6th International Workshop on Semantic Business Process Management

Service Adaptation Recommender in the Event Marketplace: Conceptual View

Yiannis Verginadis
Ioannis Patiniotakis
Nikos Papageorgiou
Roland Stuehmer

Information Management Unit
Institute of Communication and Computer Systems
National Technical University of Athens

FZI Forschungszentrum Informatik, Karlsruhe, Germany

Overview of presentation

- Vision
- Conceptual Architecture
- Service Adaptation Recommender (SAR)
- Situation-Action-Networks (SANs)

Motivating example

*Paul is a businessman who **has been flying** from Paris to New York. He **used the entertainment service** on board, but **hasn't finished watching** the movie before the landing. Two hours later he **is entering his room** in the downtown hotel he booked earlier and wow: the room entertainment **service is ready to PLAY** the movie Paul was watching in the plane – of course only the unfinished part.*

Motivating example

*Paul is a businessman who **has be**[illegible]**ng** from Paris to New York. He **used th**[illegible] **entertainment ser**[illegible] board, but **hasn't fini**[illegible]**l watching** th[illegible]movie before the lan[illegible]wo hours later he **is er**[illegible]**ring his room** in the downtown hotel he [illegible]ked earlier and wow: the room entertainment **service is ready to PLAY** the movie Paul was watching in the plane – of course only the unfinished part.*

Vision

- To develop and validate an **elastic** and **reliable** architecture for **dynamic and complex**, **event-driven interaction** in large highly **distributed** and **heterogeneous** service systems.
- Such an architecture will enable **ubiquitous exchange of information** between heterogeneous services, providing the possibilities **to adapt and personalize** their execution, resulting in the so-called **situational-driven adaptivity**

PLAY

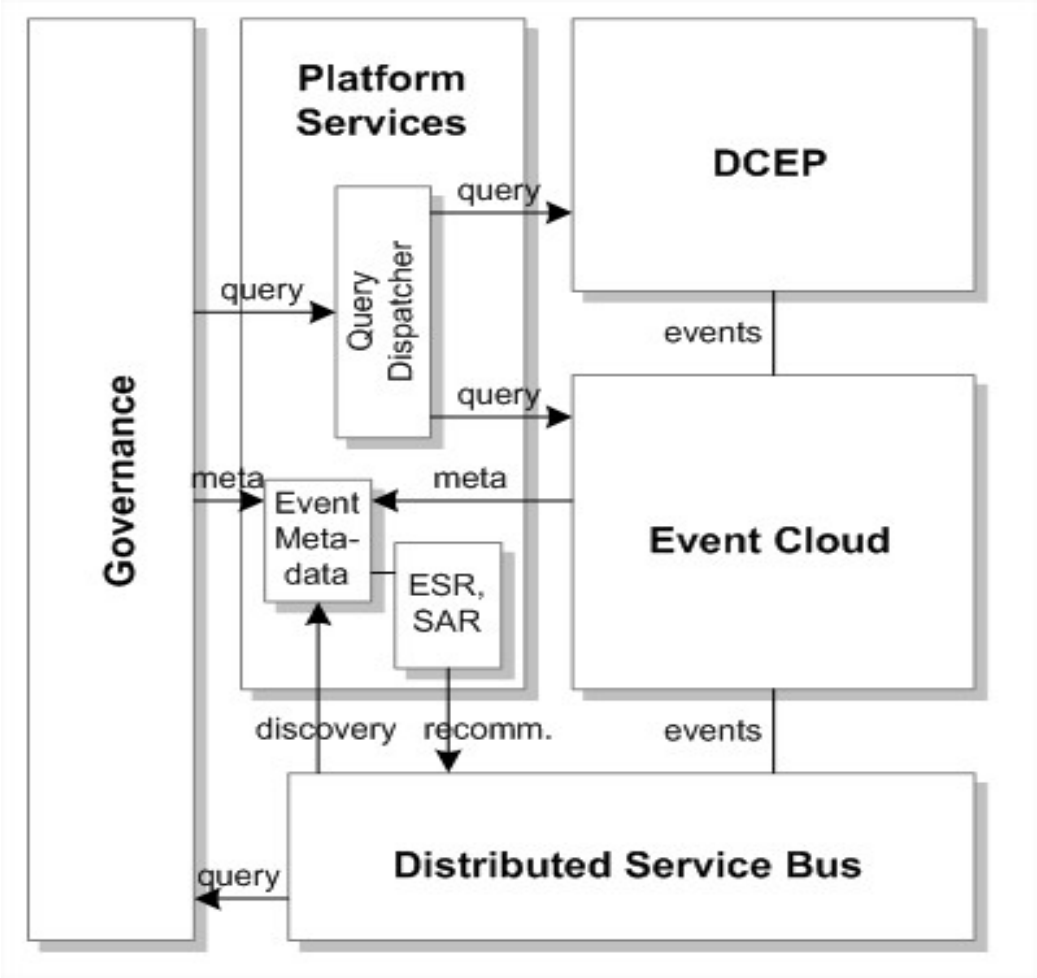
SEVENTH FRAMEWORK PROGRAMME

PLAY project Core Message
PLAY will revolutionize the way People, Things and Services will cooperate and coexist in the Future Internet, by introducing ubiquity in the communication and proactivity in handling large scale distributed environments.

Overview of presentation

- Vision
- Conceptual Architecture
- Service Adaptation Recommender (SAR)
- Situation-Action-Networks (SANs)

Conceptual Architecture



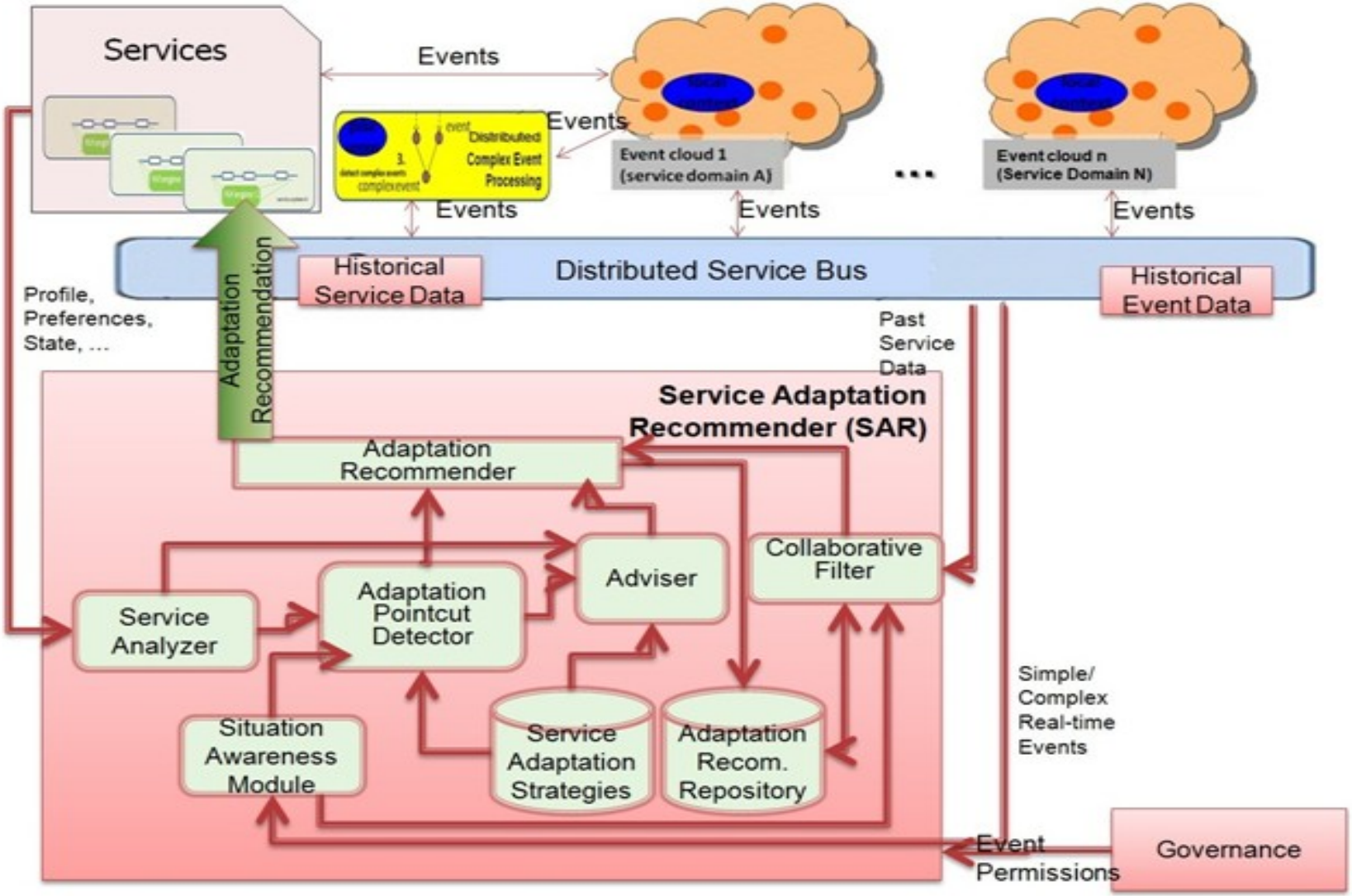
Overview of presentation

- Vision
- Conceptual Architecture
- Service Adaptation Recommender (SAR)
- Situation-Action-Networks (SANs)

Service Adaptation Recommender (SAR)

- The objective of SAR is:
  - to suggest service administrators, changes (adaptations) of their services' configurations, composition or workflows, in order to overcome problems or achieve higher performance.
- Based on recognized situations, SAR will be able to:
  - define adaptation pointcuts (points in a service flow that need to be adapted as a reaction to a certain situation) and
  - advices (what to adapt and how)

SAR: Conceptual View



Overview of presentation

- Vision
- Conceptual Architecture
- Service Adaptation Recommender (SAR)
- Situation-Action-Networks (SANs)

Motivation

- We can not predefine everything in advance (at design time)
  - But we can predefine some general **Goals** in advance
- We need our system to react to dangerous/interesting **situations**
- (re)**Actions** to the detected situations can not be fully mapped in advance
  - we need a dynamic service composition mechanism
- In order to react (or recommend reactions) successfully, **contextual** and **operational** information about user, events and services is needed
- We propose Situation Action Networks (SANs) to model this knowledge
  - SANs use concepts from Hierarchical Task Networks (HTNs) and Behavior Trees (BTs)

Related Work (1/5)

- Hierarchical Task Network Planning (1/2)
  - *High-level tasks are decomposed into simpler tasks until a sequence of primitive actions solving the high-level tasks is generated*.
  - A natural representation for many real-world domains, including military planning (Mitchell,1997), encoding strategies in computer games (Hoang et al., 2005), and manufacturing processes (Nau et al.,2005).
    - Nau, D., Au, T.-C., Ilghami, O., Kuter, U., Muñoz-Avila, H., Murdock, J. W., Wu, D., and Yaman, F. Applications of **SHOP** and **SHOP2**. IEEE Intelligent Systems 20(2). 2005
    - Mitchell, S.W. A hybrid architecture for real-time mixed-initiative planning and control. Proceedings of the IAAI-97. AAAI Press, 1997.
    - Hoang, H., Lee-Urban, S., and Muñoz-Avila, H. Hierarchical Plan Representations for Encoding Strategic Game AI. Proceedings of AIIDE-05. AAAI Press.

Related Work (2/5)

- Hierarchical Task Network Planning (2/2)
  - HTN planners (like all planners) model changes made to the environment, as they search for a sequence of actions that will achieve the desired goal. There is no input from the environment, during the planning process, and execution of the action sequence found is outside the domain of the planner. Consequently they rely on the environment being relatively stable. HTN planning can be categorized into partial-order and total order HTN planning.

Related Work (3/5)

- Dynamic or Reactive Planning
  - **CYPRESS** system, integrates a planning system with an execution system. It has the ability to react to the unanticipated changes in the environment by re-planning.
  - **RETSINA** system interleaves planning with execution and supports planning for dynamic and real environments.
  - **Repair-SHOP** is capable of performing plan adaptation and plan repair
    - D. Myers, L. Wesley, and A. Center. CYPRESS: Reacting and Planning under Uncertainty. In DARPA Proceedings: Rome Laboratory Planning Initiative, page 111. Morgan Kaufmann, 1994.
    - Paolucci, M.; Kalp, D.; Pannu, A. S.; Shehory, O.; and Sycara, K. A planning component for RETSINA agents. In Intelligent Agents VI. Springer. 1999.
    - Warfield, I.; Hogg, C.; Lee-Urban, S.; and Munoz-Avila, H. 2007. Adaptation of hierarchical task network plans. In Proceedings of the Twentieth International FLAIRS Conference (FLAIRS-07).

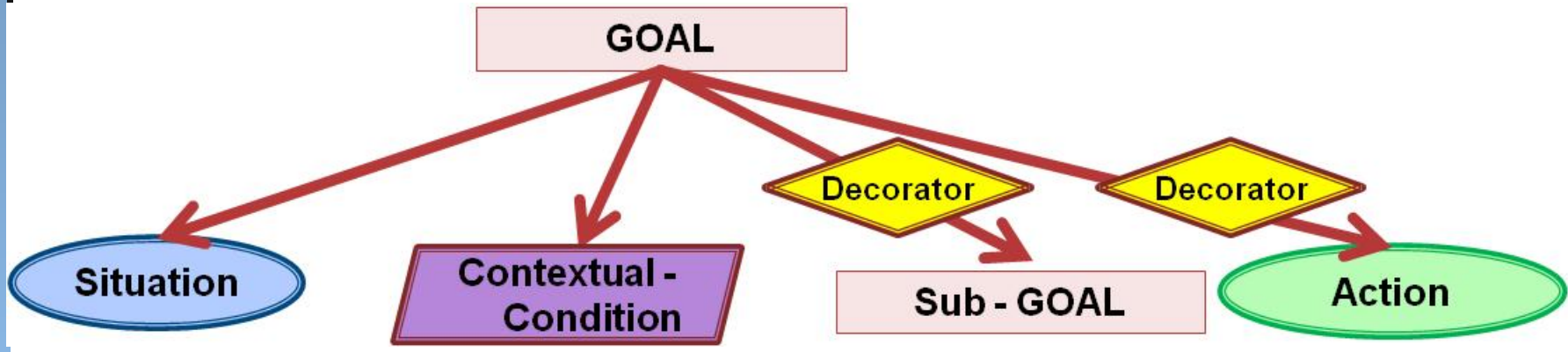
Related Work (4/5)

- Behavior Trees in Behavior Engineering
  - *A Behavior Tree is a formal, tree-like graphical form that represents behavior of individual or networks of entities which realize or change states, make decisions, respond-to/cause events, and interact by exchanging information and/or passing control*
  - Behavior Engineering is an integrated discipline that supports the systems and software engineering of large-scale, dependable software-intensive systems
    - Dromey, R. G. "Formalizing the Transition from Requirements to Design", in "Mathematical Frameworks for Component Software - Models for Analysis and Synthesis", Jifeng He, and Zhiming Liu (Eds.), World Scientific Series on Component-Based Development, pp. 156-187, (Invited Chapter) (2006)
    - Colvin, R., Grunske, L. and Winter. K., "Timed Behavior Trees for Failure Mode and Effects Analysis of Time-Critical Systems", in Journal of Systems and Software, Elsevier, accepted for publication 03/2008.
    - Myers, T., Fritzson, P., Dromey, G., "Co-Modeling: From Requirements to an Integrated Software/Hardware Model" , Computer, 09 Sept. 2010. IEEE computer Society Digital Library. IEEE Computer Society, http://doi.ieeecomputersociety.org/10.1109/MC.2010.270

Related Work (5/5)

- Behavior Trees in Game AI
  - Mehta, M. and A. Ram (2009). "Runtime Behavior Adaptation for Real-Time Interactive Games." Computational Intelligence and AI in Games, IEEE Transactions on **1(3): 187-199.**
  - Lee, S., P. Holme, et al. (2011). "Emergent hierarchical structures in multiadaptive games." Physical Review Letters **106(2): 028702.**
  - Palma, R., P. A. Gonz lez-Calero, et al. (2011). Extending Case-Based Planning with Behavior Trees. Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference.
  - Lim, C. U., R. Baumgarten, et al. (2010). "Evolving Behaviour Trees for the Commercial Game DEFCON." Applications of Evolutionary Computation: 100-110.
  - Skorupski, J. and M. Mateas (2010). "Novice-friendly Authoring of Plan-based Interactive Storyboards."
  - Michael Mateas, Andrew Stern, "A Behavior Language for Story-Based Believable Agents," IEEE Intelligent Systems, pp. 39-47, July/August, 2002

A Basic SAN



- Goal expresses **what** is pursued to be achieved by the SAN. It can be a descriptive tag or name.
- SAN is evaluated from left to right (i.e. Situation, Condition, Action)
- Situation branch “monitors” whether a specific situation occurs before continuing. Blocks while situation not occurs.
- Condition “checks” whether a condition concerning SBA context holds. Condition is a boolean expression.
- Action is taken if situation occurs and also condition is true

Goals in SANs

GOAL

- Goals are “tags” or “brief declarations of objectives”
- By evaluating the “Root Goal” of a SAN the whole SAN is evaluated

- A Research Challenge !!
  - Dynamic generation of Goal Branches during runtime, i.e. when Goal needs to be evaluated

Situations in SANs

- Situations are modeled as CEPATs and upon their detection we know that the situation occurred
- Situations can be **Abstract**

- Challenge : Dynamic generation of CEPATs

Conditions in SANs

Contextual Condition

- Conditions (or Preconditions) are boolean expressions
- They are evaluated second, after the Situation occurs
- They can use contextual information for their evaluation
- Conditions can also :
  - Call functions or external APIs or other programmatic artifacts
  - Query datastores (e.g. Event Cloud, Databases, GIS's)

Actions in SANs

- Actions are tasks which are executed when the situation occurs and also the (pre)condition is true
- Actions can be :
  - Composite tasks in the form of “Sub-Goal” nodes, or
  - **Abstract Actions** defined as a “Pool” of possible actions
  - Primitive tasks
- Actions may not complete immediately but they can take a significant amount of time to finish
- Actions return a status, which can be **Success** or **Failure**
- The return status of the Action is also the return status of the parent Goal node

SANs Legend

means sequential execution of actions / sub-goals.
Successful termination if all actions are successful too

// <any>

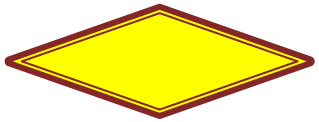
means parallel execution of actions / sub-goals.
Successful termination if ANY of the actions is successful

// <all>

means parallel execution of actions / sub-goals.
Successful termination if ALL actions are successful

?

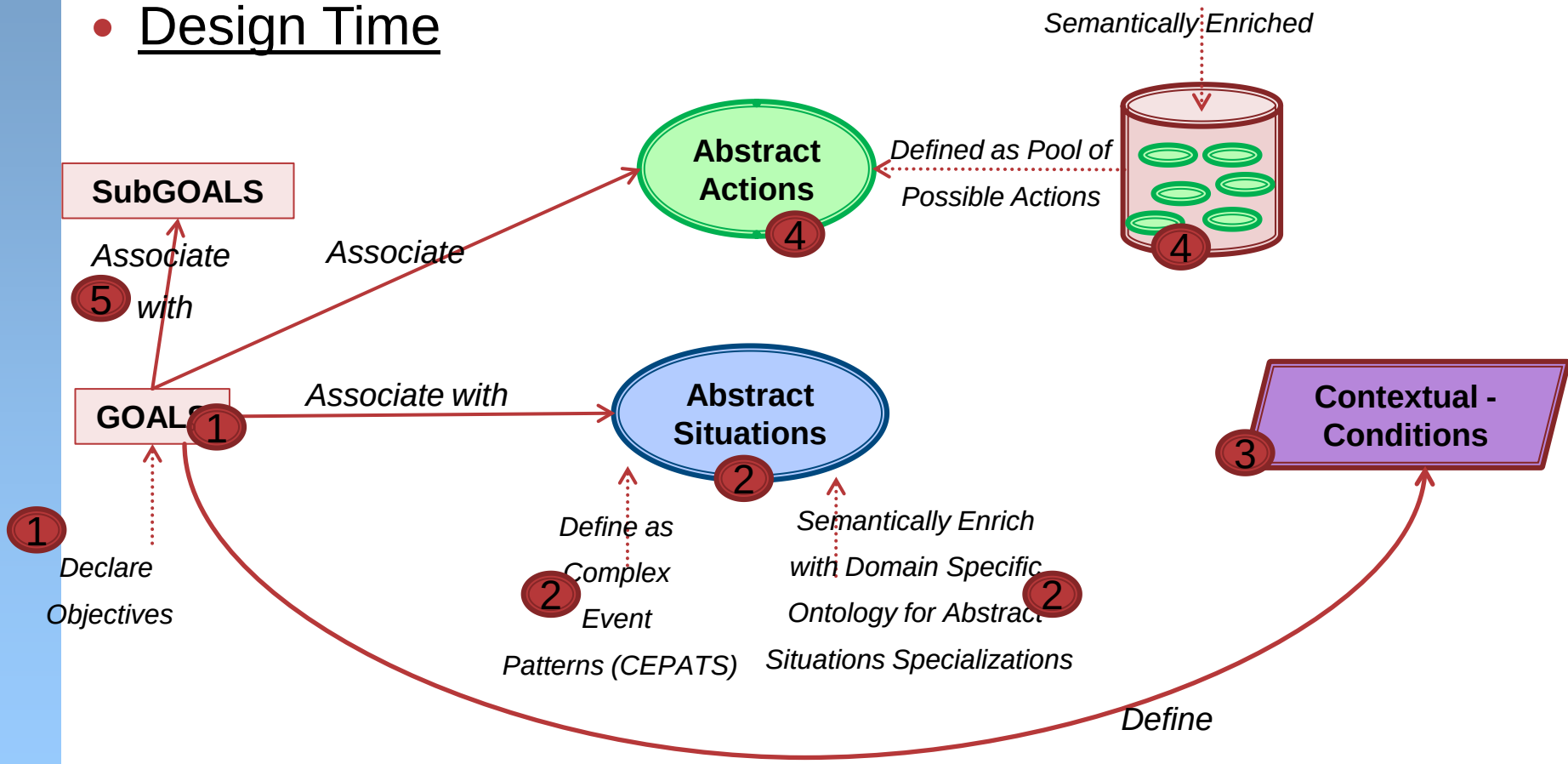
means sequential execution of actions / sub-goals.
Successful termination at the first successful action



Decorator. E.g. Loop, Counter, Timer.

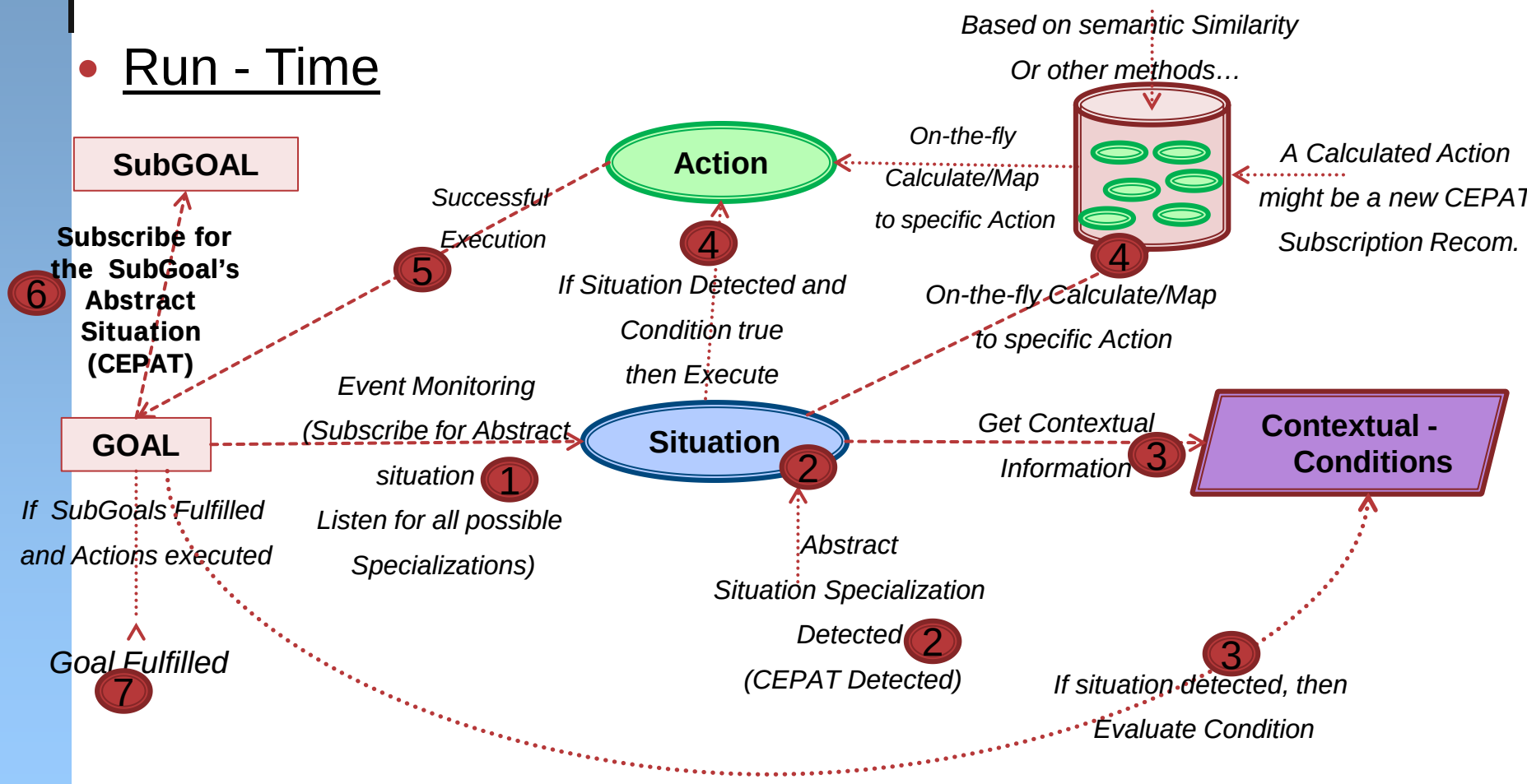
Life Cycle of a SAN (1/2)

- Design Time

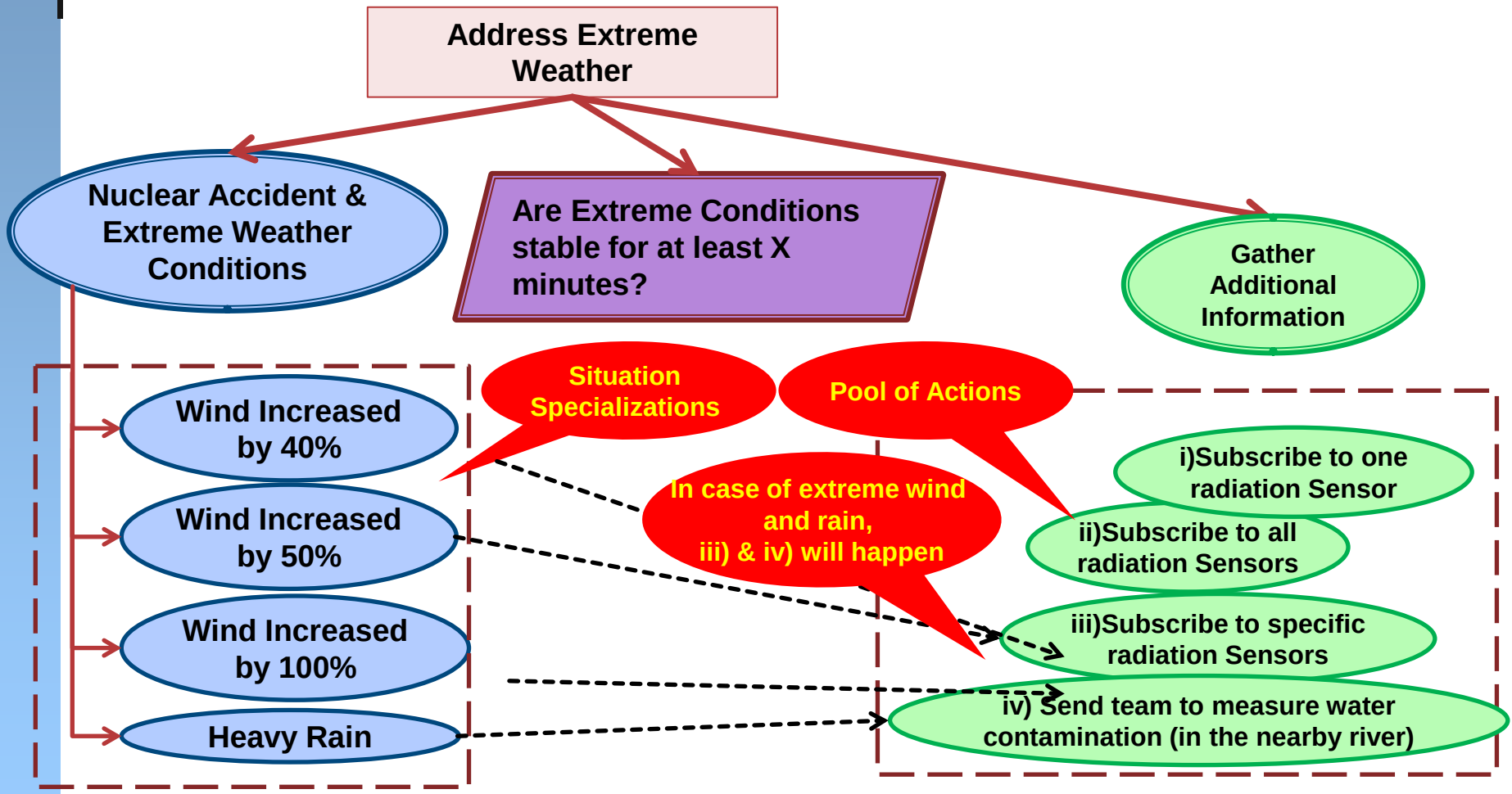


Life Cycle of a SAN (2/2)

- Run - Time



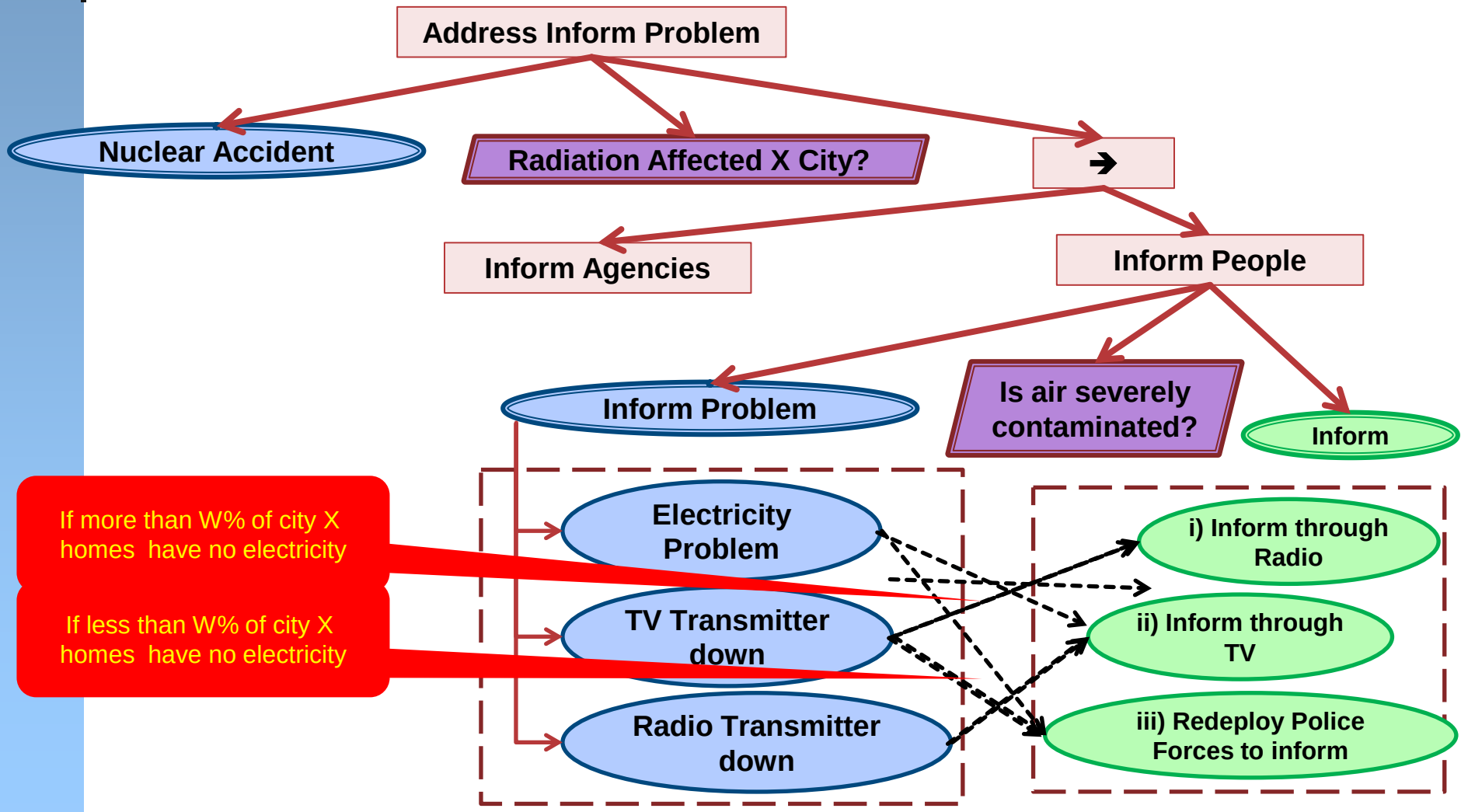
Nuclear Accident Example - "Address Extreme Weather" SAN



iii) Location ➔ opposite of wind direction (e.g. if North-West wind, then location= South-East),

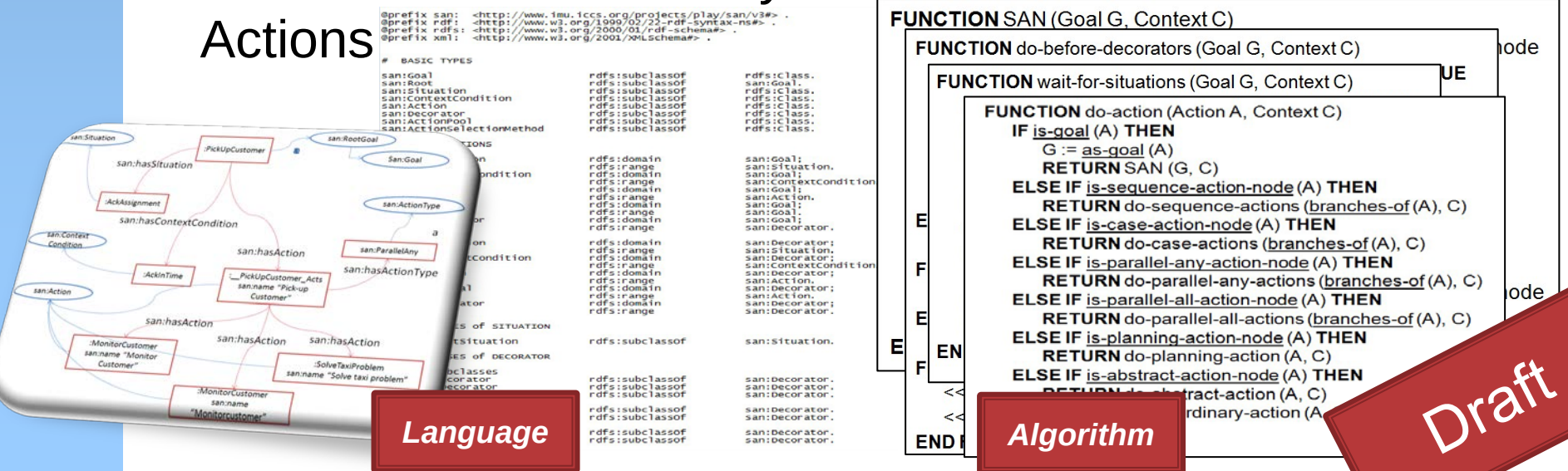
Distance ➔ 0 to D, where D = ((wind-speed * X minutes)/60) +/- 1, without/with rain (example formula, e.g. wind_speed = 8 Beaufort ≈ 65 Km/h, D= 6,5 Km in sunny conditions)

Nuclear Accident Example -“Address Inform Problem” SAN



Ongoing Work

- Model SANs with an RDF-based language
- Develop an Execution Algorithm for SANs
- Development of a proof-of-concept prototype of a SAN execution engine
- Semantic-based situation detection with ontologies
  - Reasoning with CEP Engine
- Mechanism for on-the-fly Mapping of Situations and Actions



Thank you for your attention!!!