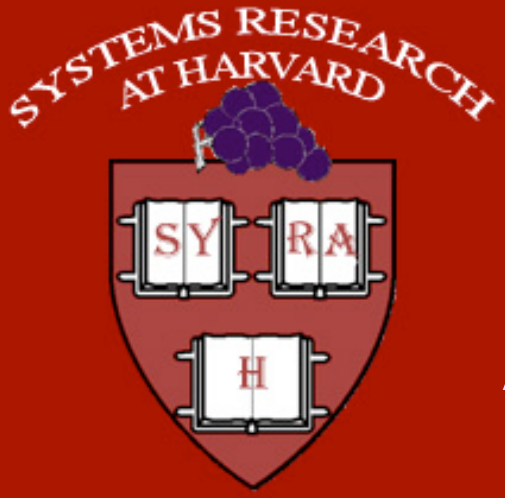
Towards Query Interoperability:

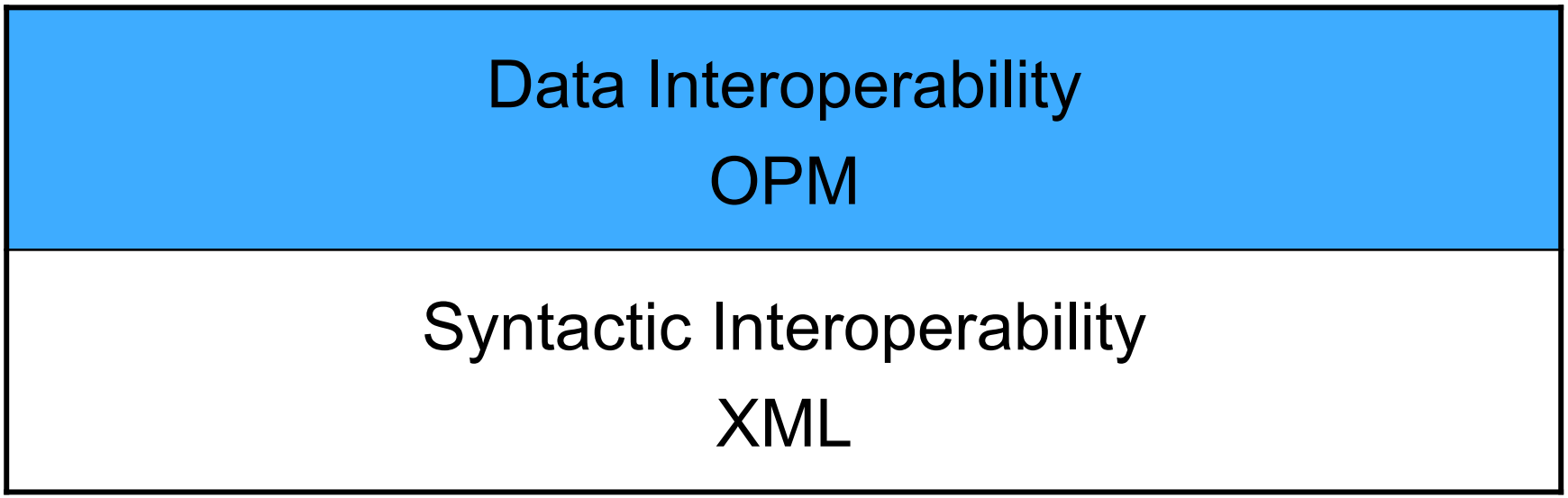
PASSing PLUS



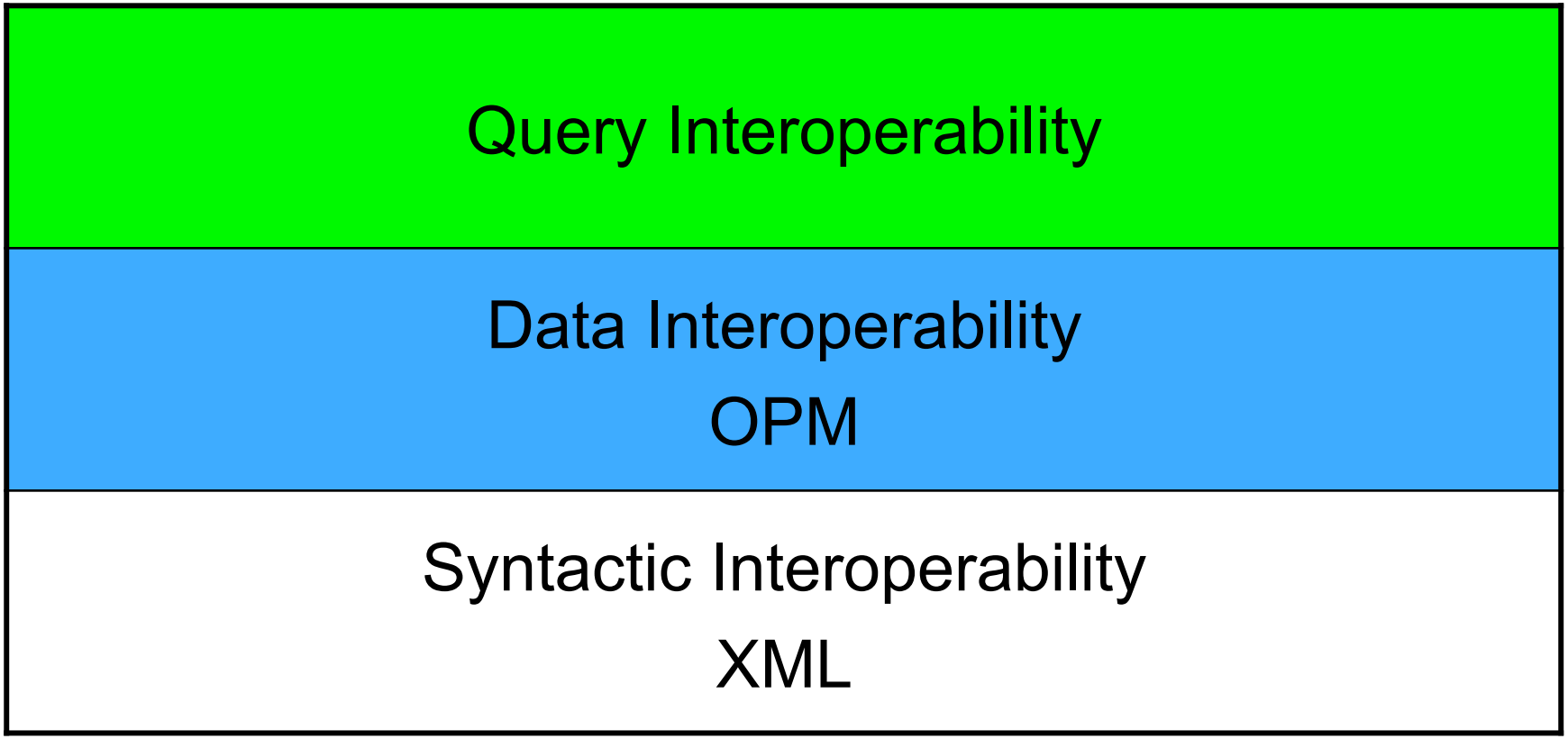
Uri Braun and Margo Seltzer
Harvard University

Adriane Chapman, Barbara Blaustein,
M. David Allen and Len Seligman
MITRE

Interoperability

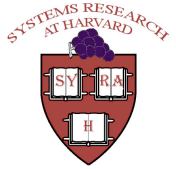


Goal


Query Interoperability
Data Interoperability
OPM
Syntactic Interoperability
XML

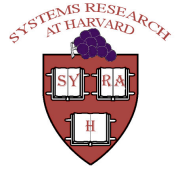
Current Status

- Harmony
  - Desire to share
  - Open Provenance Model (OPM)
- Fragmentation
  - Data sets
  - Semantics
  - Visualization tools
  - Query interfaces



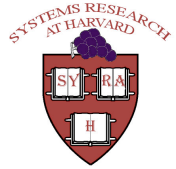
What is Query Interoperability?

- Given data collected on PASS
- Query results are the same
  - Query on PASS
  - Import and query on PLUS



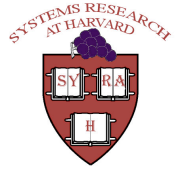
Towards Query Interoperability

- Include *all* documented relationships among entities
- Identify *all* activities
- Resolve entity names
- Match process arguments



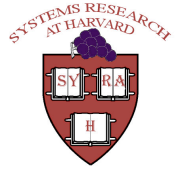
PASSing PLUS

- Approach interoperability by example
- Start with two systems that
  - Seek to adhere to OPM
  - Work fine on their own
- Find that they do not interoperate well



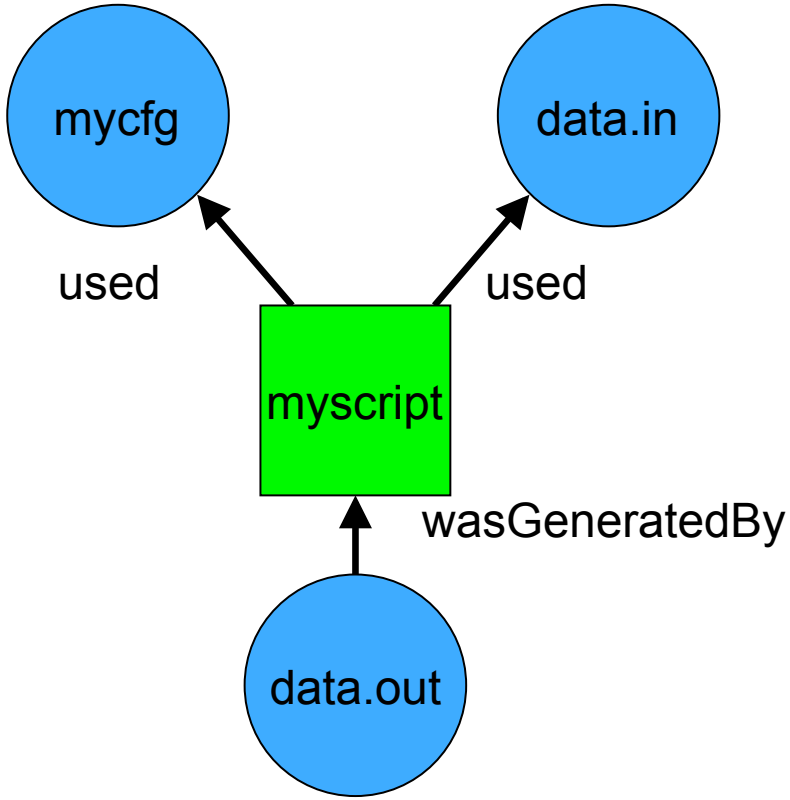
Summary of Systems

	PASS	PLUS
Team	Harvard	MITRE
Target	OS level	Workflow
Implementation	Linux Kernel	Java & MySQL
Recording	System calls	API

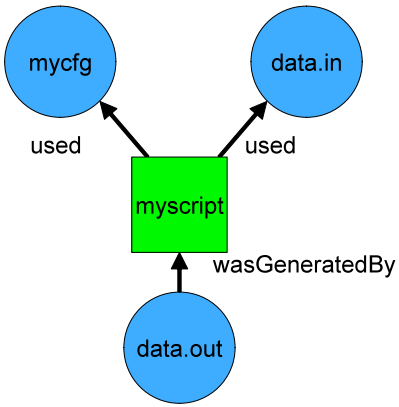


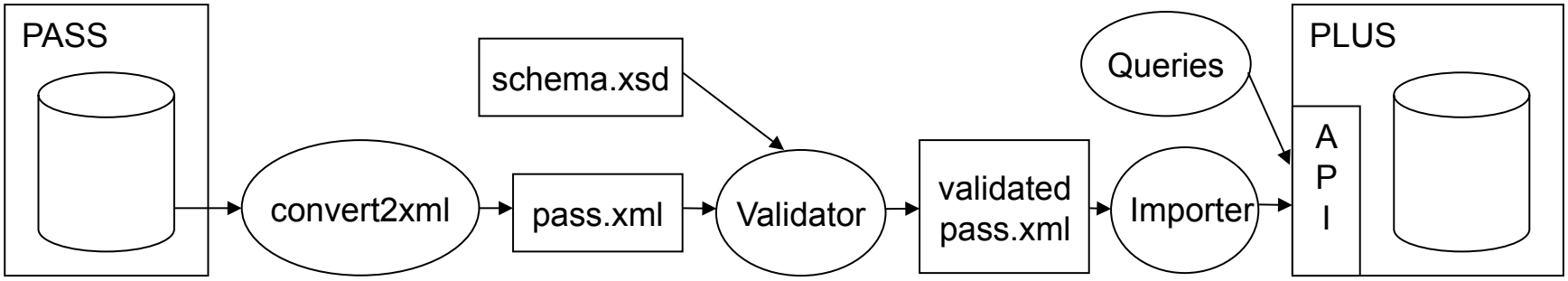
PASS Example

myscript -v --ignore-case -v --case-sensitive mycfg data.in data.out

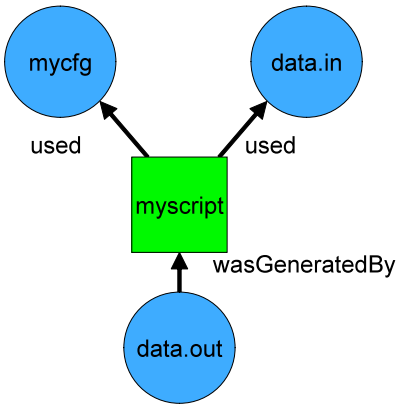


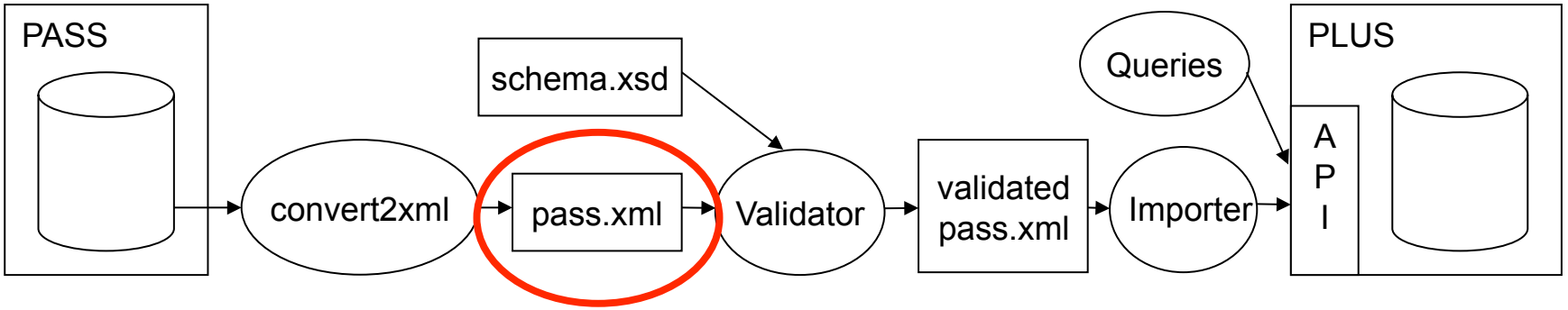
PASS-PLUS Architecture





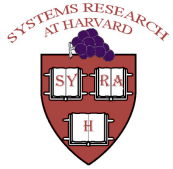
Provenance Transfer





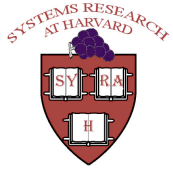
PASS XML: File

```
<provenance pnode="2" version="0">
  <record type="TYPE">
    <data>FILE</data></record>
  <record type="NAME">
    <data>myscript</data></record>
  ...
```

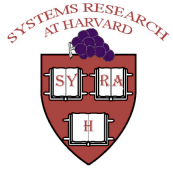
PNodes and Versions

```
<provenance pnode="2" version="0">
  <record type="TYPE">
    <data>FILE</data></record>
  <record type="NAME">
    <data>myscript</data></record>
  ...
```

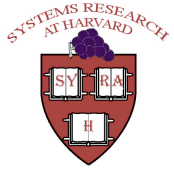
Identity Information

```
<provenance pnode="2" version="0">
  <record type="TYPE">
    <data>FILE</data></record>
  <record type="NAME">
    <data>myscript</data></record>
  ...
```

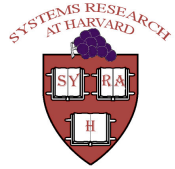
Challenges with Entities

- Identification
  - <pnode,version> vs. uid
- Type specification
  - Version 0 only
- Version representation



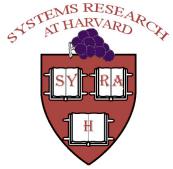
Solutions for Entities

- Identification — Dictionaries
  - <pnode,version> vs. uid
- Type specification — Attach to all
  - Version 0 only
- Version representation — Collections



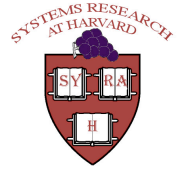
PASS XML: Process

```
<provenance pnode="13" version="2">
  <record type="TYPE">
    <data>PROC</data></record>
  <record type="ARGV">
    <data>myscript -v --ignore-case -v
--case-sensitive mycfg data.in data.out
    </data></record>
  ...
```

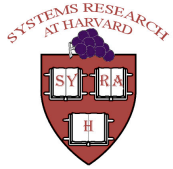
Arguments

```
<provenance pnode="13" version="2">
  <record type="TYPE">
    <data>PROC</data></record>
  <record type="ARGV">
    <data>myscript -v --ignore-case -v
--case-sensitive mycfg data.in data.out
    </data></record>
  ...
```



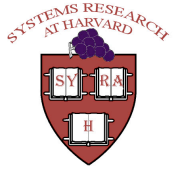
Activities

- Activity is what we are executing
  - myscript
- If we run myscript 20 times we have 20 invocations of the same activity
- PASS does not distinguish the activity



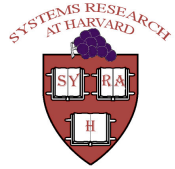
Challenges with Processes

- Encoding Arguments
- Identifying Activities



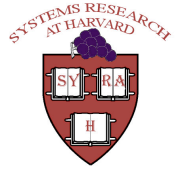
Solutions to Processes

- Encoding Arguments — Collections
- Identifying Activities — Make explicit



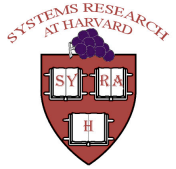
PASS XML: Relationship

```
<provenance pnode="13" version="2">
  <record type="INPUT">
    <xref pnode="2" version="1"/></record>
  ...
```

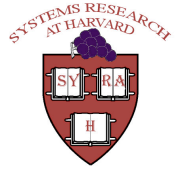
Recording Relationships

- Where to records dependencies:
  - PASS descendant
  - PLUS process
- Conceptual model
  - Long living processes/files
  - Versions



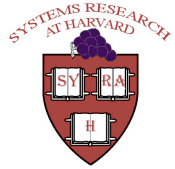
Forward References

<provenance pnode="13" version="2">
  <record type="INPUT">
    <xref pnode="15" version="1"/></record>
  ...
<provenance pnode="15" version="1">
  ...



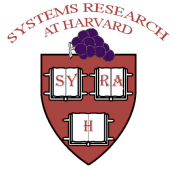
Unresolved References

- Refer to an entity that does not appear
- Caused by semantic assumption
  - PASS does not reify unchanged versions



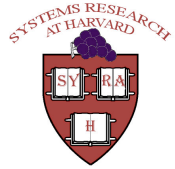
Challenges with Relationships

- Entity to attach to
- Forward References
- Unresolved References

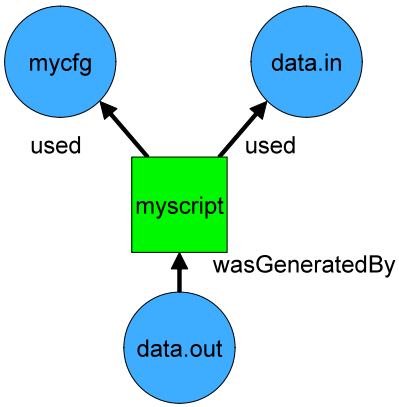


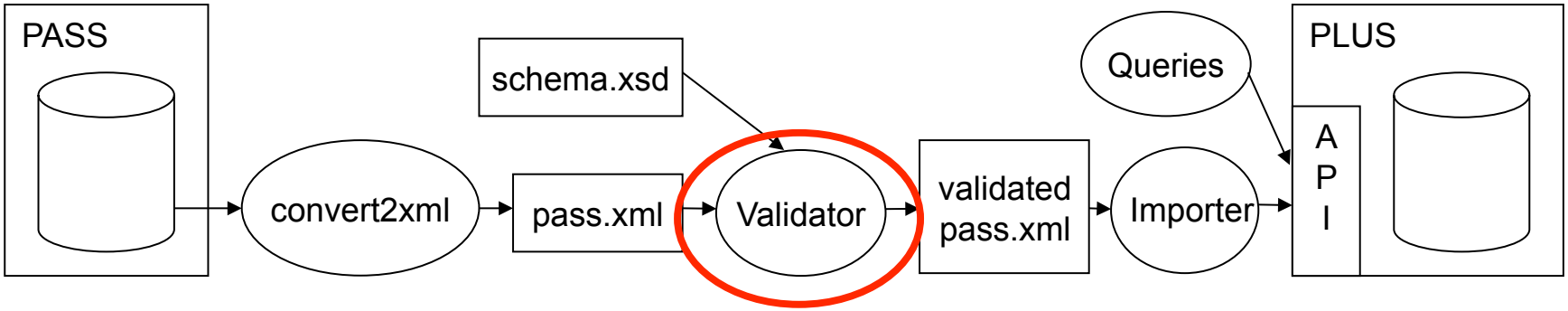
Solutions to Relationships

Entity to attach to	Descendant
Forward References	Sort
Unresolved References	Disallow



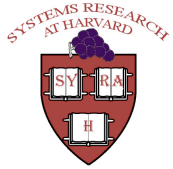
Validation


mycfg
data.in
used
used
myscript
wasGeneratedBy
data.out


PASS
schema.xsd
Queries
PLUS
convert2xml
pass.xml
Validator
validated
pass.xml
Importer
A
P
I

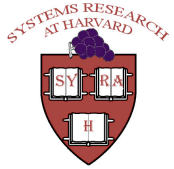
Challenges with Validation

- OPM validation is via XML schema definition (xsd)
- No way to enforce:
  - No self-loops
  - Acyclic
  - Other problems we just discussed



Solutions to Validation

- Need something stronger than xsd
- Schematron or a similar tool could enforce these rules

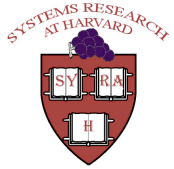


Summary of Challenges & Solutions

Challenges	Solutions
Identification	Dictionaries
Type Specification	Attach to all versions
Versions & Arguments	Collections
Activities	Make Explicit
Relationship Recording	Descendant
Forward References	Sort
Unresolved References	Disallow
Validation	Schematron

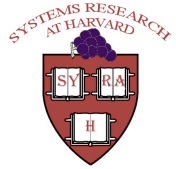
Suggestions

- Concepts
  - Dictionaries
  - Collections
- Constraints
  - Sort
  - No Unresolved
- Validation
  - Schematron



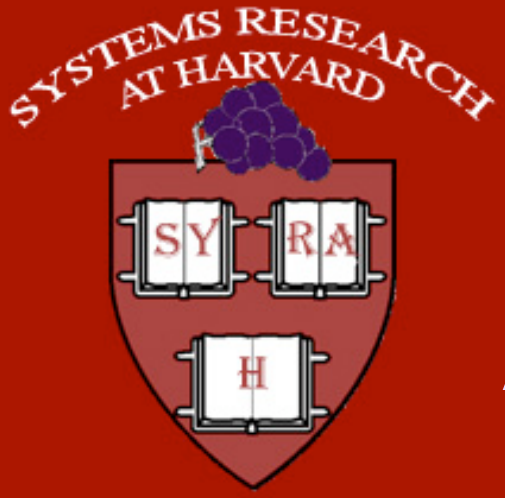
Conclusions

- Our goal should be query interoperability
  - Enable sharing of tools
- We have more work to do on semantics
  - Concepts
  - Constraints
  - Validation



Towards Query Interoperability:

PASSing PLUS



Uri Braun and Margo Seltzer
Harvard University

Adriane Chapman, Barbara Blaustein,

M. David Allen and Len Seligman

MITRE