Declarative Specification of Scalable Workflows

Motivation
- Analysts use a variety of data sources and complex tools
- How can we enable analysts to efficiently use data sources and tools in a dynamic, exploratory fashion, while keeping track of data provenance?
- Dataflow concurrency can improve the efficiency of a workflow as in the map/reduce pattern
- Enable recommendation and verification of workflows, and identify potential scalability opportunities

Approach
- Provide a declarative language for specifying workflows in terms of data dependencies between actions
- Represent each interaction as a message with in and out fields
- Specify key fields as the unique ID for each message instance
- Identify potential scaling with in key parameters; e.g., the message below produces one L for each H
- Provide provenance tracking with in parameters; H and Q are both data sources for L
- Reduce input via grouping by a shared subset of keys; \{R\} in the message below collects all R values grouped by \(Q,H\), as R has keys \(Q,H,E\)

Specifying a workflow

```
workflow Decision-formation {
  roles: Hgen, Egen, Rgen, Lgen, DCgen
  parameters: out Q key, out DC
  private: H, E, R, L

  User -> Hgen, Egen: Question[out Q]
  Hgen -> Rgen: Hypothesis[in Q, out H key]
  Egen -> Rgen: Evidence[in Q, out E key]
  Rgen -> Lgen: Relationship[in Q, in H key, in E key, out R]
  Lgen -> DCgen: Likelihood[in Q, in H key, in \{R\}, out L]
  DCgen -> User: Decision[in Q, in \{L\}, out DC]
}
```

Benefits
- Clear, concise specification of workflows that can be reused and built into new compositions
- Semantic relationships between messages allow greater flexibility than explicitly sequenced messages
- Typed messages enable recommendation of sources for required data
- Metadata enables automatic workflow scaling and automation for scaling decisions
- Checking message types against available data and component capabilities enables automatic verification of workflow enactments

Message
- Sender: Lgen -> DCgen
- Name: Likelihood
- Parameters: in Q, in H key, in \{R\}, out L
- Recipient(s): DCgen
- Group-by: \{R\}
- Collection: Output

Scaled Workflow