Semantics of ARIS Model

Jon Atle Gulla

An analysis of the ARIS modeling language with respect to:
- conceptual foundation and
- formal properties

van der Aalst, W. M. P.: Formalization and Verification of Event-Driven Process Chains

Semantics of ARIS model

Why is Semantics Important?

Conceptual model is the bridge between:
- customer and project
- domain experts and analysts
- analysts and designers/programmers
- people and CASE environments
- project and end-users

Why is Semantics Important?

Outline

- ARIS modeling
  - ARIS conceptual model
  - Interpretation of ARIS models
- Analysis with respect to ontological foundation
  - BWW ontological model
  - Mapping of ARIS model to BWW ontological model
- Formalization of ARIS process model (EPC) using Petri nets
  - Formalization of EPC process model
  - Petri nets
  - Mapping from EPC to Petri nets
  - Verification of EPCs

I. ARIS Conceptual Model

- Multi-view conceptual language for modeling business processes and organizational structures
  - Organizational view
  - Function view
  - Data view
  - Control view
  - (Output view)
- Pre-defined models for ERP functionality
  - Low-level processes correspond to specific transactions in ERP system
  - Organizational structures correspond to structural elements in ERP system

I. ARIS Conceptual Model

ARIS Organization view

- Model of organizational units and positions/employees that are important for reporting purposes, areas of responsibilities and user profiles in ERP system

ARIS Function view

- Hierarchical list of manual and automated functions included in the business processes

ARIS and ERP Systems
ARIS and ERP Systems

Interpretation of ARIS Models

- How to analyze semantic aspects of a conceptual model?
  ✔ Can map model to (other) formal models
  ✔ Can define model with respect to mathematical or logical theory
  ✔ Can map model to ontological theory of information systems
  ✔ Mapping to computerized system introduces implementation details that do not belong to conceptual realm

BWW Ontological Constructs

- Ontological constructs for describing deep structure phenomena of information systems:

II. BWW Ontological Model

- Introduced by Bunge, Wand and Weber

- Nature of ontological model:
  ✔ Deep structure of information systems comprises those properties that manifest the meaning of the underlying real-world system
  ✔ Ontological model describe and predict the impact of deep structure phenomena

- Components of ontological model:
  ✔ Representation model
  ✔ State-tracking model
  ✔ Good decomposition model

- Purpose of ontological model:
  ✔ Study of internal structures of information systems that are independent of implementation decisions
  ✔ Theoretical foundation for designing and evaluating modeling languages
  ✔ Comparison of conceptual modeling languages

Ontological Analysis of ARIS

- Table showing analysis of ARIS model components.
Ontological Weaknesses of ARIS

- No representation for conceivable state space, lawful state space, conceivable event space, or lawful event space
- Sufficient focus on identifying all important state and transformation laws may not be present
- No representation of system, system environment, system structure, history, acts on, unstable state, and poorly defined event
- Defining scope and boundaries of system may be difficult
- Decomposition not sufficiently supported
- Function view ontologically redundant

III. Formalization with Petri nets

- Event-driven Process Chains (EPC)
  - Process model (control view) of ARIS
  - Used to describe business processes
- Formalization of EPC syntax
  - Mapping EPCs onto Petri nets
    - Provides a formalization of EPC
    - Defines the semantics of EPC
    - Enables checks for consistency, completeness, soundness, etc.

The Structure of EPC Models

Formalization of EPC Syntax

- EPC = (E, F, C, T, A):
  - E is a finite set of events
  - F is a finite set of functions
  - C is a finite set of logical connectors
  - T is a function which maps each connector onto a connector type
  - A is a set of arcs

Syntactic properties:
- The sets E, F, and C are pairwise disjoint
- For each e ∈ E: |•e| ≤ 1 and |e•| ≤ 1
- There is at least one start event and one final event
- For each f ∈ F: |•f| = 1 and |f•| = 1
- For each c ∈ C: |•c| ≥ 1 and |c•| ≥ 1
- The graph induced by EPC is weakly connected
- C
c
  - The graph induced by EPC is weakly connected
  - C
c

Petri nets - Basics

- Petri nets are a tuple (P, T, F):
  - P is a finite set of places
  - T is a finite set of transitions (P T =)
  - F is a set of arcs (flow relations)
- Petri nets have a mathematical foundation
- Formal semantics
- Automatic model analysis

Petri nets - Dynamics

- Tokens:
  - A place contains zero or more tokens
- Firing rule:
  - A transition τ is said to be enabled if each input place p of τ contains at least one token
  - An enabled transition may fire. If transition τ fires, then it consumes one token from each input place p of and produces one token in each output place p of τ
- State M:
  - Distribution of tokens over places
  - M

Example: State M = p1 = 2p3
- Transition τ1 is enabled, τ3 is not enabled
Formalization of EPC Semantics

- Defining EPC semantics in terms of Petri nets
- Basic mapping of EPC to Petri nets:
  - Place
  - Transition

- Mapping of connectors more complex
  - Depends on type of connector
  - Depends on events and/or functions linked to connector

Mapping from EPC to Petri nets

Verifications of EPC Models

- Regular:
  - An EPC is regular if
  - EPC has two special events: start and end
  - Every node in N is on a path from $s_{start}$ to $s_{end}$

- Sound:
  - A regular EPC is sound if
  - For every state $M$ reachable from the initial state, there exists a firing sequence leading from state $M$ to the final state
  - The final state is the only state reachable from the initial state where $s_{end}$ holds
  - There are no dead functions

- Well-structured
  - An EPC is well-structured if for any pair of connectors $c_1$ and $c_2$ such that one of the nodes is in CON and the other in CR,
  - For any pair of elementary paths $p_1$ and $p_2$, leading from $c_1$ to $c_2$ and from $c_2$ to $c_1$, $p_1 = p_2$

An erroneous EPC (1)

Model not sound:
- Assume that no billing needed event holds.
- Produce article will never be fired

ARIS and ERP Systems
An erroneous EPC (2)

Two models to the right are not well-structured

Conclusions

- ARIS rooted in informal modeling languages
- Semantic analysis of ARIS
  - BWB ontological model
  - Evaluation of modeling language with respect to deep structure of information systems
  - Petri nets
  - Formalization of EPC semantics by mapping EPC diagrams to Petri nets