Provenance Challenge 2

Joe Futrelle
National Center for Supercomputing Applications

Dead Greeks Agree

"The unapparent connection is more powerful than the apparent one."

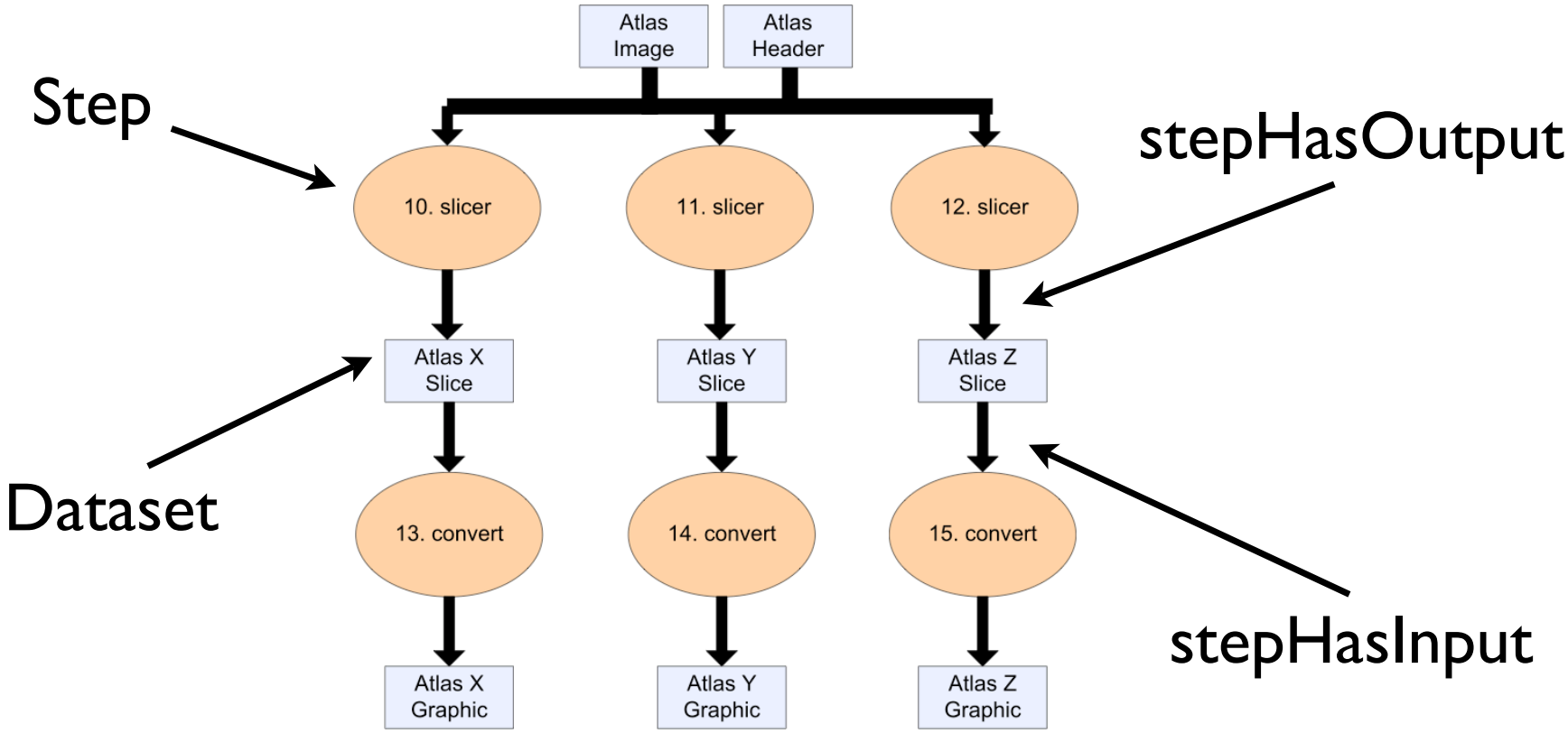
- Heraclitus, Fragment 54

Methodology

Approach

- Develop a minimal model of workflow provenance based on last year's results
- Interpret each team's trace using that model
- Manually assert correspondences between each team interpretation and the challenge workflow
- Perform queries over all n^3 combinations of partial interpretations w/correspondences

Minimal model



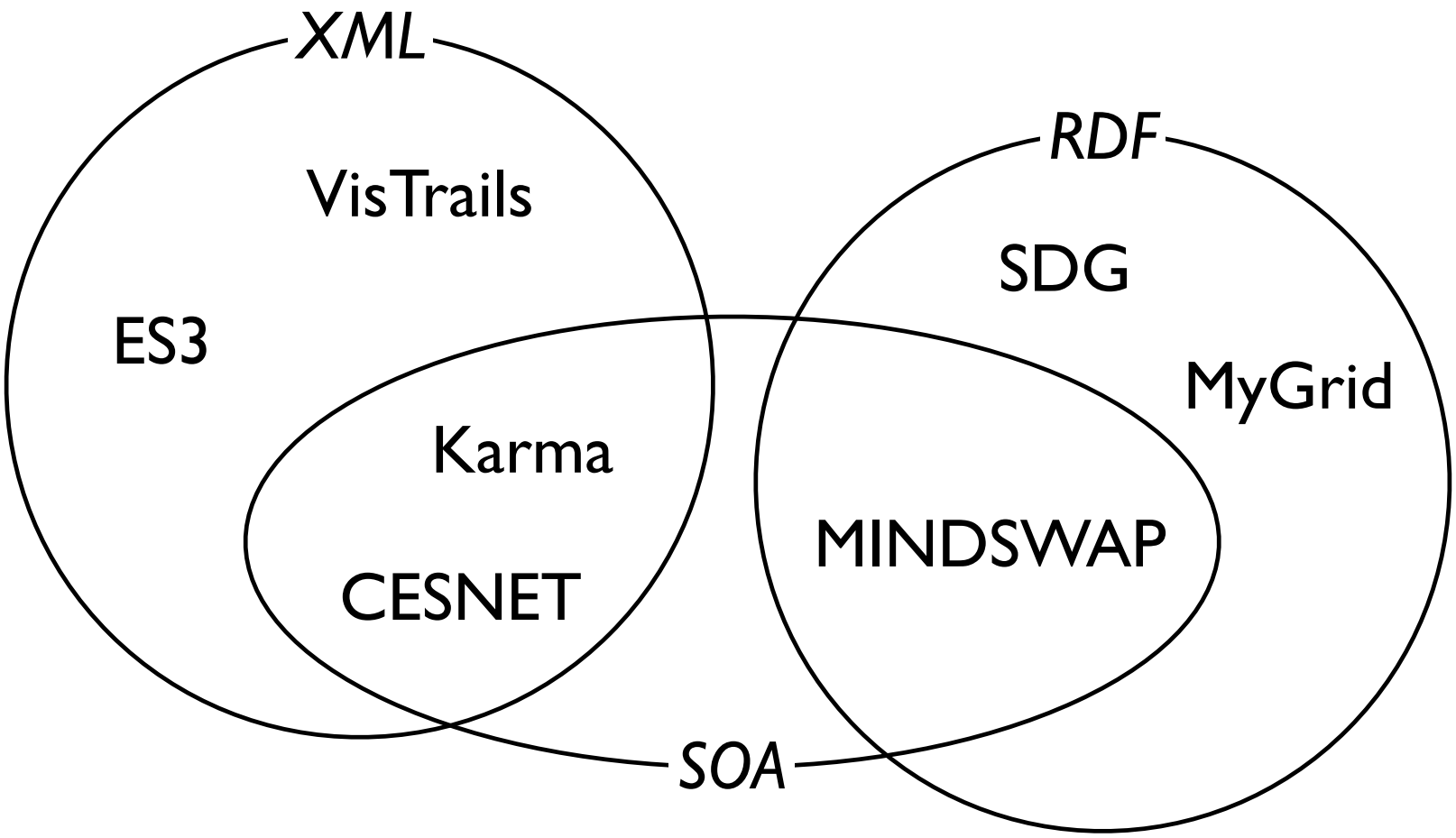
* should have called them “procedure” and “data item”

Interpretation Assumptions

- Naïve interpretation
  - Teams all implemented the challenge workflow, just described it differently
- Open-world assumption
  - Any necessary information apparently missing from a workflow trace is implied by it

Implementation

Teams selected



Integration Strategy

Interpretation → n-way merge

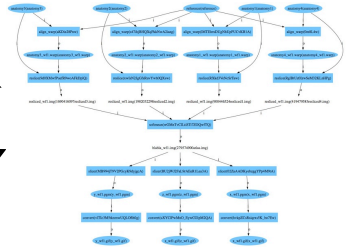
XML

or

RDF

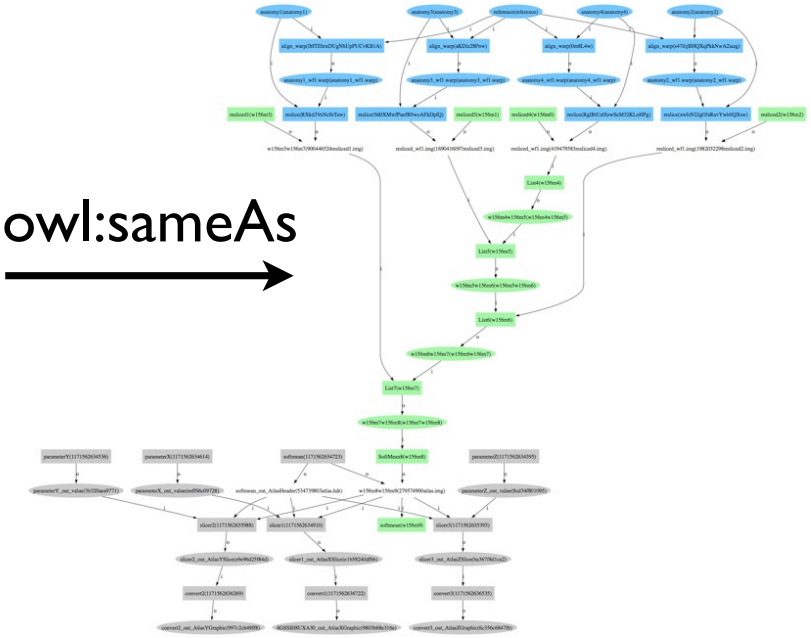
xslt

rules



RDF

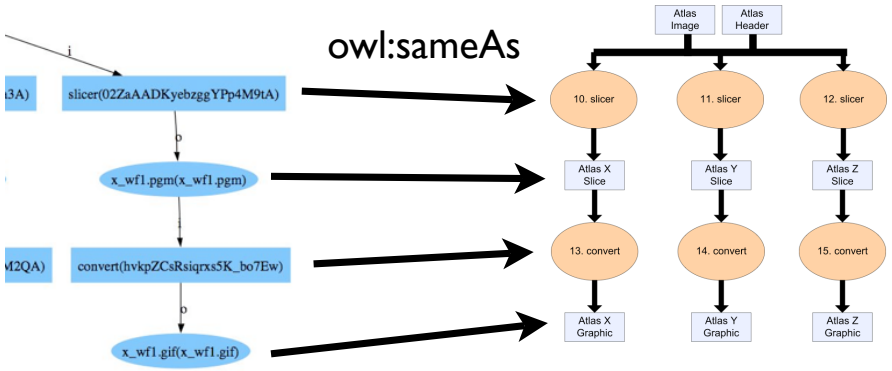
owl:sameAs



7 models → 7^3 models

Query strategy

- Assert equivalence between team-specific Step/Dataset identifiers and corresponding abstract challenge workflow Step/Datasets



- Perform poss. query-specific rules (e.g., infer transitive dependency relationship)

Query 1 model

$\forall a \forall b$: stepHasInput(a,b) $\rightarrow$ dependsOn(a,b)

$\forall a \forall b$: stepHasOutput(a,b) $\rightarrow$ dependsOn(b,a)

$\forall a \forall b \forall c$: dependsOn(a,b)

$\wedge$ dependsOn(b,c)

$\rightarrow$ dependsOn(a,c)

$\forall a$: $a \in R$, dependsOn(atlasXGraphic,a)

where R is the set of all Atlas X Graphic's antecedents

Results and findings

Didn't finish

- XML interpretation was complex because identifiers were difficult to find, assemble, and/or generate from XML
- Manually establishing and checking correspondences across 7 teams was time-consuming
- Ran out of time to finish annotations and do annotation-based queries (just did query #1)

General observations

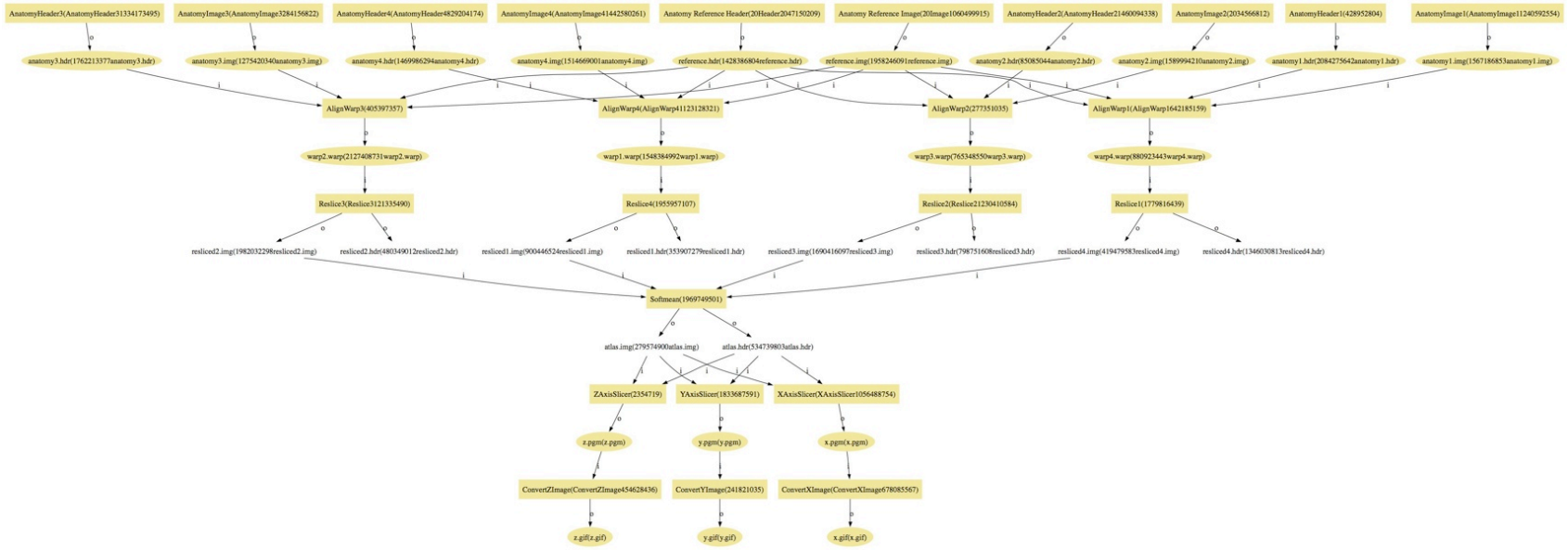
- General agreement with minimal model
- Some traces uninterpretable without a priori knowledge of the challenge workflow (Karma, MINDSWAP)
- Ad-hoc addressing schemes abound
- Metadata often embedded in unstructured data

How hard was it?

Team	B/Java	B/XSLT
SDG	2511	-
MyGrid	3627	-
CESNET	1226	3875
VisTrails	1338	4338
ES3	583	5226
MINDSWAP	6397	-
Karma	611	8261

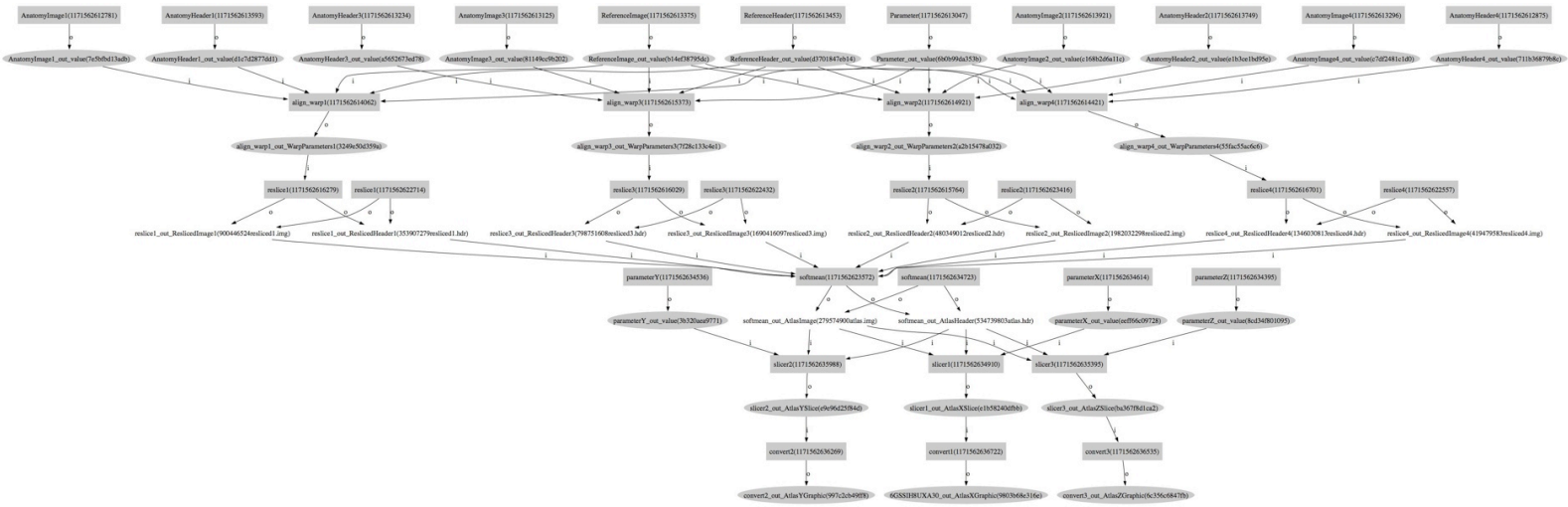
Teams: SDG

- Minimal transformation required
- Modeled part 1 outputs as single data items



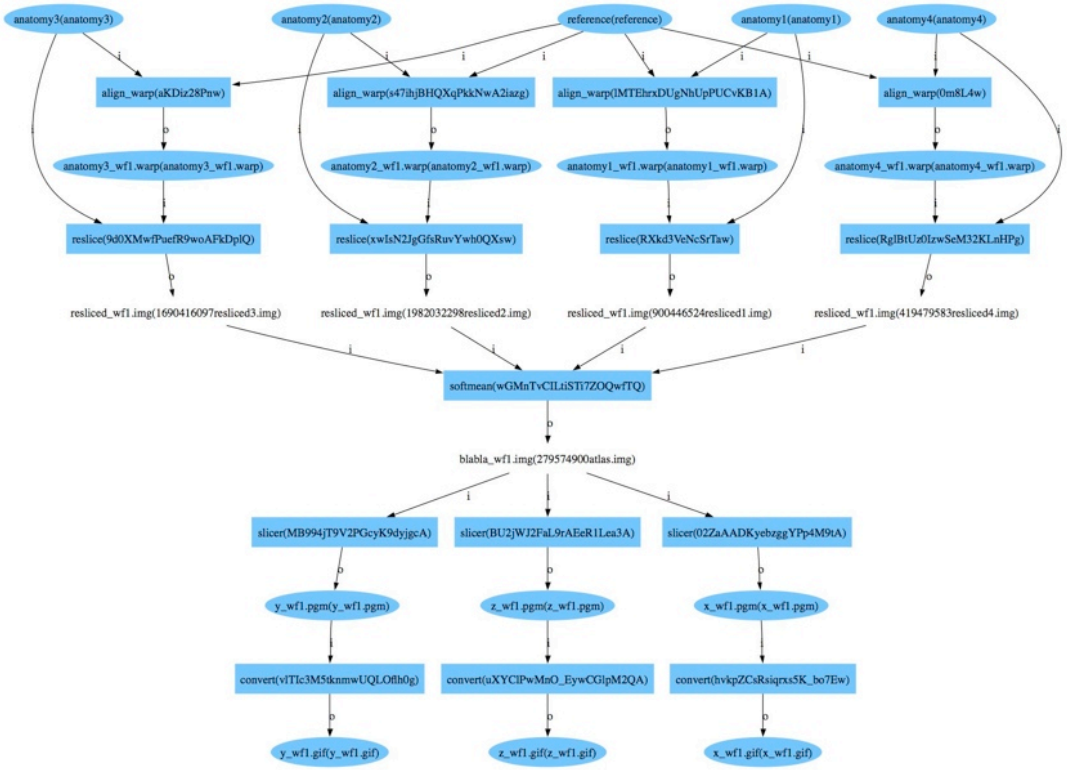
Teams: MyGrid

- Rules used to establish equivalences across workflow parts
- "Extra" inputs representing parameters, etc.



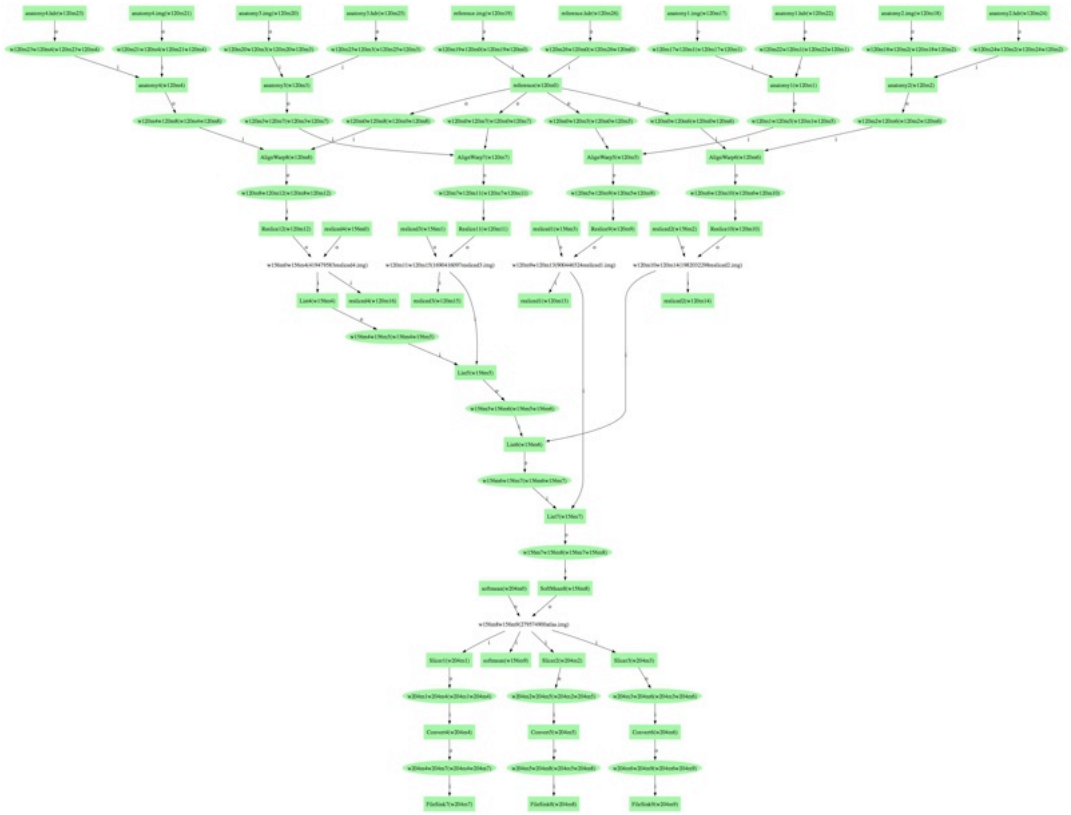
Teams: CESNET

- XML organized by "job," job descriptions contained I/O
- URN and UUID addressing
- No distinction between headers and images



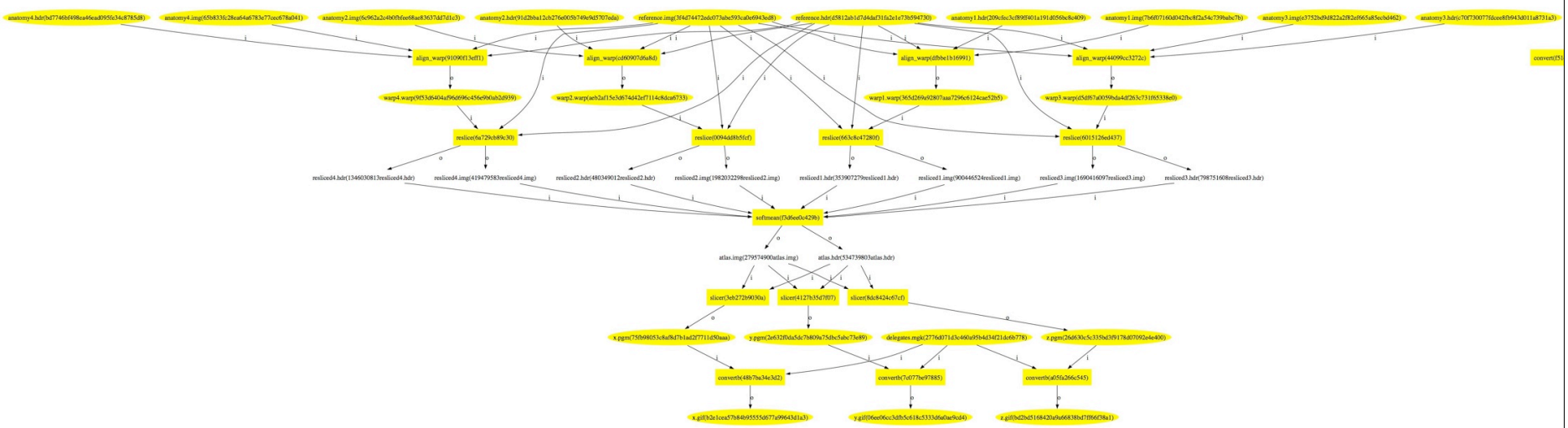
Teams: VisTrails

- No distinction between "procedures" and "data items" (everything is a "module")
- Some modules appear structural e.g., to merge inputs
- Small-integer addressing



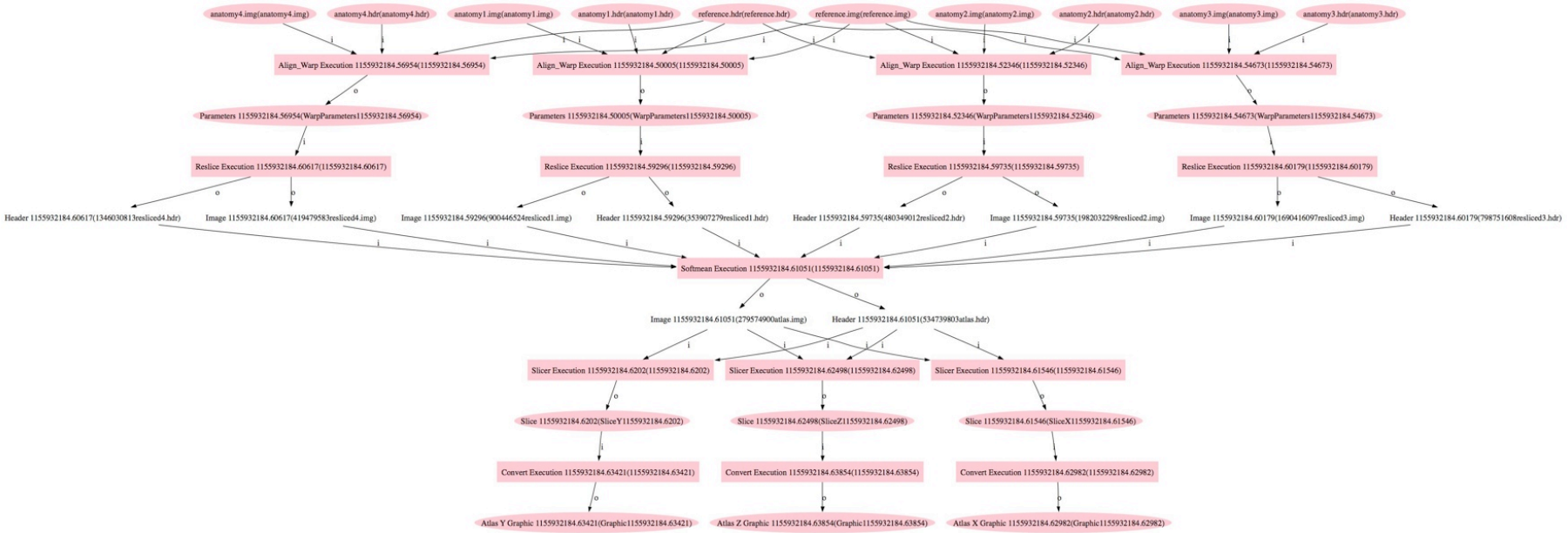
Teams: ES3

- Very close match to model
- "Link" (stepHasInput/Output) between "transformation" (Step) and "file" (Dataset)
- UUID addressing
- Files identified w/ pathnames, so md5sums used instead



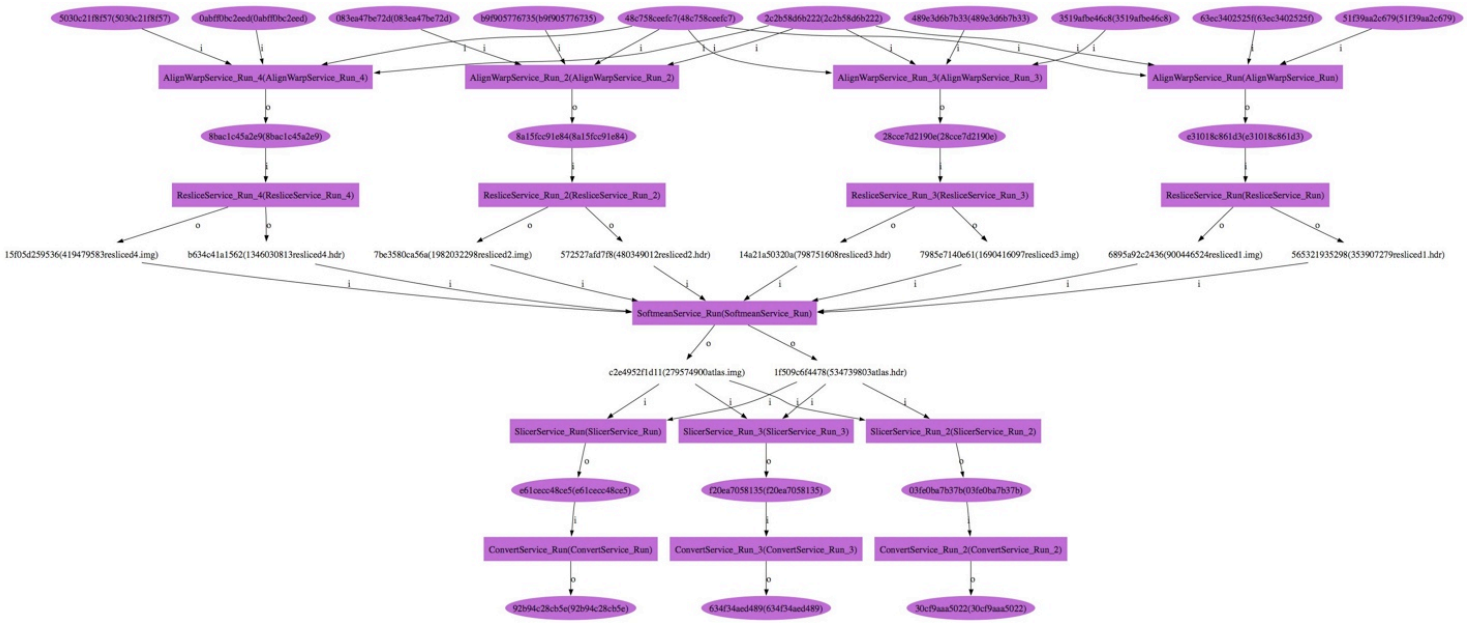
Teams: MINDSWAP

- Challenge-workflow-specific ontology
- Data as "opaque" parameters



Teams: Karma

- Service/event model
- Data as "opaque" parameters
- Challenge-workflow-specific data structures



What not to do

- Use implicitly-scoped identifiers (e.g., "3")
- Imply the existence of procedures and/or data items without identifying them (e.g., by characterizing locators as service-specific parameters)
- Embed important metadata in unstructured data, e.g., identifiers (e.g., "resliced3.img")
- "Ambiguity is maybe sort of bad, I guess"

What to do instead

- Identify everything described using identifiers with explicit scoping guarantees (e.g., UUID's, URI's, URN's, xml:id's)
- Agree on vocabulary--not structure
  - Unlike structures, vocabularies must be mapped by hand