TM

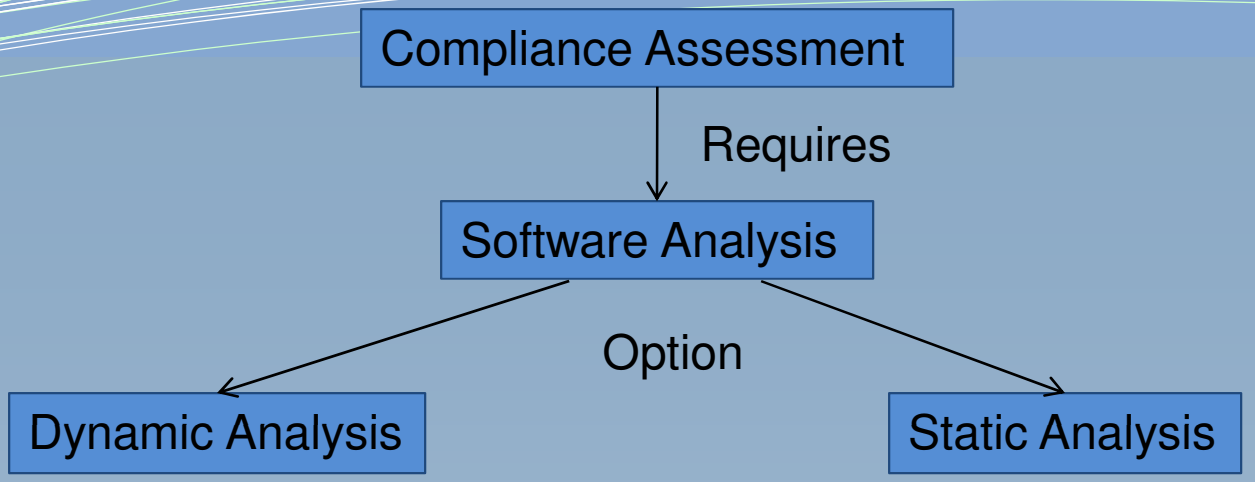
Redefining Static Analysis

A Standards Approach

Mike Oara

CTO, Hatha Systems

Software Analysis for Compliance



Performed on executable programs built from that software system running on a real or virtual processor

Performed without executing programs, but by employing complex analytical tools which investigate the actual source code of the system

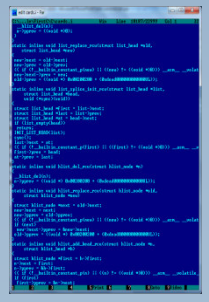
Dynamic Analysis

Methods	Benefits	Limitations
Inspecting the user interfaces Performing multiple incident specific tests Measuring performance Checking memory allocation Configuration Management	No source code required A realistic view of the software system in action Performance data available Locking down system removing unnecessary access	No certainty of code coverage Number of tests maybe overwhelming Difficult to replicate runtime environment

Flight 501, which took place on June 4, 1996, was the first, and unsuccessful, test flight of the European Ariane 5 expendable launch system. Due to an error in the software design (inadequate protection from integer overflow), the rocket veered off its flight path 37 seconds after launch and was destroyed by its automated self-destruct system when high aerodynamic forces caused the core of the vehicle to disintegrate. It is one of the most infamous computer bugs in history. (Wikipedia)

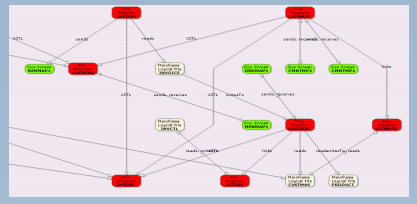
Static Analysis

Methods	Benefits	Limitations
Architecture diagrams Data flows Control flows Investigative queries Software measurements Pattern discovery Simulation	System understanding in breath and depth Direct discovery of non-compliance Pinpointing errors in the code, architecture, process Proof of compliance No “guesswork” testing	No performance data Some run-time conditions may not be captured Source code is required



Source code

Parsing

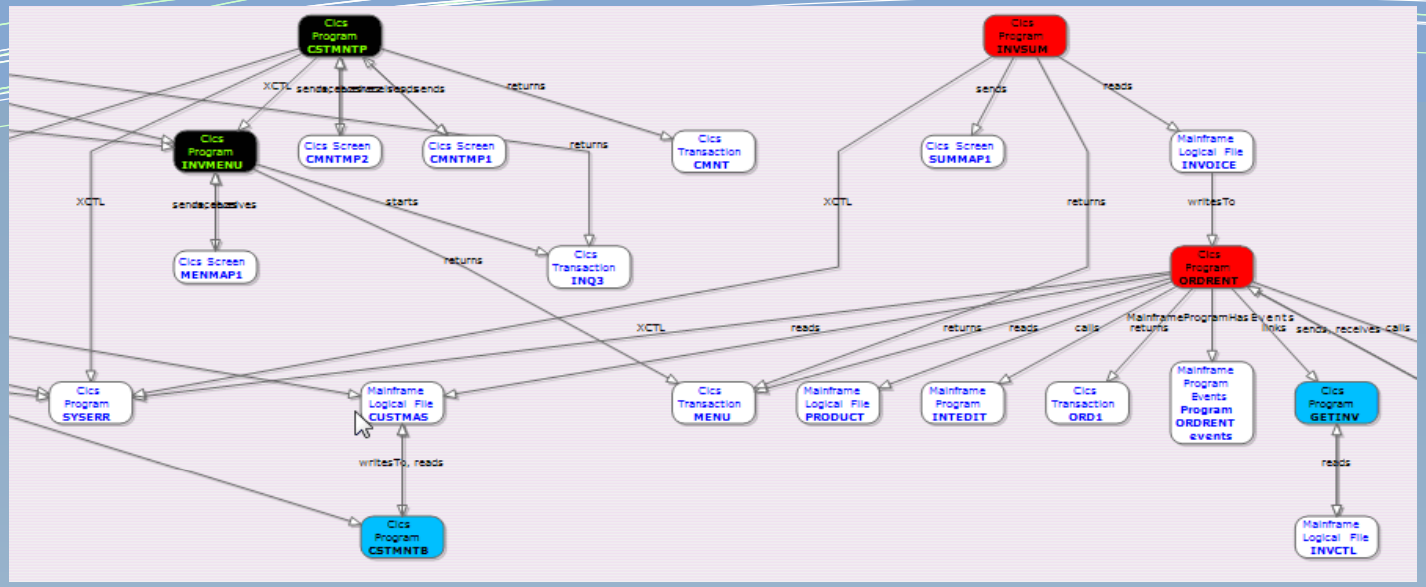


Repository model

Analysis

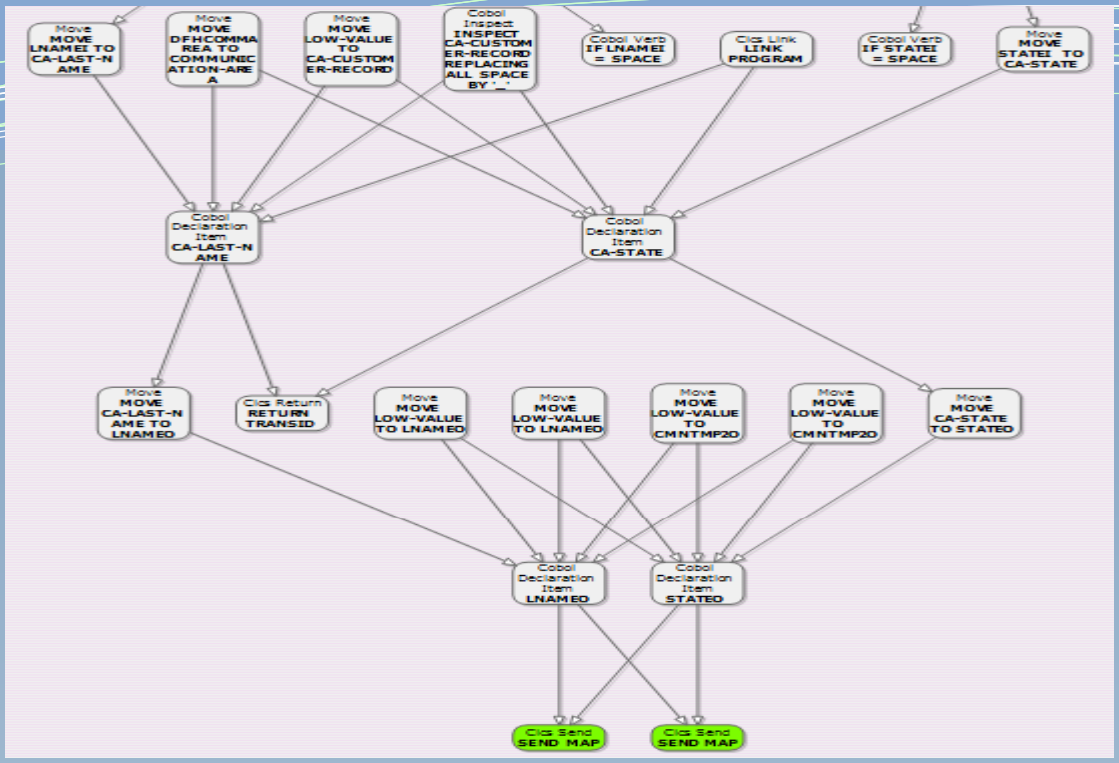
Comprehensive System Diagrams
User Definable Queries
Both comprehensive and Custom/specialized analysis tools

Static Analysis: Platform Diagrams



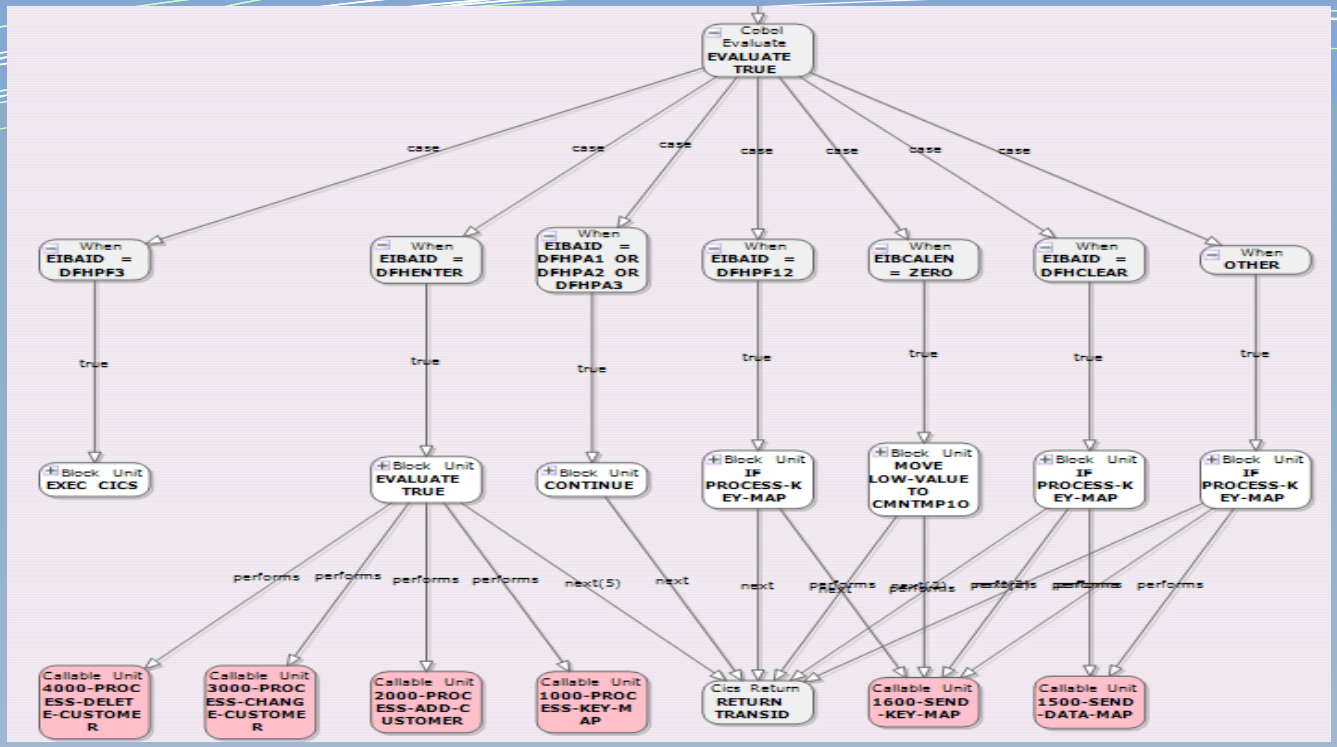
Definition	Compliance applications
A diagram of relationships between runtime elements of the application (programs, transactions, tables, files, queues, etc.)	Is the application properly layered? Are all data access methods done through an API layer? Which elements of the application work with confidential or secret information?

Static Analysis: Data Flows



Definition	Compliance applications
Shows dependencies between variables	What data appears in a user interface? (HIPAA) Was data encrypted before being stored or transmitted?

Static Analysis: Control Flow



Definition	Compliance applications
Shows flow dependencies between statements	Is data validated before being processed? What conditions lead to a particular action? (example: approval of a request)

Static Analysis: Queries

Static analysis tools look at the code as raw data. They can therefore query and discover predefined patterns of concern for compliance.

Example

To guard against internal attacks, find if there is any place in the code where special logic is executed for a particular hard-coded account.

```
//ActionElement[@kind="Branch" and exists(->Argument[@name like "*CUST*"]) and exists(->Argument:ValueElement)]
```

Query	Compliance applications
Find mixed IO programs	Compliance to architecture standards. User interfaces should be separated from data access.
Find data validations	Is data correctly validated before being accepted to be processed?
Find coding patterns	Discover common code weaknesses

Query results can be immediately seen in the context of data flows, control flows or platform architecture!

Static Analysis: Offering evidence

Reports based on queries

11/24/2010

SporaEng - Weknesses

Function	Use of strcat()	Use of fgets()	Use of free()
isdigit	0	0	0
process in rs field	0	0	0
teacher in cr report	0	0	0
select list helper	0	0	0
add to teacher collection		)	0
show comment	0	0	0
add_select_list	0	0	0
strtok	0	0	0
save attrs	0	0	0
ferror	0	0	0
strcopy	0	0	0
null char matrix	0	0	0
isalnum	0	0	0
mark select list item	0	0	0

Activity log

Error List | Search Results (14) | Bookmarks | Activity Log

Activity: all | Redo | Remove

Time	Action	Selected	Parameters	Result
This session				
4:13 PM	build	SporaEng	action=rebuild; files=34;	errors=0; warnings=0;
4:15 PM	search	SporaEng	query=//ActionElement[((@kind = 'Branch') and exists(->argume...	count=0;
4:15 PM	search	SporaEng	query=//CallableUnit[(@name = 'fgets')]->iscalledby:*;	count=8;
4:16 PM	search	SporaEng	query=//CallableUnit[(@name = 'free')]->iscalledby:*;	count=3;
4:16 PM	search	SporaEng	query=//CallableUnit[(@name = 'strcat')]->iscalledby:*;	count=14;

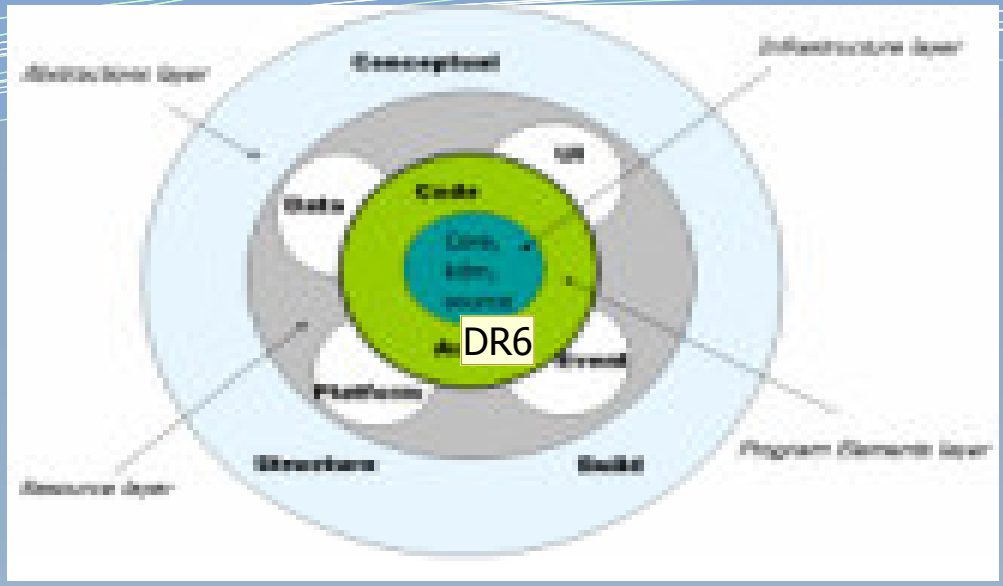
Current Static Analysis Approach – Practical difficulties

Issue	Consequences
Large variety of languages and technologies	No static analysis tools cover all technologies. If many tools are used, results are hard to merge.
Large variety of regulatory requirements	No static analysis tool covers all requirements. Some may offer good data flow analysis, some good queries for weaknesses, some good architecture diagrams.
Large variety of regulatory authorities	Each regulatory authority may require a certain type of reporting
Flexible use of advancements in the technology and methodology of analysis	Tools and their approaches can quickly become obsolete; stovepiped solutions while some tools may improve more. How can you switch without sacrificing the current investments?

Solution

Use of standard models, commonly accepted terms, commonly accepted methodologies and reporting.

Standard models: KDM



- **<u>Knowledge Discovery Metamodel (KDM):</u> an ISO/OMG Standard providing ontology (a set of definitions) for system knowledge extraction and analysis. KDM provides a framework for the capture of code, platform and other software system characteristics. This further allows the extraction of data flows, control flows, architectures, business/operational rules, business/operational terms, and the derivation of business/operational process; the extraction can be delivered from source, binary, or byte code. Additionally the intermediate representation of the extraction is in executable models creating the possibility of simulation and code generation.**

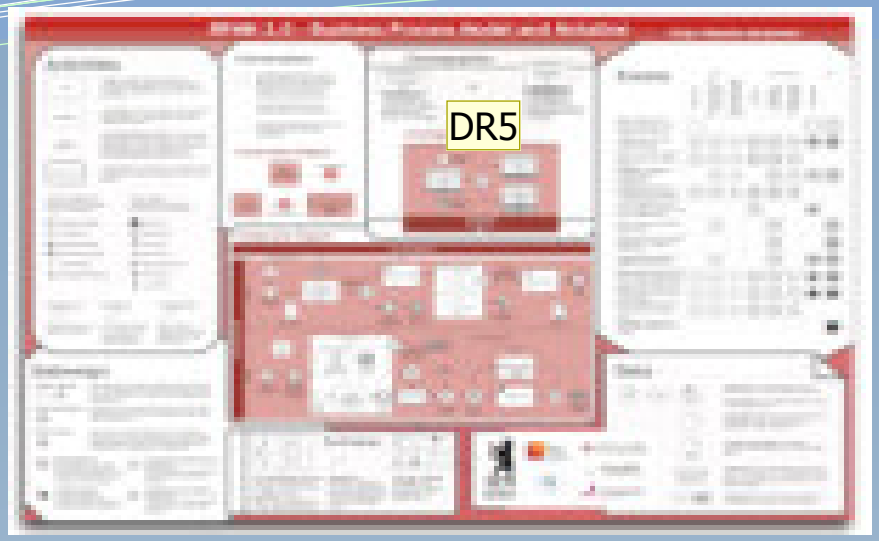
DR6 Recapture from OMG site

Dennis Rosynek, 11/26/2010

Standard models: SBVR

- **<u>SBVR (Semantics of Business Vocabulary and Business Rules):</u> An ISO/OMG standard, this specification provides a structured process for formalizing, in natural language, the existing English language representation of compliance points. The standard enables the various compliance specifications (e.g. FISMA, HIPAA, SOX, FIPs, CWEs, etc) to be formalized reducing the room for interpretation from organization to organization when implementing the compliance and auditing requirements.**

Standard models: BPMN



- **<u>Business Process Modeling Notation (BPMN)</u>: an OMG standard delivering a modeling notation used to capture business/operational processes in support of system and organizational process simulation and analysis. It is used today to capture both human and IT system processes for the purposes of simulating environments both 'as is' and 'to be' for software modernization. This notation is compatible with KDM so that system extraction can be represented in BPMN for gap analysis of the current state of the system vs. what is thought to be the current state of the system – critical for modernization and compliance.**

Slide 13

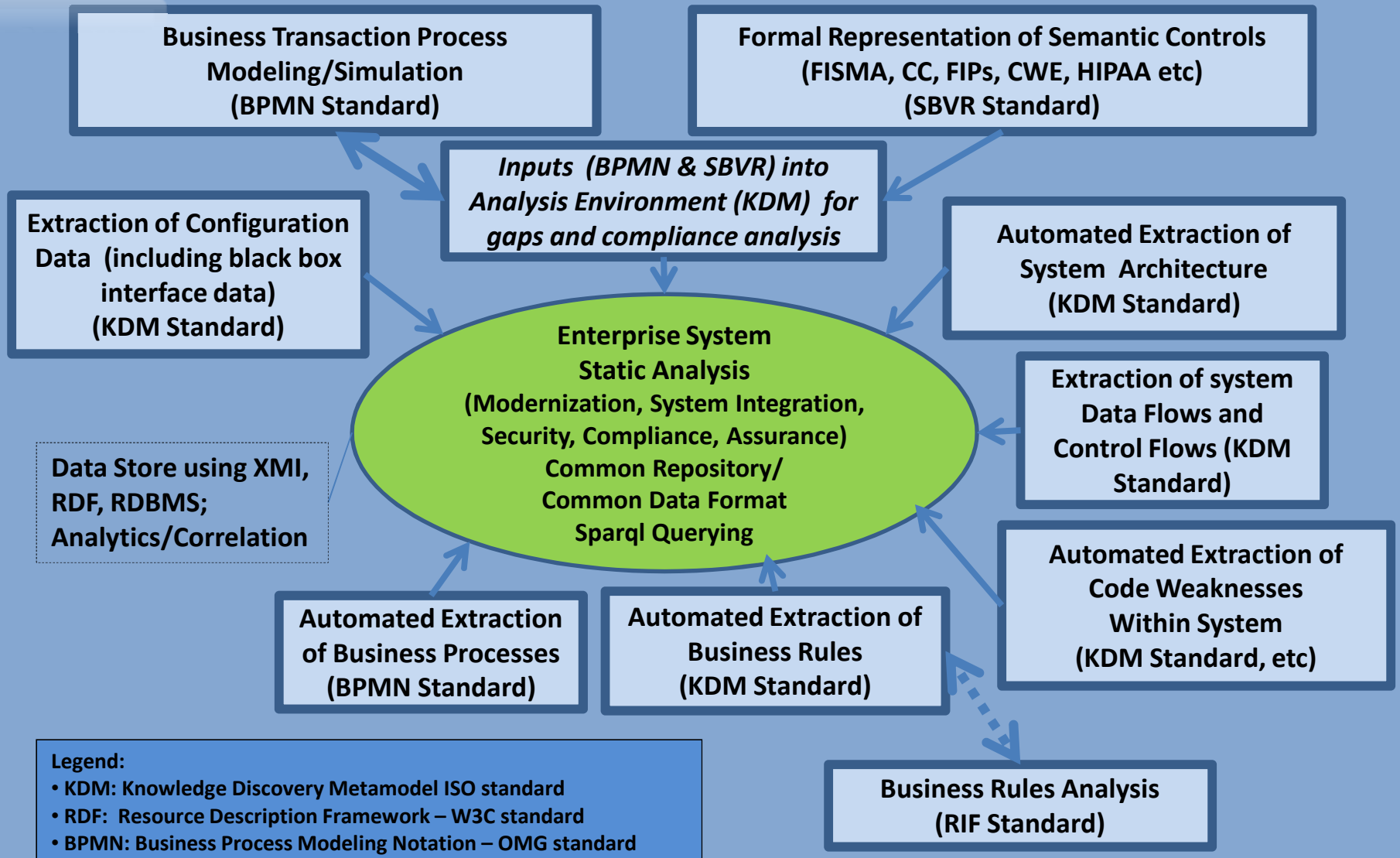
DR5 Recapture from internet the photo for the granularity

Dennis Rosynek, 11/26/2010

Standard models: RDF

- Data/Metadata Storage Standard (RDF): With the emergence of the standards noted above and the need for storing this information for analysis, a set of storage standards needed to be embraced. XMI, RDMBS, and RDF (Resource Description Framework) are the three formats that are compatible with these standards. RDF - perhaps the least known of them - is a W3C standard that is compatible with KDM and BPMN. There is a specific approach in the standard called RDF triple store which is currently being used in semantic web applications. The value of RDF is that it can manage large amounts of data and metadata which is critical for doing comprehensive static analysis.

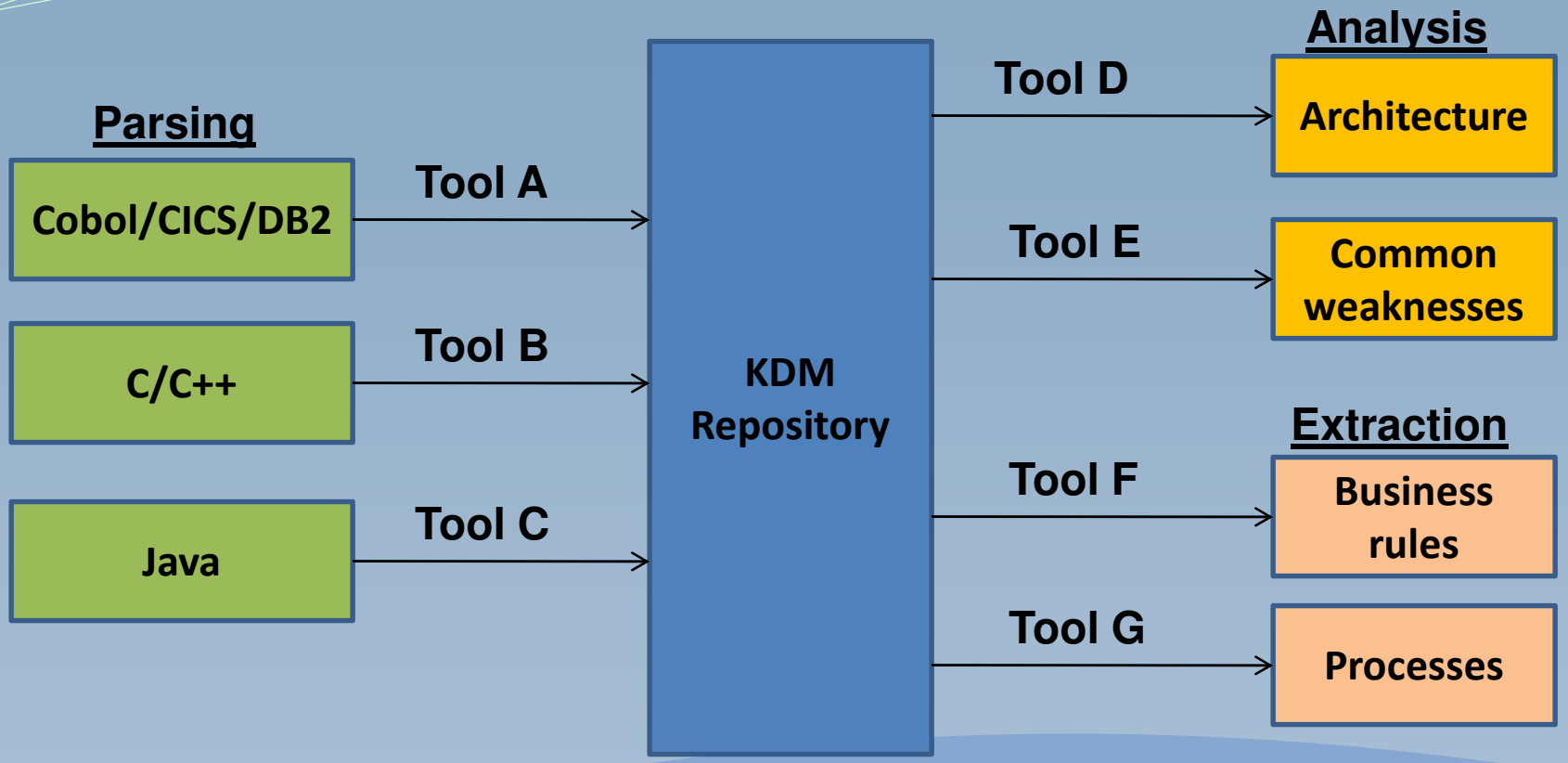
Enterprise Software Systems Static Analysis Standards Mapping



Legend:
- KDM: Knowledge Discovery Metamodel ISO standard
- RDF: Resource Description Framework – W3C standard
- BPMN: Business Process Modeling Notation – OMG standard
- SBVR: Semantics of Business Vocabulary and Rules - ISO standard

Static Analysis & Standards

A Scenario



™

Thank You

Mike Oara

mike.oara@hathasystems.com

919-931-9997