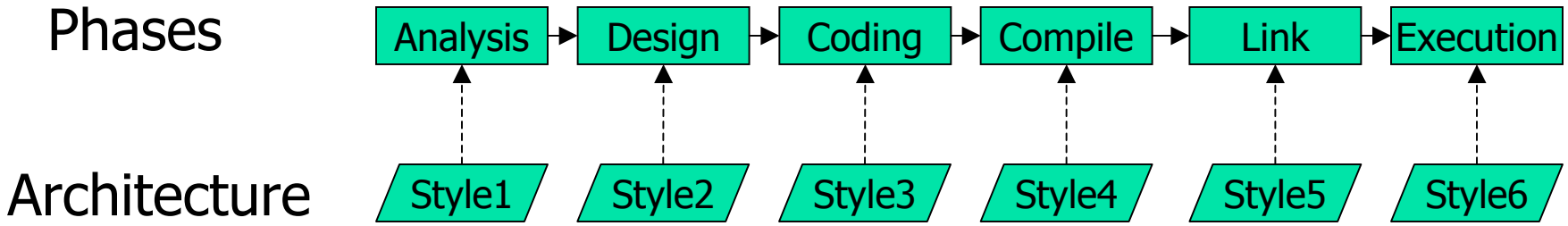
From cradle to grave: An architecture substrate for software lifecycles

Nicolas Rouquette
Principal Member of Technical Staff
Jet Propulsion Laboratory
California Institute of Technology

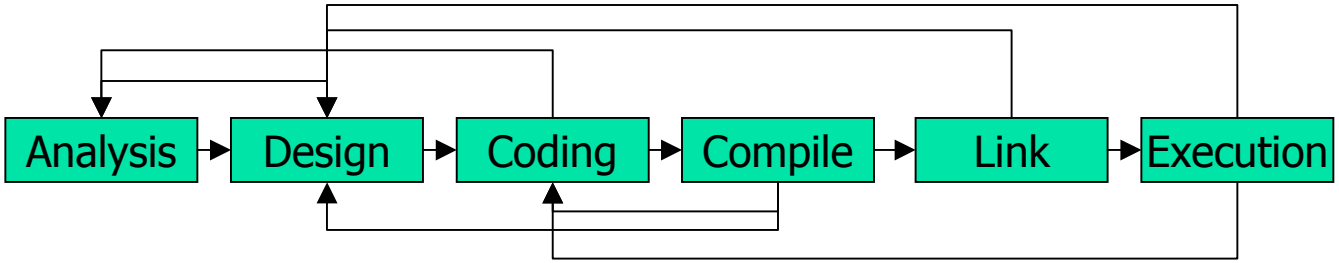
The waterflow lifecycle revisited



- Is there a consensus that architecture is a pervasive concern?
  - Issue: Architecture as a document vs. an engineering model
- Is there a continuity of architecture throughout?
  - Issue: Traceability back and forth among phases
- Do we need architecture everywhere?
  - Issue: Representation & semantics of architecture in each phase
- Is Architectural change propagation cost-effective?
  - Issue: transforming from one phase to the next (manual, assisted, …)

The Mission Data System perspective

Phases



- The process methodology is flexible…
- … as long as State Analysis is applied throughout

- Subtle difference between:
  - Explicit architecture representation everywhere
  - Explicit architecture awareness
  - The former is "nice to have"
  - The latter is a pragmatic tradeoff for size & fit

State Analysis & SW Architecture

Similar Dimensions of Concerns	Code level	Component Architecture	System Architecture
Type information	- *procedural code* - *functions* - *classes (OO)*	- *methods* - *interfaces (sets of methods)* - *ports (interface signature)* - *components (sets of ports)* - *connectors (sets of ports)* - *hierarchical composition*	- *types* *(from state analysis)* *State variables* *Achievers, etc…* - *domain-specific types* *Units, etc…*
Instance information	- *variables* - *events* - *objects*	- *component instances* - *connector instances* - *links (pairs of port bindings)* - *hierarchical composition*	- *instances* *(from state analysis)*
Composition Mechanisms (how is it built?)	- *function calls* - *symbolic references/linkages* - *shared variables* - *ad-hoc runtime mechanisms*	- *prescription languages* *(requires type database)* *Base schema: xADL instances* *Extensible via XML schemas* *Compressible via transformations*	- *subject to lower-level mechanisms*
Description Mechanisms (what's inside?)	- *ad-hoc runtime mechanisms*	- *description languages* *(requires instance database)* *Base schema: xADL instances* *Extensible via XML schemas* *Compressible via transformations*	- *subject to lower-level mechanisms*

Architecture Composition in MDS

Analysis

- xADL extension for component/connector implementation inheritance

Design

- Separation of structure (defined in xADL) & implementation

Coding

- xADL extension for component/connector implementation inheritance
- Separation of structure (defined in xADL) & implementation
- Architecture profiling for optimizing transformations of xADL to code

Compile

- Packaging of architecture elements into shared objects

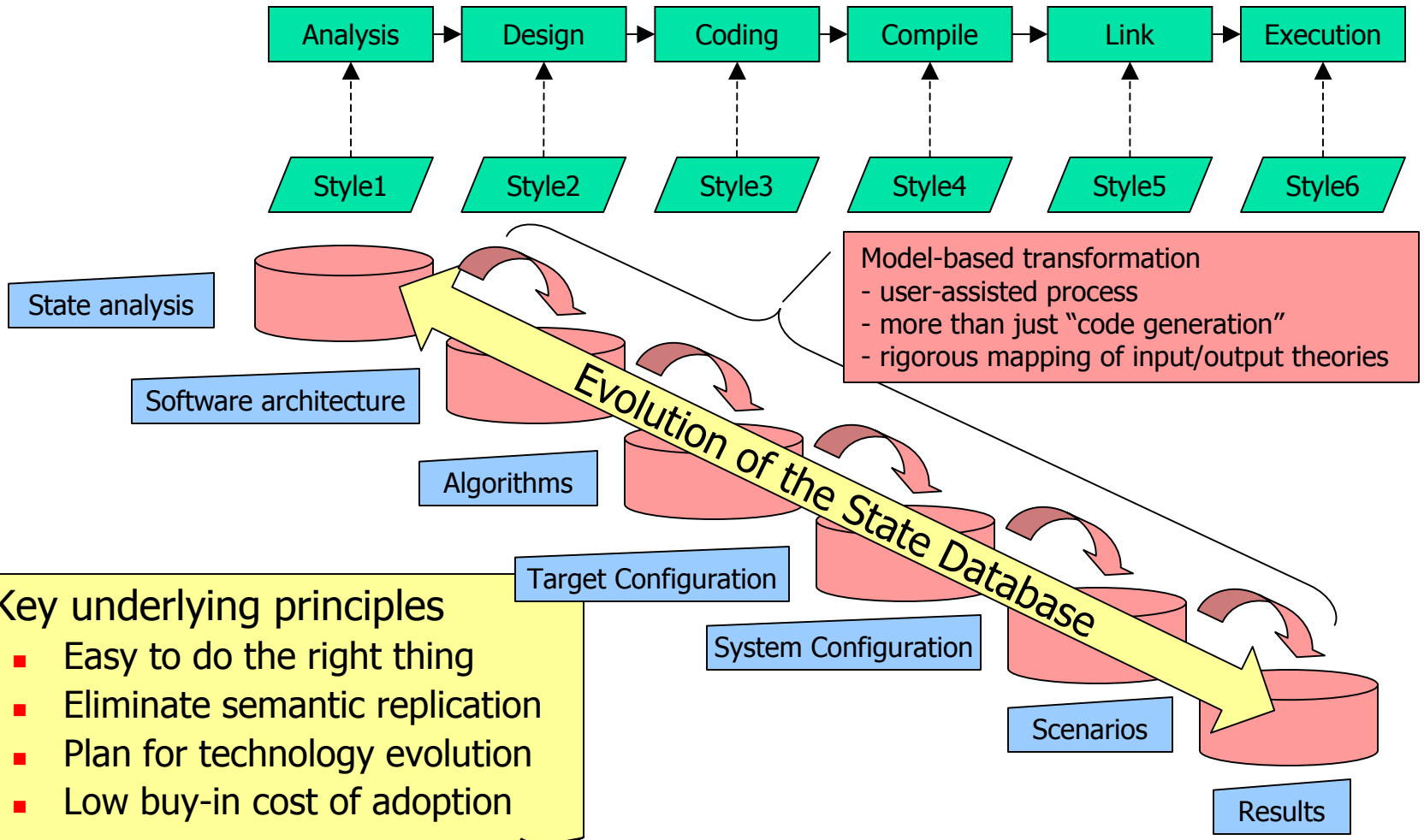
Link

- Dynamic registration of architecture elements at shared object init/fini

Execution

- Extensible prescription protocols support connector optimizations
- Architecture evolution includes types & instances
  - Type reconfiguration via dynamic object loading/unloading
  - Instance reconfiguration via prescription changes

The architecture waterflow



- Key underlying principles
  - Easy to do the right thing
  - Eliminate semantic replication
  - Plan for technology evolution
  - Low buy-in cost of adoption