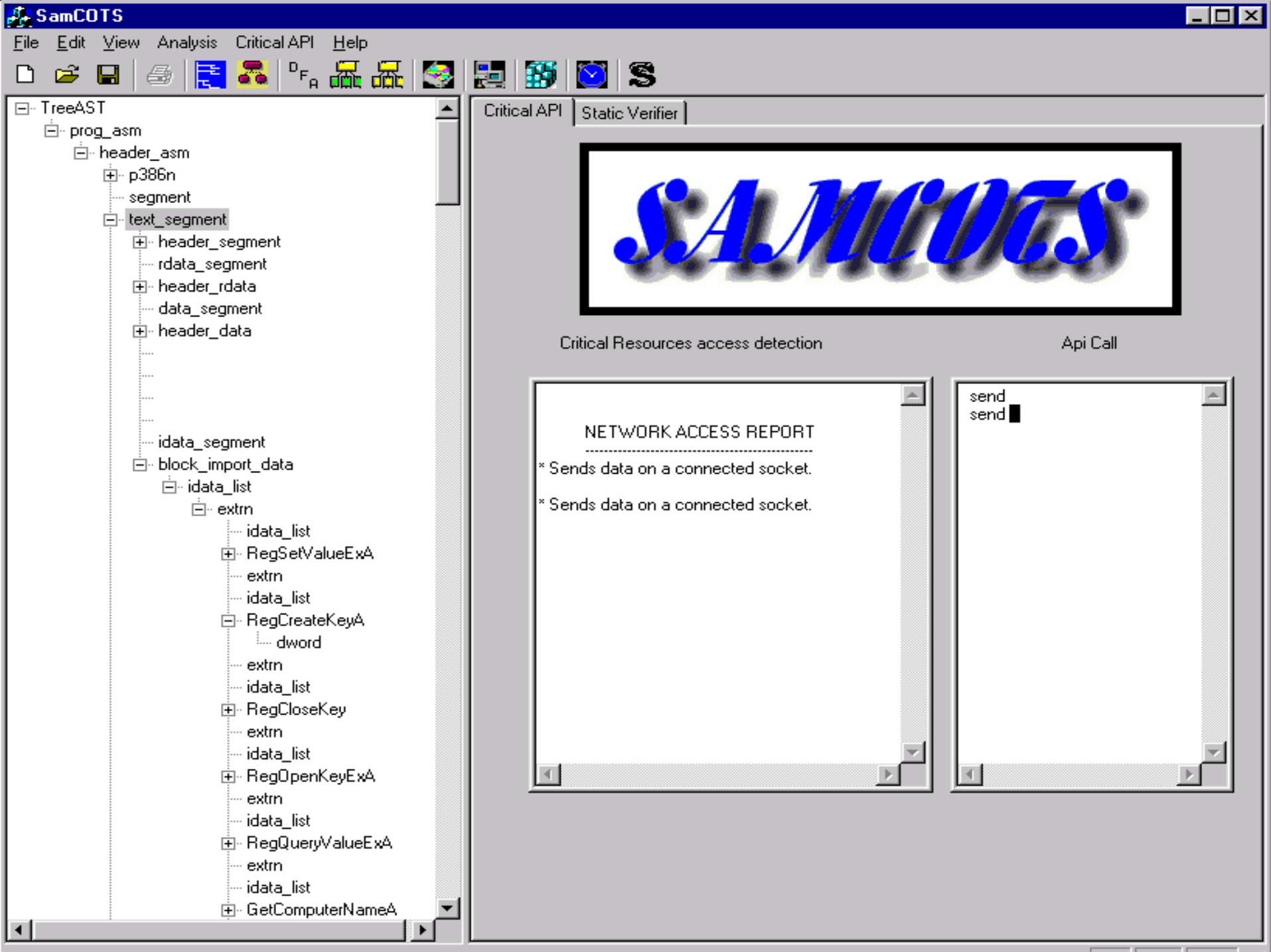


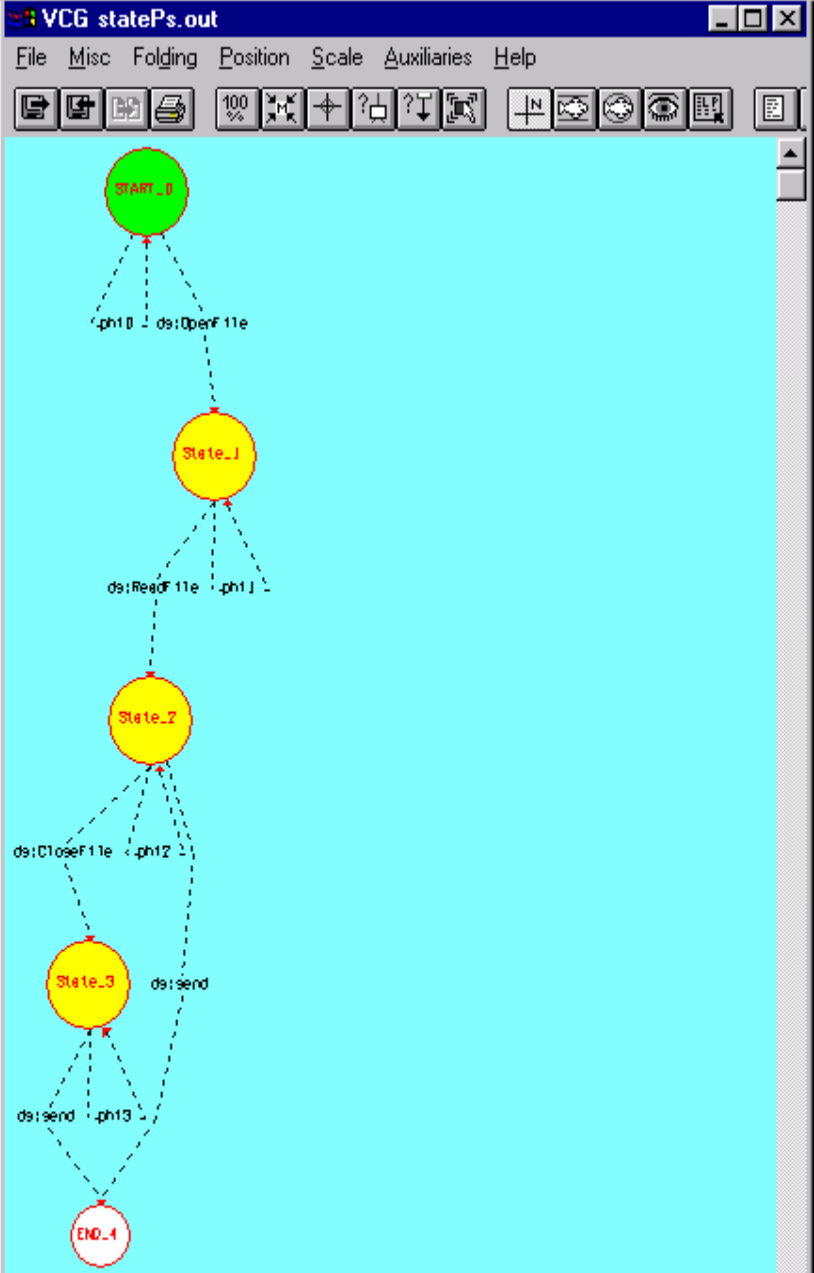

VCG basCFG.out
File
Misc
Folding
Position
Scale
Auxiliaries
Help


VCG basCFG.out
File
Misc
Folding
Position
Scale
Auxiliaries
Help


VCG basCFG.out
File
Misc
Folding
Position
Scale
Auxiliaries
Help


VCG basCFG.out
File Misc Folding Position Scale Auxiliaries Help
IN : { ax bx sp bp si di eax ebx esp ebp esi edi }
--- ENTRY1__WinMain@16 ---
OUT : { ax bx sp bp si di eax ebx esp ebp esi edi }
IN : { ax bx sp bp si di eax ebx esp eb
--- B2_sub_0_402547 ---
DEF(B3)={ cx bx sp bp si di ecx ebx esp
USE(B3)={ ax bx sp bp si di eax ebx esp
push ebp DEF={}
USE={ bp ebp }
mov ebp, esp DEF={ bp ebp }
USE={ sp esp }
sub esp, 10h DEF={ sp esp }
USE={ sp esp }
and [expression], 0 DEF={}
USE={}
push ebx DEF={}
USE={ bx ebx }
push esi DEF={}
USE={ si esi }
push edi DEF={}


SamCOTS
File Edit View Analysis Critical API Help
TreeAST
prog_asm
header_asm
p386n
segment
text_segment
header_segment
rdata_segment
header_rdata
data_segment
header_data
idata_segment
block_import_data
idata_list
extrn
idata_list
RegSetValueExA
extrn
idata_list
RegCreateKeyA
dword
extrn
idata_list
RegCloseKey
extrn
idata_list
RegOpenKeyExA
extrn
idata_list
RegQueryValueExA
extrn
idata_list
GetComputerNameA
Critical API
Static Verifier
SAMCOTS
Critical Resources access detection
Api Call
NETWORK ACCESS REPORT
* Sends data on a connected socket.
* Sends data on a connected socket.
send
send




Critical API | Static Verifier

SAMOTS

Resources checking

- (•) Disk-Network Access
- ( ) Disk Access
- ( ) Registry-Network Access
- ( ) Registry Access

VERIFICATION: Send information from the Disk on the net DISK_NET

Static verifier

Security Automate

# Static Analysis

- PROS:
  - Ideal pre-filter for the monitoring
  - Analysis of program behaviours over all possible execution paths
  - Analyze once, execute everywhere
- CONS:
  - Undecidability of many interesting properties
  - Hard on binary executables
  - Illegal on most COTS software

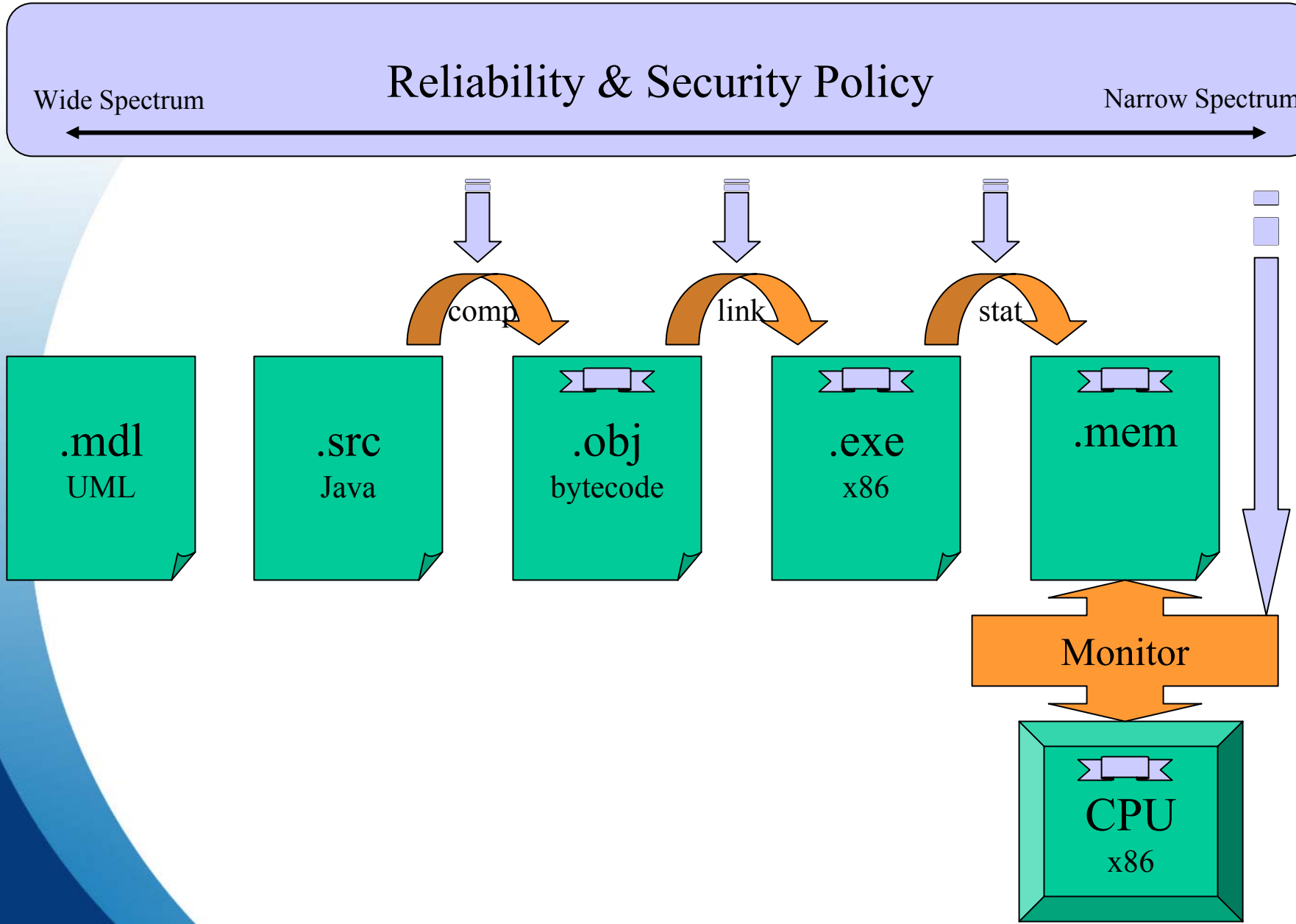
# Increasing Complexity

| | | Lines of code |
|---|---|---|
| • Win 3.1 | 1992 | *3 million* |
| • Win NT | 1992 | *4 million* |
| • Win 95 | 1995 | *15 million* |
| • Win NT 4 | 1996 | *16.5 million* |
| • Win 98 | 1998 | *18 million* |
| • Win 2000 | 2000 | *40- 60 million* |

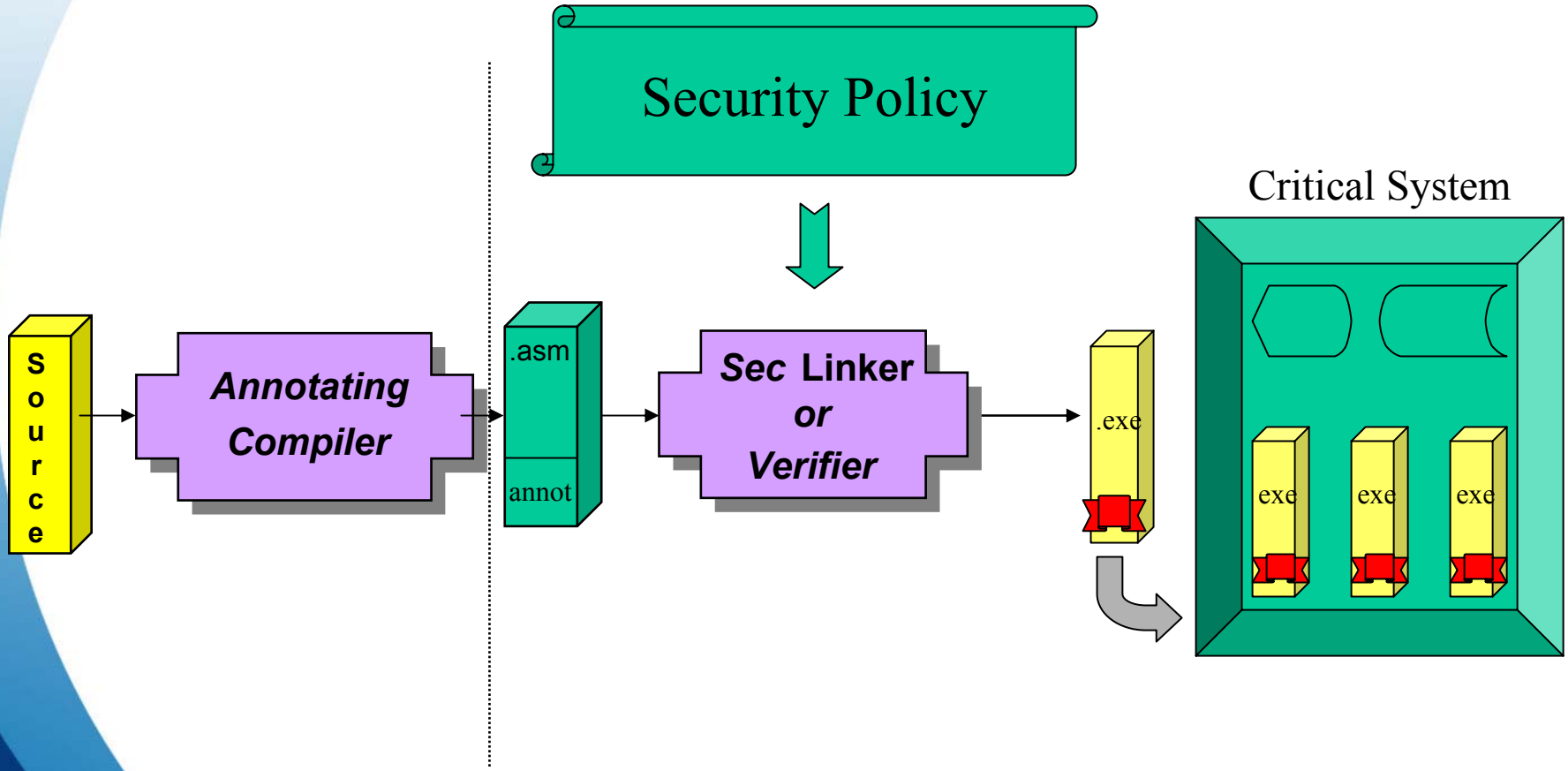
Ref: M. Sues, M. Gingras, *Secure Programming and Development Practices,* Cinnabar Networks, CITSS Symposium , June 2001

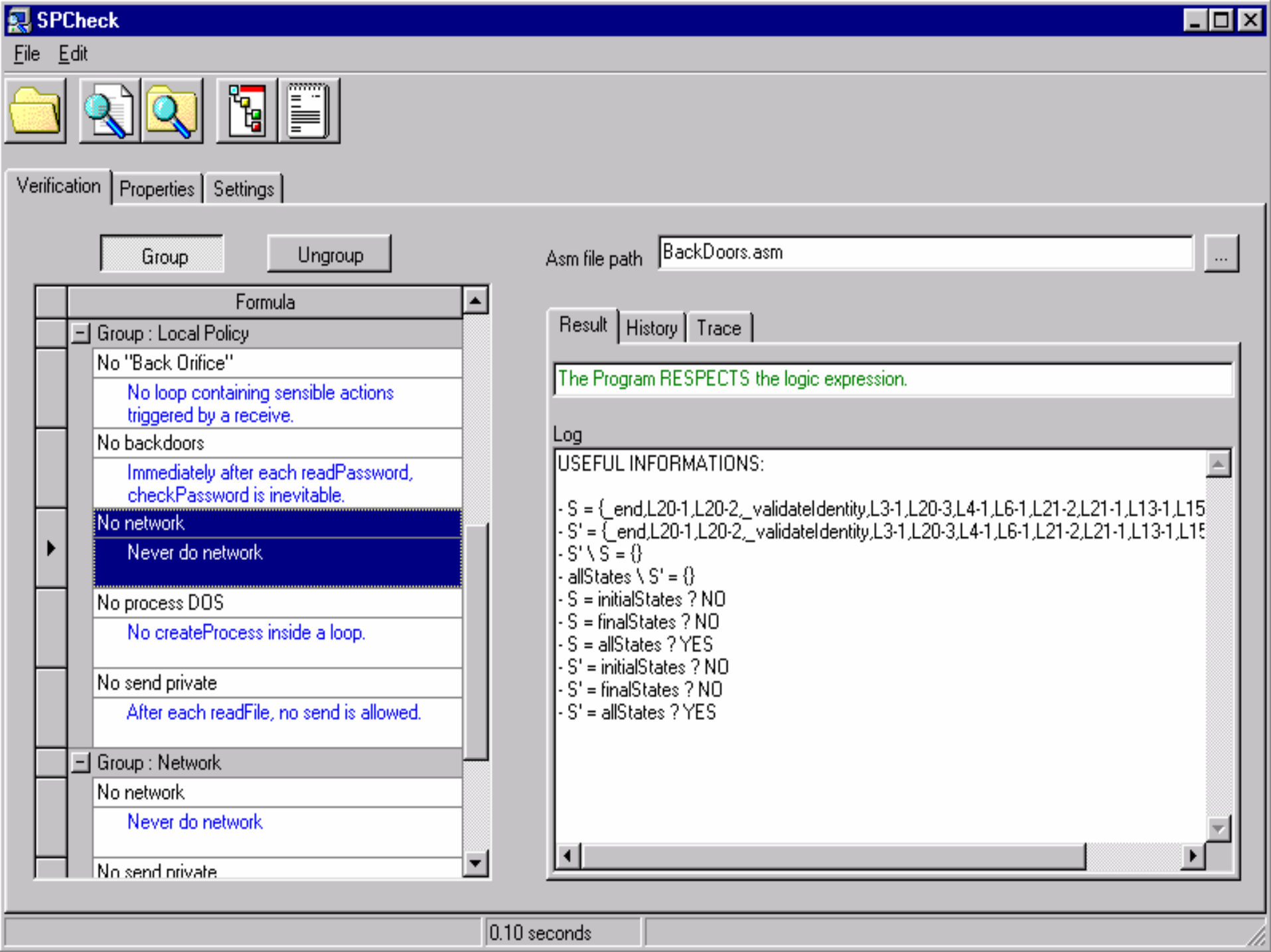
# Software Certification Techniques

- Static analysis of code
- Dynamic monitoring of execution
- Certifying compiler technology


Reliability & Security Policy
Wide Spectrum
Narrow Spectrum
comp
link
stat
.mdl
UML
.src
Java
.obj
bytecode
.exe
x86
.mem
Monitor
CPU
x86

# Two-step Certification




SPCheck
File Edit
Verification Properties Settings
Group Ungroup
Asm file path BackDoors.asm ...
Formula
Group : Local Policy
No "Back Orifice"
No loop containing sensible actions triggered by a receive.
No backdoors
Immediately after each readPassword, checkPassword is inevitable.
No network
Never do network
No process DOS
No createProcess inside a loop.
No send private
After each readFile, no send is allowed.
Group : Network
No network
Never do network
No send private
Result History Trace
The Program RESPECTS the logic expression.
Log
USEFUL INFORMATIONS:
- S = {_end,L20-1,L20-2,_validateIdentity,L3-1,L20-3,L4-1,L6-1,L21-2,L21-1,L13-1,L15
- S' = {_end,L20-1,L20-2,_validateIdentity,L3-1,L20-3,L4-1,L6-1,L21-2,L21-1,L13-1,L15
- S' \ S = {}
- allStates \ S' = {}
- S = initialStates ? NO
- S = finalStates ? NO
- S = allStates ? YES
- S' = initialStates ? NO
- S' = finalStates ? NO
- S' = allStates ? YES
0.10 seconds

SPCheck

File Edit

Verification | Properties | Settings

Group | Ungroup

Asm file path: BackDoors.asm

| Formula |
| --- |
| Group : Local Policy |
| No "Back Orifice" |
| No loop containing sensible actions triggered by a receive. |
| No backdoors |
| Immediately after each readPassword, checkPassword is inevitable. |
| No network |
| Never do network |
| No process DOS |
| No createProcess inside a loop. |
| No send private |
| After each readFile, no send is allowed. |
| Group : Network |
| No network |
| Never do network |
| No send private |

Result | History | Trace

The Program DOES NOT RESPECT the logic expressior

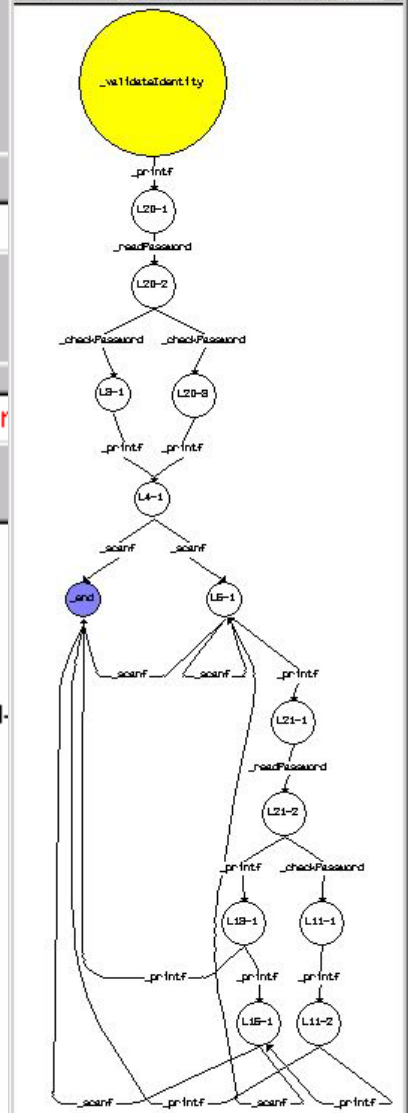
Log

USEFUL INFORMATIONS:

- S = {_end}
- S' = {_end}
- S' \ S = {}
- allStates \ S' = {L20-1,L20-2,_validateIdentity,L3-1,L20-
- S = initialStates ? NO
- S = finalStates ? YES
- S = allStates ? NO
- S' = initialStates ? NO
- S' = finalStates ? YES
- S' = allStates ? NO

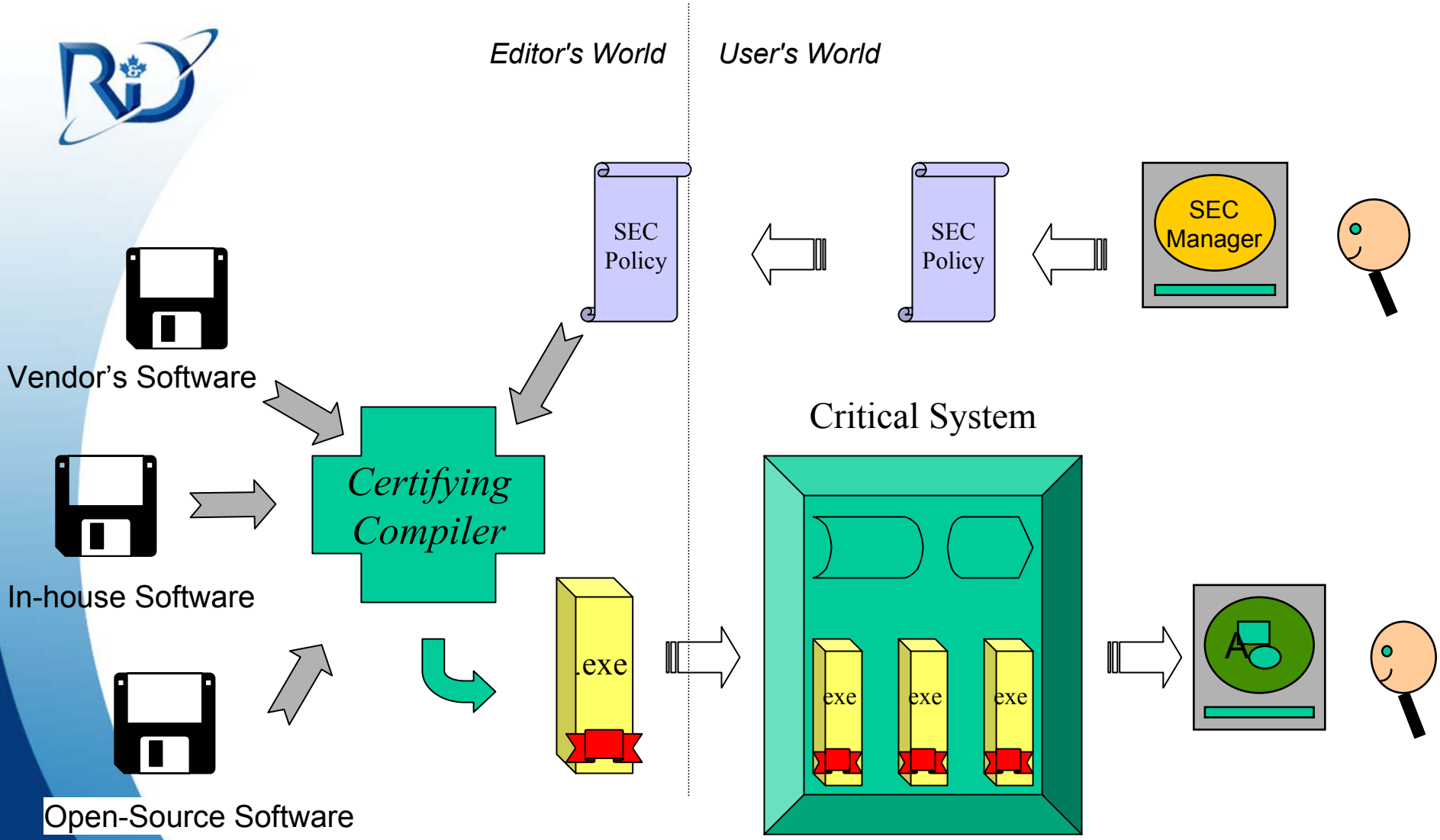
aiSee (ERP version) result.gdl

File Misc Folding Position Scale
Auxiliaries Help




(253,1)

0.70 seconds


Editor's World
User's World
SEC Policy
SEC Policy
SEC Manager
Vendor's Software
In-house Software
Open-Source Software
Certifying Compiler
.exe
Critical System
exe
exe
exe

# Certifying compiler

- PROS:
  - Large COTS certification --- rapidly
  - Detailed and exhaustive enforcement
  - Protect IP
  - Execution not slower
  - Enforcement of  security, maintainability, interoperability (...) specs
- CONS:
  - Emerging technology
  - May need support from the monitor for full enforcement

# Research Outcomes --- *MaliCOTS project*

- Market survey
- MaliCOTS prototypes:
  - *SamCOTS* --- *Static Code Analyser*
  - *DaMon* --- *Runtime Monitor*
  - *TalCC* --- *ANSI C Certifying Compiler*
  - *JACC* --- *Java Certifying Compiler*
  - *SPCheck* --- *Security Policy Checker*
- Lots of publications
  - http://www.drdc-rddc.gc.ca/researchtech/malicots/home_e.asp

# The MaliCOTS Project

## A very successful Project

TechnoFed Gold Medal 2000
Partnership

OCTAS 2001

Octas 2001
Future Scientist

CIPA

CANADIAN INFORMATION PRODUCTIVITY AWARDS

CIPA 2001 Institutions Awards

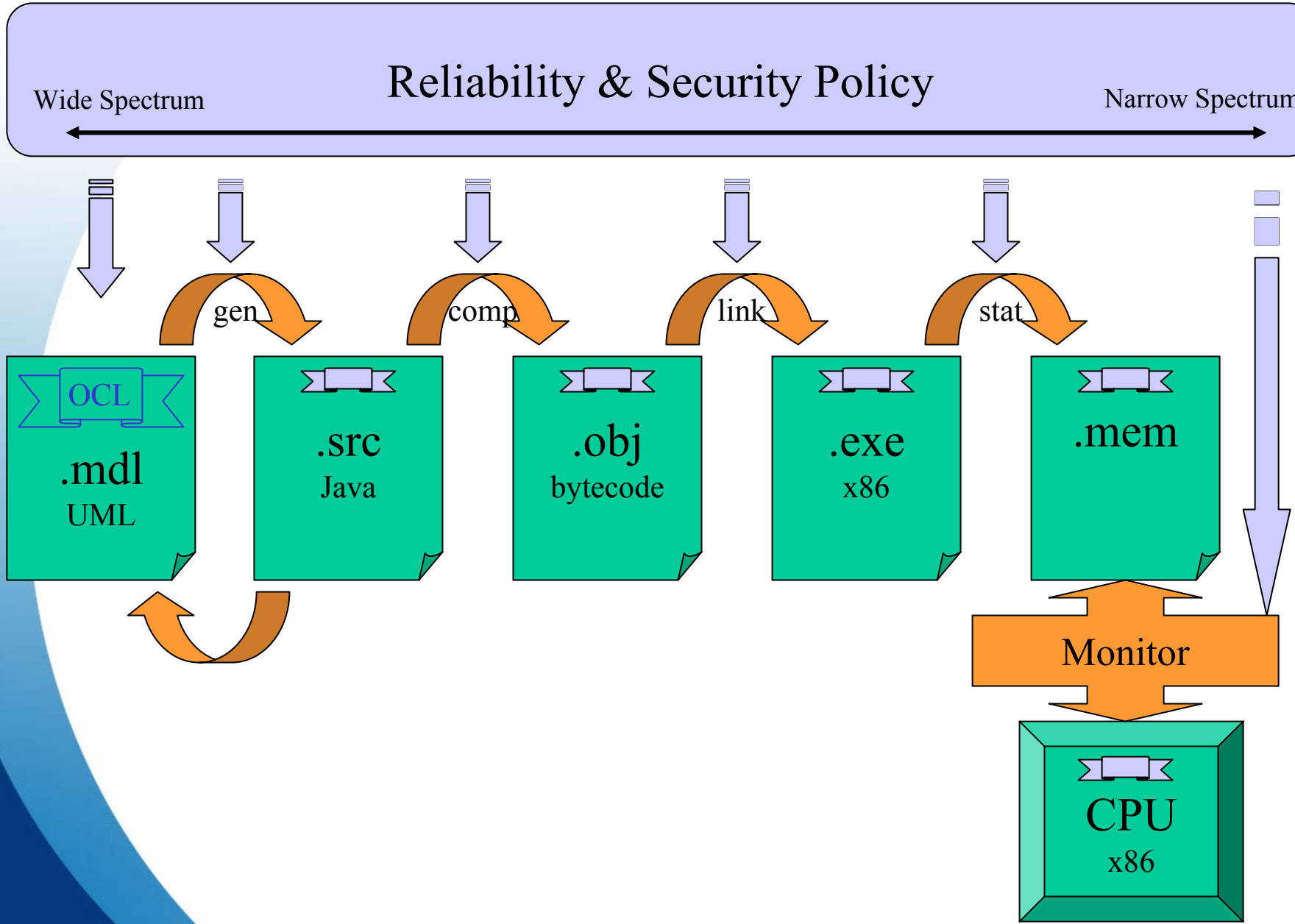
# Conclusion form the MaliCOTS Project

- Security must be clearly defined by a policy to be manageable
- Static and dynamic approaches combined in a test-bed:
  - *offers a short-term solution*
  - *may be lengthy and cumbersome processes*
- Certifying compilers:
  - *emerging technology for large COTS certification --- rapidly*
  - *capabilities confirmed by MaliCOTS prototypes*

# SOCLe project:
## Secure OCL expressions for C2IS Modelling

*Socle* *is defined by Webster's as:*

*the base of a column, pedestal or a superstructure*


Reliability & Security Policy
Wide Spectrum
Narrow Spectrum
gen
comp
link
stat
OCL
.mdl
UML
.src
Java
.obj
bytecode
.exe
x86
.mem
Monitor
CPU
x86

# UML and OCL

- UML is the de-facto standard software notation
  - UML v 2 is its final approval stage at the OMG
  - Used for C2IS and other Government critical systems
- OCL is a complementary constraint language
  - To formulate pre/post conditions and invariants
  - To eliminate ambiguities in software design

*UML: Unified Modeling Language*
*OCL: Object Constraint Language*
*OMG: Object Management Group*

# OCL: *a Good Technology*

- OCL improves quality
  - U. Laval (1997) – *OCL improves greatly software quality*
  - Nurun (1999) – *10% to 15% additional effort in C2IS design*
- Can be used for security
  - SecureSoft (2001) – *OCL has the expressivity for security*
  - SecureSoft (2001) – *Constraints can be imposed on user's behaviour*
- UML/OCL can be formalized to a large extent
  - Poly-MTL (2002) – *OCL is evolving in UML v 2*
  - Poly-MTL (2002) - *Model-checking OCL Constraints is feasible*

# Secure OCL Demonstration (2002-2005)

- Design C2IS in the usual way :
  - State-chart, class diagrams, collaboration diagram, etc
  - + OCL constraints for reliability and security
- OCL checker constructs an underlying model
  - Formally verified by « hidden » model-checking for coherence and completeness
  - Transparent to the designer
- Detailed risk management delegated to appropriate certification engines

# Technical Conclusions

- Software certification from design to binary
  - Via multiple certification engines
    - specialized and activated on request
- Integration of security policy in design
  - Better understanding of security constraints
  - Assignment of responsibilities to the best engine
  - Progressive security enforcement

# Defensive Software Design & Programming

- Dependence on software steadily increasing
- Software quality is good to very good

  *but inadequate for C2IS and other critical systems*
- C2IS reliability & security

  *from methodologies (1990-2000)*

  *to certification (2001-2010)*

Robert.Charpentier@drdc-rddc.gc.ca
http://www.drdc-rddc.gc.ca/researchtech/malicots/home_e.asp