IEEE Communications Society
Distinguished Lecture

Introduction to Cognitive Situation Management for Tactical Operations

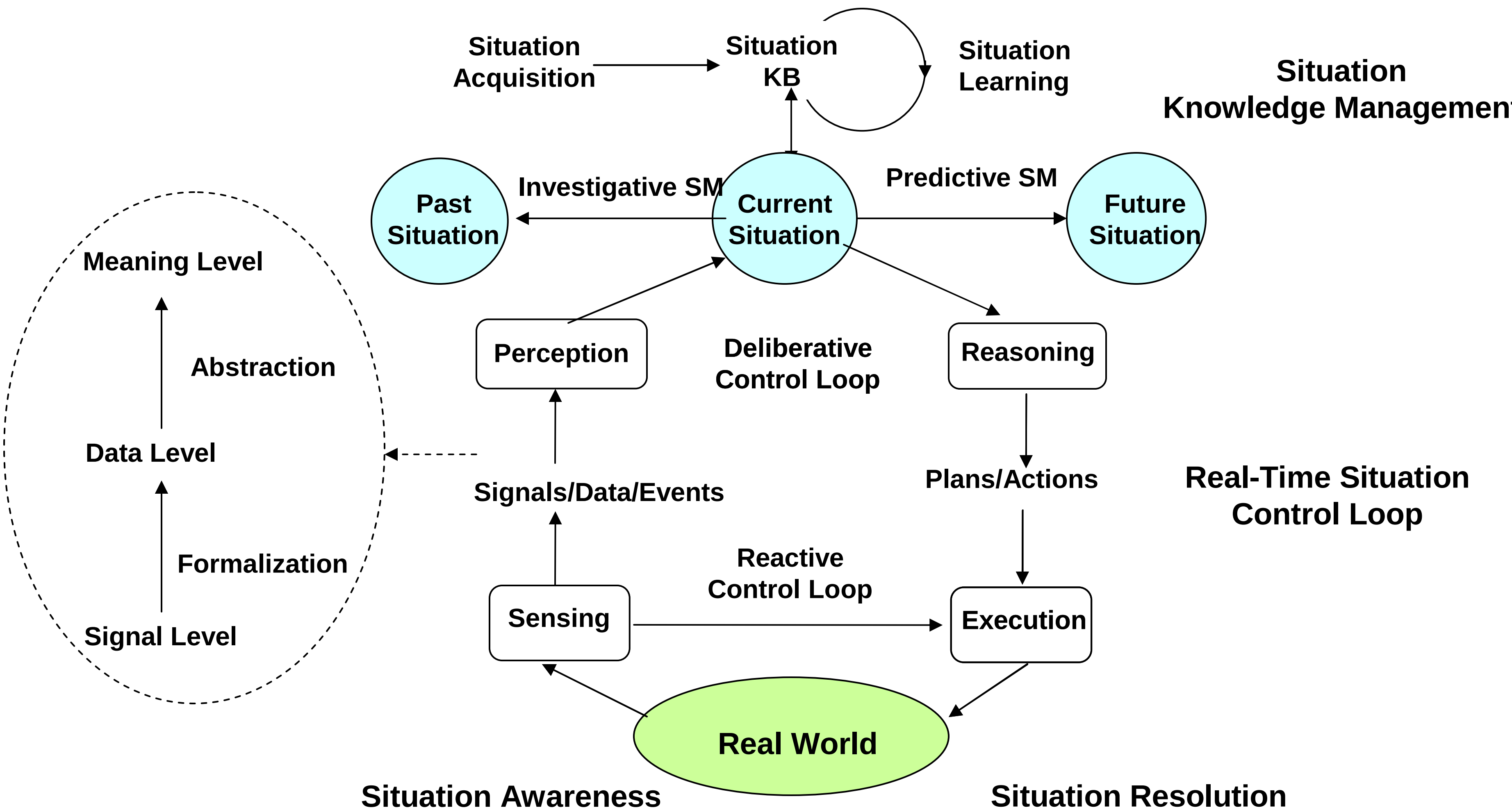
Dr. Gabriel Jakobson
Chief Scientist, Altusys Corp, Princeton, NJ, USA

Oulu, Finland, June 23rd, 2008

Introduction

- **Situation Management (SM) is a synergistic goal-directed process of <u>analysis, control, and prediction</u> of situations happening in dynamic systems and operational spaces.**
- **Informally, situations are seen as <u>states of a dynamic system</u> observed at particular time; Complexity of the situations may range from a single attribute value of an object, or a single relation among the attribute values, to complex collections of objects interlinked by various class, structural, spatial, temporal, and other (domain-specific) relations**
- **Research in situations, situational behavior of humans, and in modeling the acts of situation resolution have been in the focus of various disciplines such as Situation Awareness, Situation Calculus, Information Fusion and Human Factors**
- **In this lecture we will present basic concepts associated with situation management and see how they are applied for the tactical operations domain**

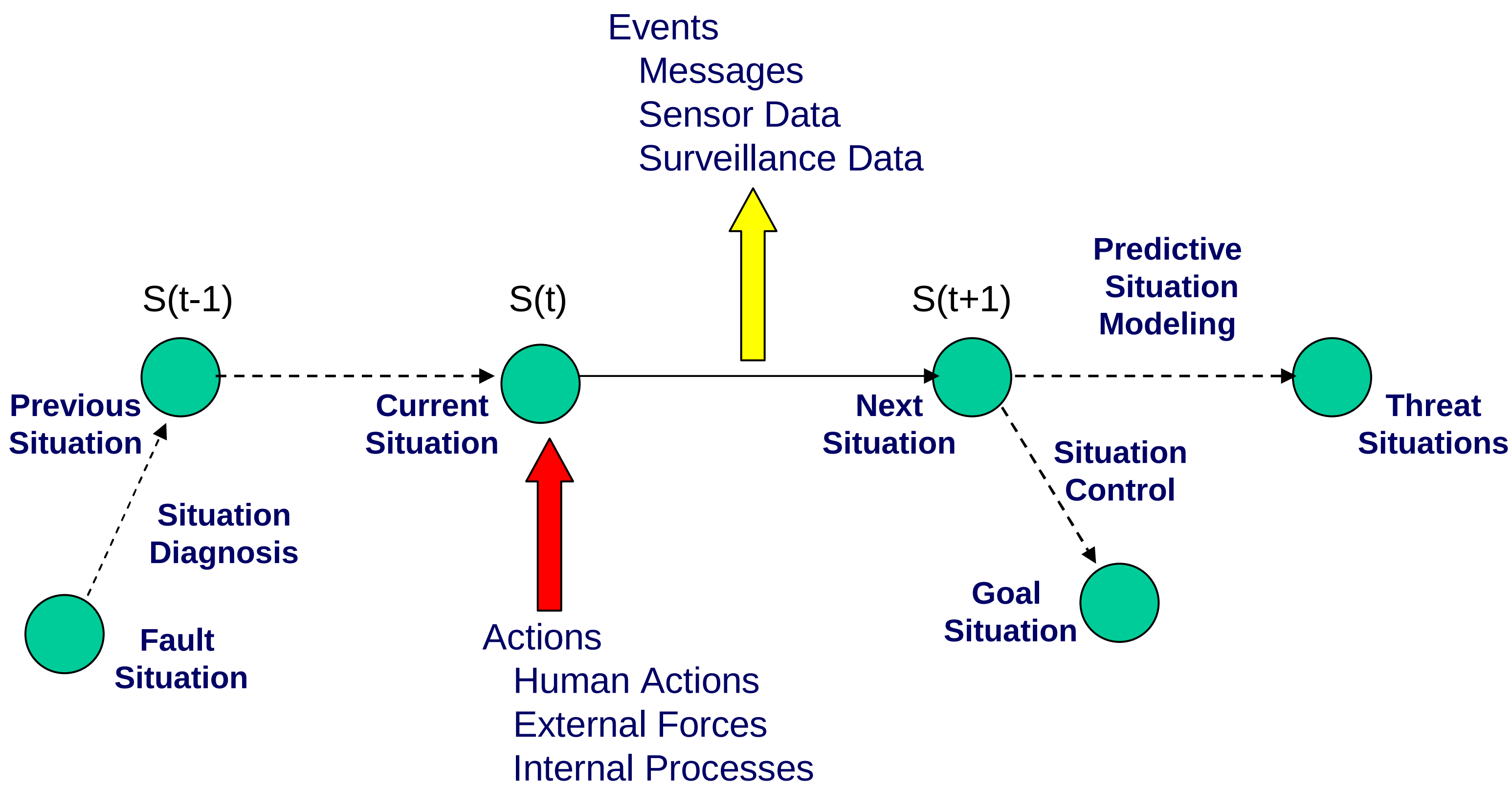
Situation Management : A Reference Diagram



The Types of Situation Management

- We identify three basic types of situation management:
  - investigative
  - control
  - predictive type
- The investigative SM is concerned with a retrospective analysis of causal situations which determine why a certain situation happened. The control type of SM aims to change or keep the current situation, while the predictive type of SM aims to project possible future situations.
- For example, finding a root of a packet transmission failure in a telecommunication network is an example of an investigative SM; moving a tank unit from the area of direct hostile fire is a control type SM; and a projection of a potential terrorist attack on a critical infrastructure element is an example of a predictive SM.

Situation Transitions



The Scope of Issues of SM

Situation Modelling, Awareness and Decision Support

- Situation monitoring and awareness
- Situation modeling, reasoning and decision support
- Predictive situation modeling
- Situation ontology and semantics
- Formal methods and situation calculus
- Learning and situation discovery

Intelligent Sensing

- Intelligent sensor networks
- Semantic sensor web
- Biology-inspired, autonomic and self-organizing sensing

Information Fusion and Event Correlation

- Multi-source, multi-resolution and hierarchical information fusion
- High-level models of information fusion
- Information fusion and situation awareness
- Event correlation algorithms
- Event-driven systems

Situation Management Architectures and Platforms

- Multi-agent systems and distributed situation management
- Collaborative models of situation management
- Peer-to-peer architectures for situation awareness
- Integration with Service-Oriented Architectures (SOA) and enterprise event processors
- Situation management platforms

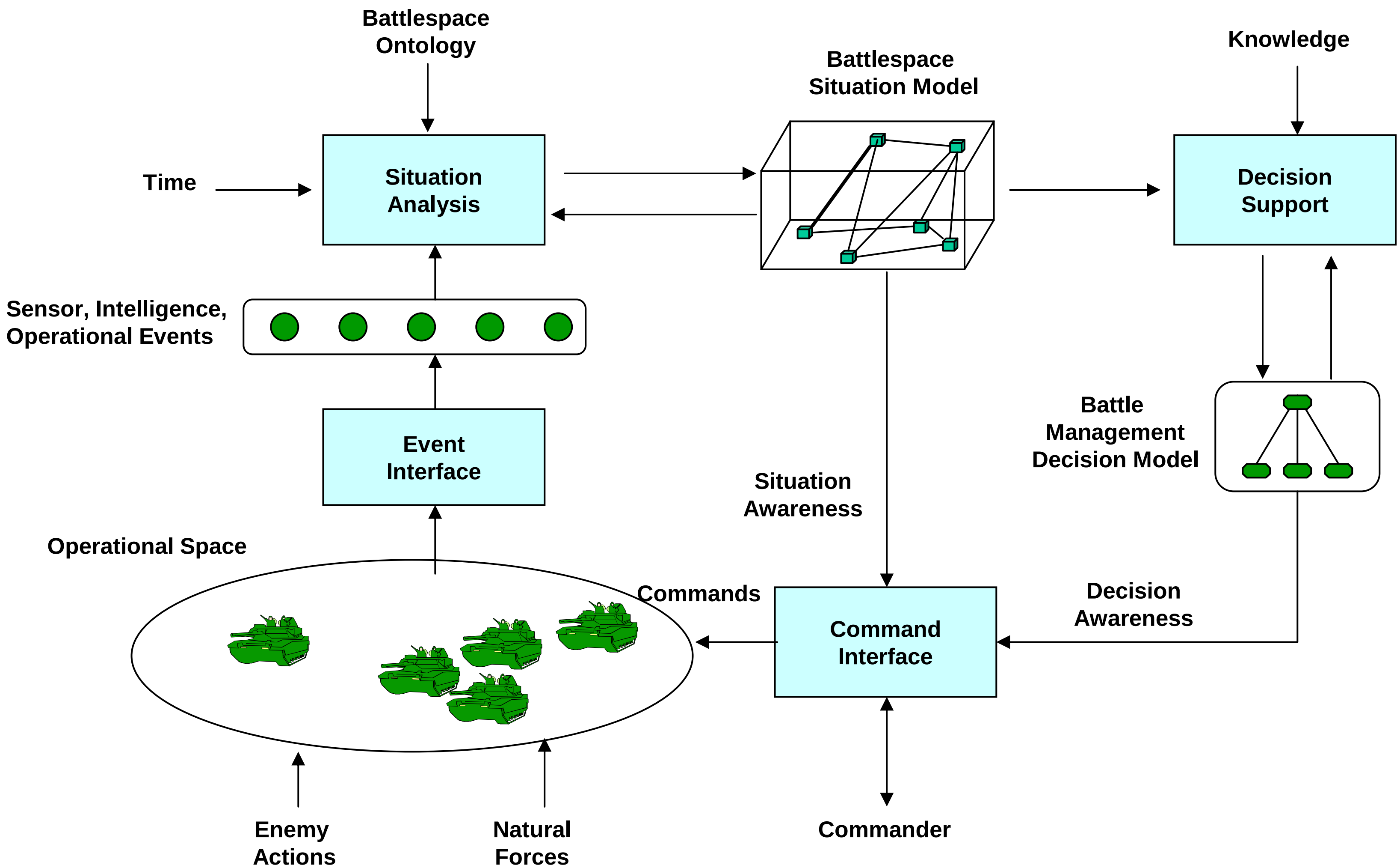
Situation Management Applications and Experiences

- Situation management in communication networks
- Asymmetric, net-centric and tactical battlespace operations
- Infrastructure and cyber security
- Earth observations, disaster response and crisis management
- Intelligent transportation, health care, and enterprise management networks

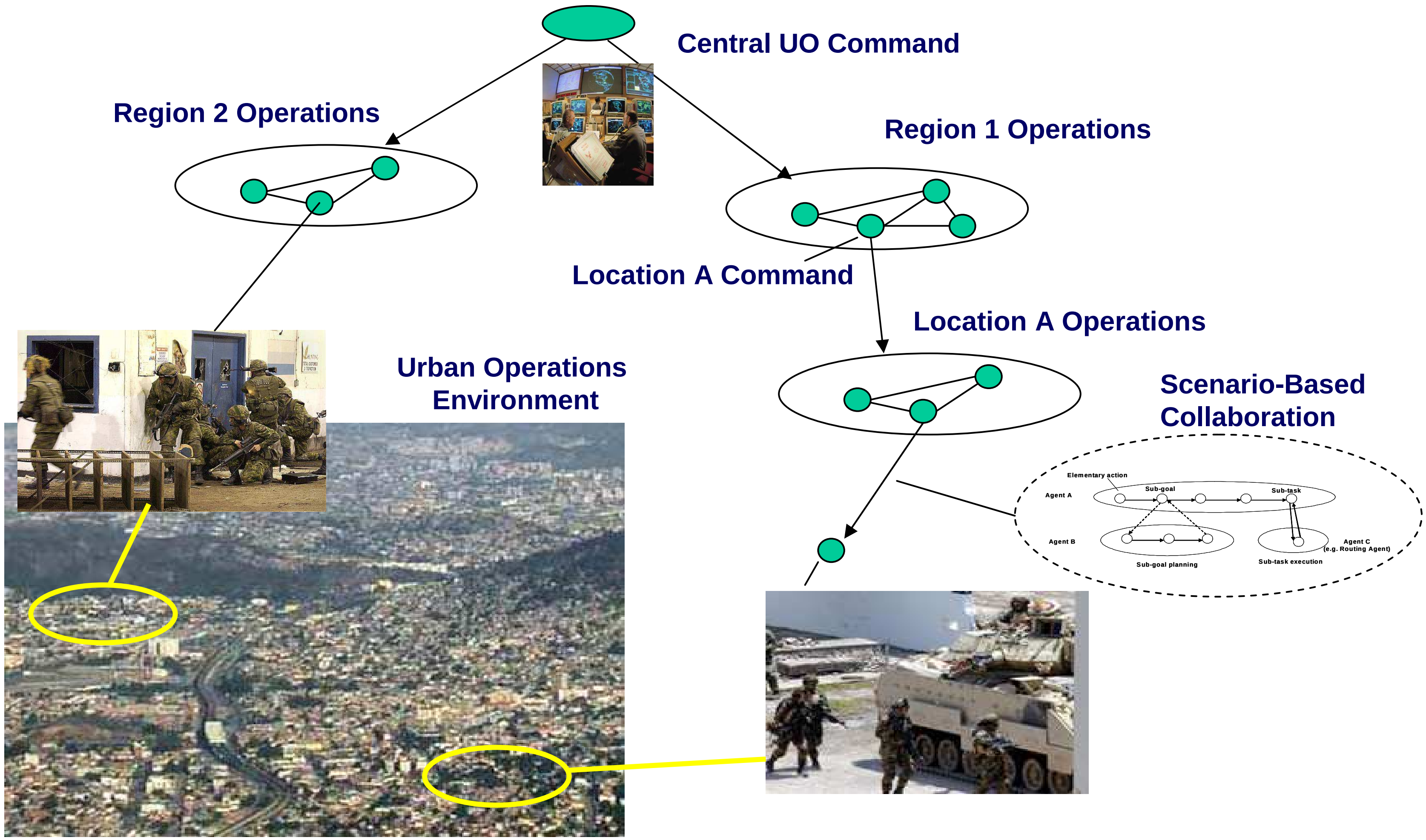
Tactical Situation Management: Characteristics and Operational Needs

- Characteristics of the future battlefield
  - High mobility of troops and weapon systems
  - Increasing operational tempo
  - Unpredictable battlespace situations
  - Convergence of real and cyber battlespace
- Operational needs from military commanders' perspective
  - Effective situation awareness
  - Recognition of emerging trends
  - Potential threat warnings
  - Decision awareness and command options

Tactical Situation Management

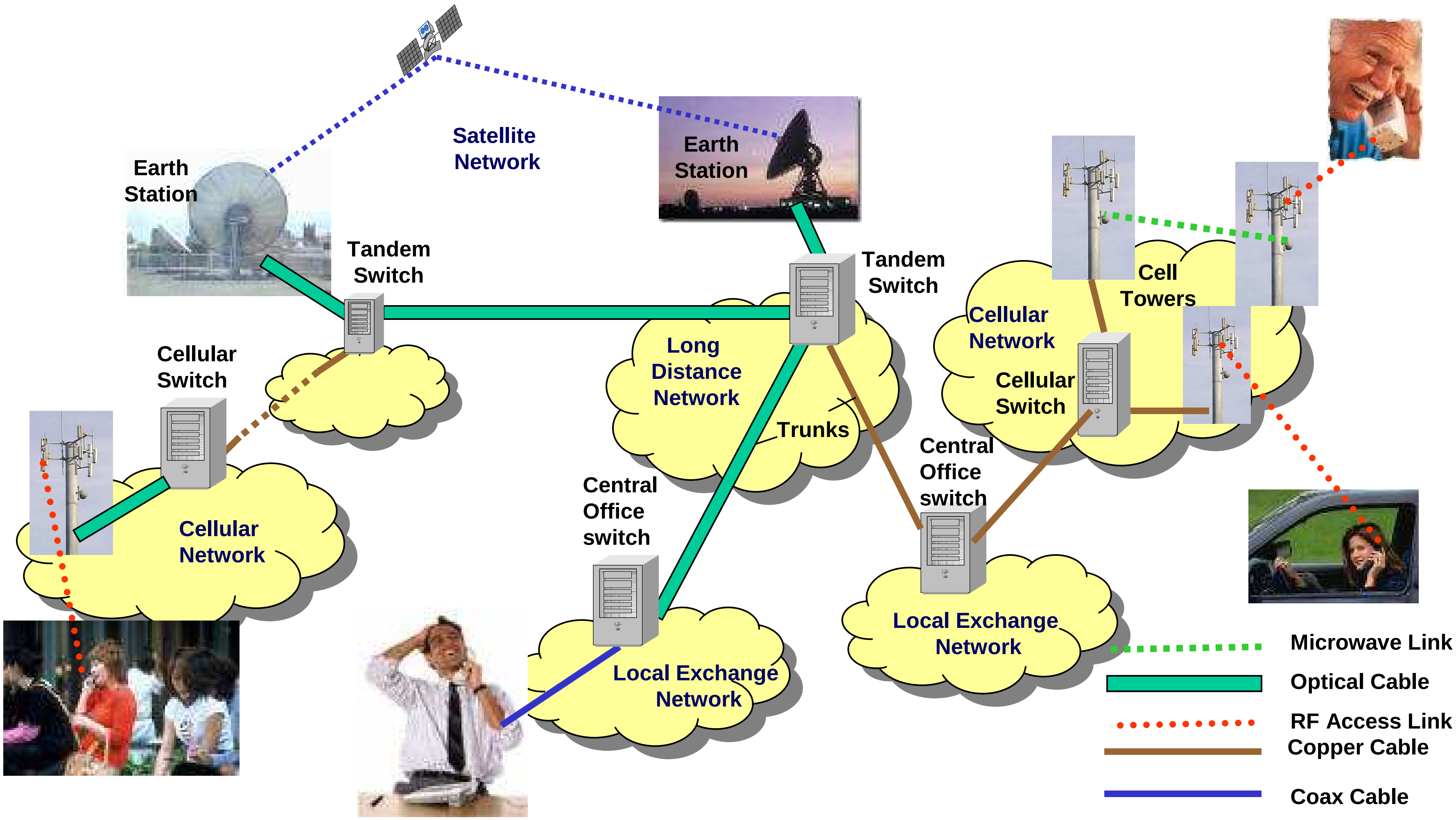


Urban Operations

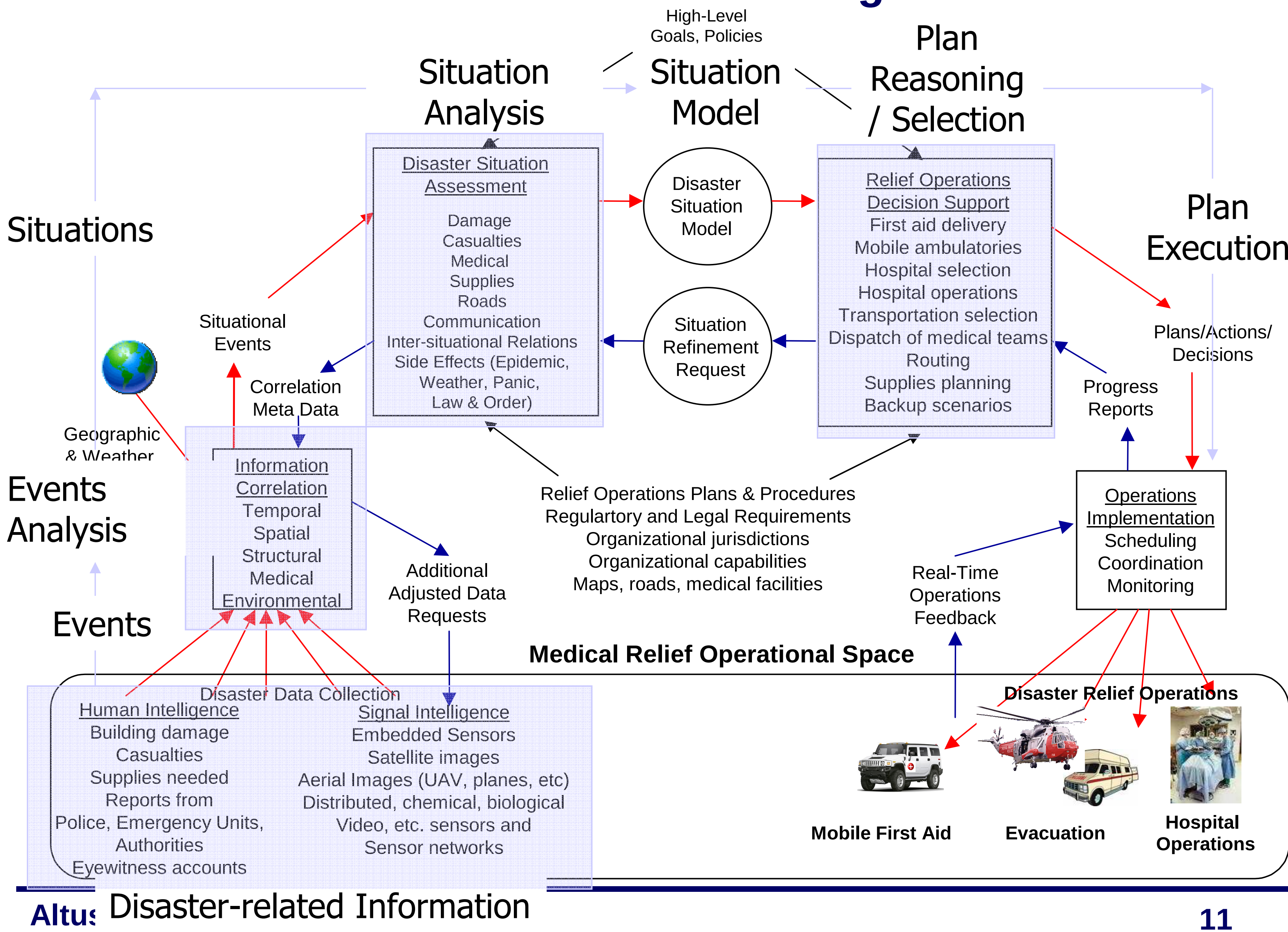


L. Lewis, G. Jakobson, J. Buford, "Inferring Threats in Urban Environments with Uncertain and Approximate Data: An Agent-Based Approach," ***Applied Intelligence,*** **2008.**

Telecommunication Network Management



Disaster Situation Management



The Scope of Issues of Situation Modeling and Resolution

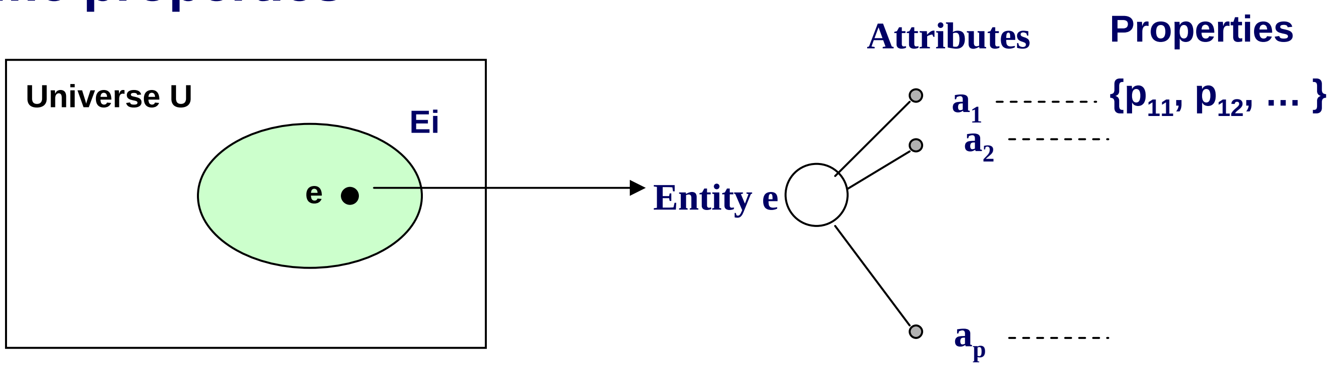
- World Model
- The Belief System, Ontology
- Situations, Events, Situation Transitions,
- Modeling of Time, Temporal Reasoning
- Goals, Plans, Actions
- Inexact and Approximate Models
- Deterministic and Statistical Approaches
- Performance and Scalability
- Distributed Management and Multi-Agent Systems
- Situation Management Languages, Tools and Platforms
- Situation Knowledge Acquisition and Engineering

Situation Modeling – General Framework

- **Structural Modeling**
  - Entities, attributes, attribute domains, constraints
  - Entity classes, class ontologies, core ontologies
  - Relations
- **Dynamic Modeling**
  - Situations
  - Events
  - Actions
  - Time
- **Representation**
  - Primary concept specification languages (set-theoretical, FSM)
  - Structural specification languages, e.g. OWL and DAML
  - Graphical modeling languages e.g. UML
  - Programming languages, e.g. SDL and GOLOG

Structural Domain Modeling

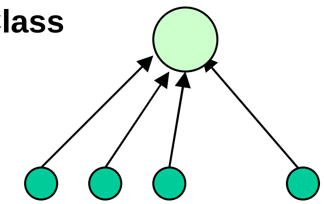
- Let's assume that there exists a universe U, real or abstract that could be sensed, perceived, reasoned and affected, and which is populated with entities $e \in E_i$, $E_i \subseteq U$
- An entity e is a thing of significance that has distinctive existence and is represented by set of attributes $\{a_1, a_2, \ldots, a_p\}$
- Each attribute is a collection of attribute properties, such as attribute name, type, value, default value, and other application-specific properties



- Attribute value is a triplet containing an actual attribute value, certainty estimation, and time, either a point or interval time during which the attribute holds its value.

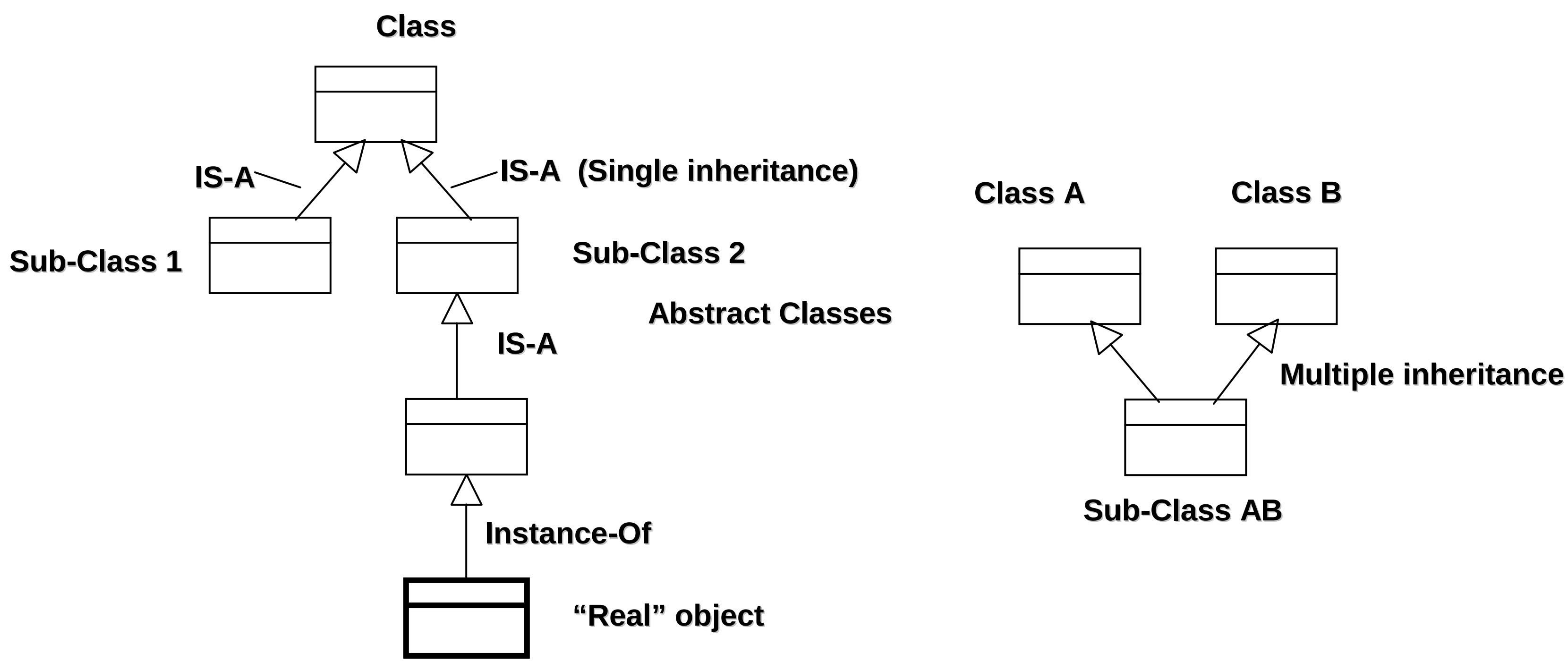
Structural Domain Modeling (2)

- A set of entities with certain common attributes defines an abstract entity class



- Some entities are active, they change their attributes and properties in time; Some of the entities can interact with other entities forming multi-entity systems
- We will consider entities as dynamic time-dependent objects with their time of creation t', time of clear t", and corresponding lifespan $\delta=(t', t")$. Any attribute value of an entity is defined only during the existence of the entity, i.e. $a(t),\ t \in \delta$

Classes, Sub-Classes and Instances



Sub-class inherits all the attributes and corresponding attribute properties from the (super-class), unless restrictions are applied to the inheritance mechanism. Multiple inheritance allows to create entities with aggregated features, however care should be taken to avoid conflicts

Reference - Grady Booch: *Object-Oriented Analysis and Design with Applications*, Addison-Wesley

Example: A Simplified Entity Class Specification of a M1-Abrams Tank

```
<DomainClass Name="M1-Abrams" Documentation="A class describing US Army M1 Abrams Main Battle Tank">
        <DomainClassParent>
                <DomainClassLink Name="Main-Battle-Tank"/>
        </DomainClassParent>
        <DomainClassLocation>
                <DCLocSlot Name="Tank-Location">
        </DomainClassLocation>
        <DomainClassTime>
                <DCUnivTimeSlot Name="Unit-Time">
        </DomainClassTime>
        <DomainClassSlots>
                <DCIntegerSlot Name="Combat-Weight" 54.5/>
                <DCIntegerSlot Name="Maximum-Speed" 45/>
                <DCIntegerSlot Name="Power-to-Weight-Ratio" 27/>
                <DCIntegerSlot Name="Total-Crew" 4/>
                <DCIntegerSlot Name="Length-of-Hull" 24.49/>
        </DomainClassSlots>
        <DomainClassMethods>
                DCDatabaseMethod SetValue "Tank-Location"
                DCDatabaseMethod GetValue "Tank-Location"
        </DomainClassMethods>
</DomainClass>
```

Entity

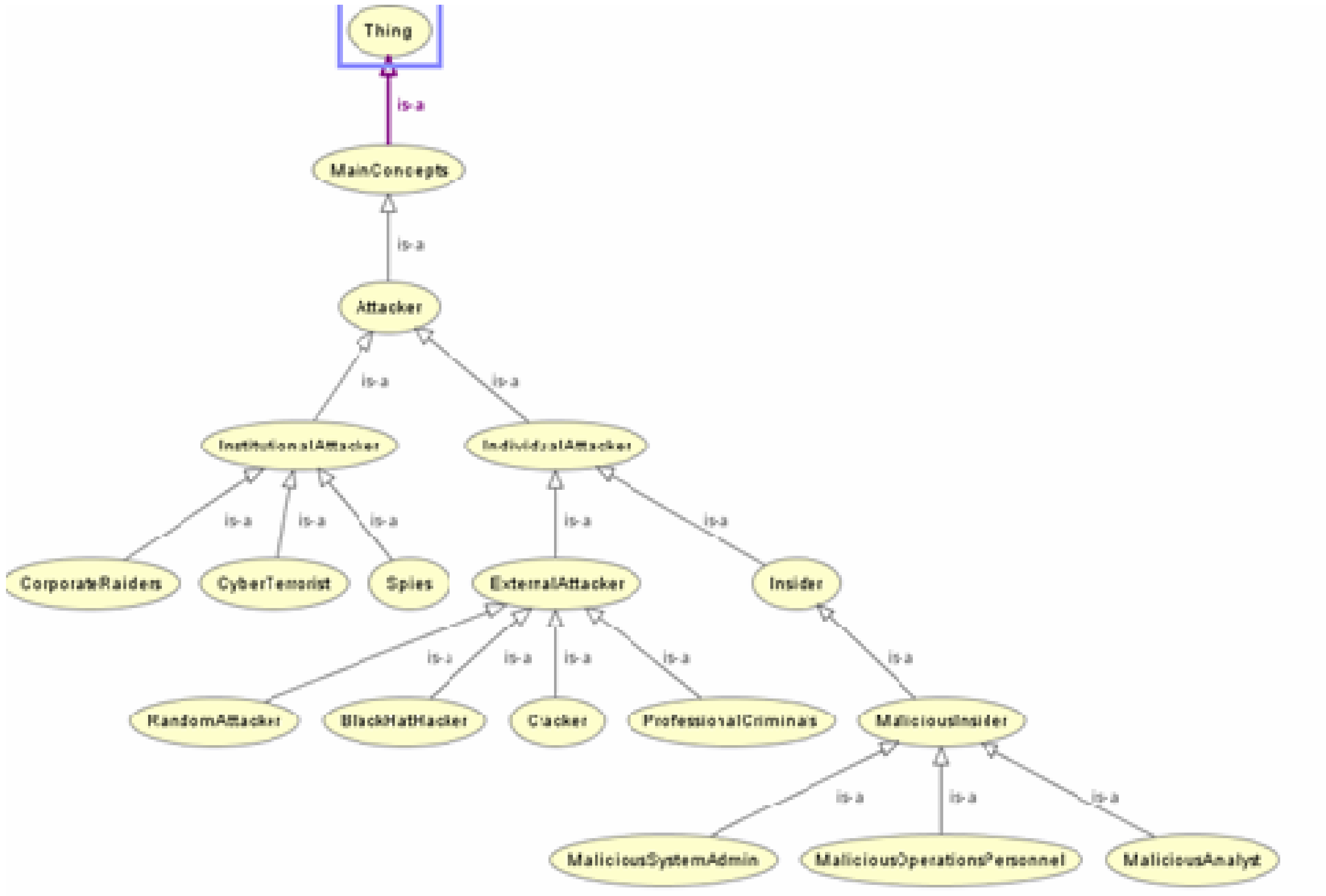
Entity Model

Entity Specification

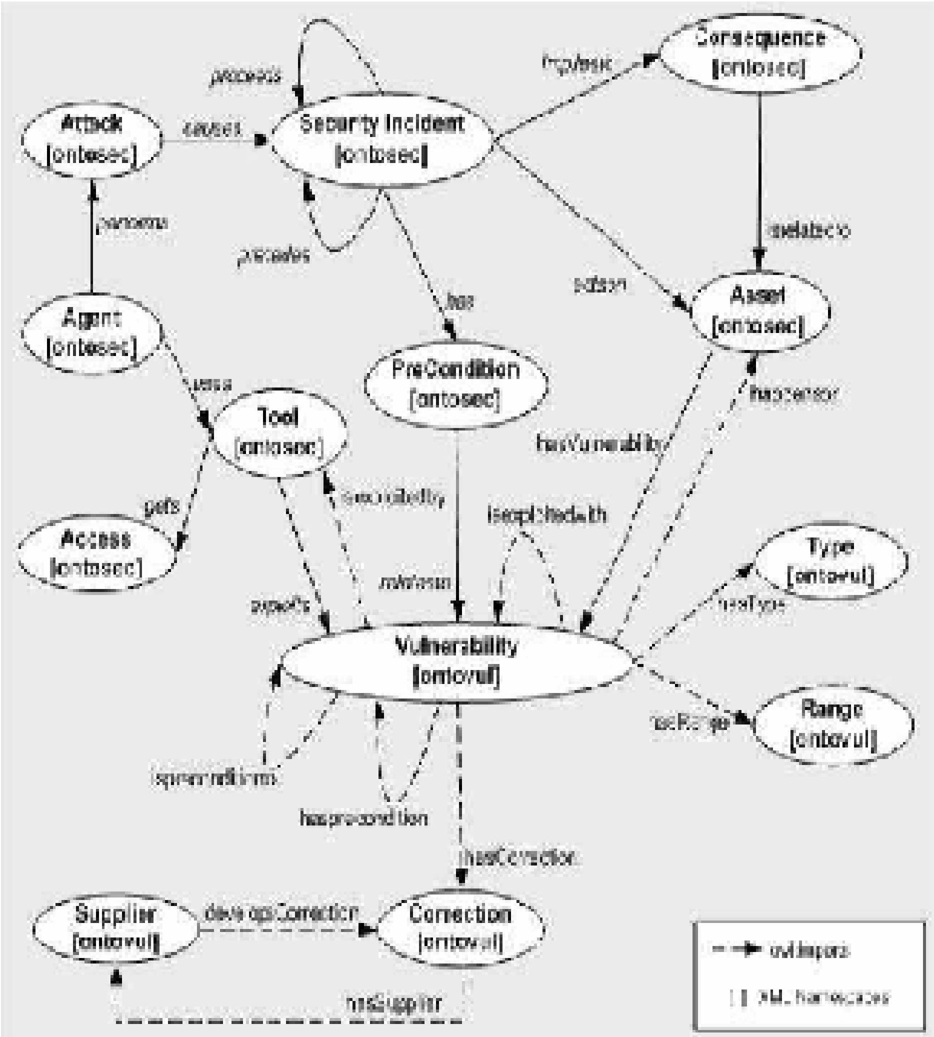
US Army M1 Abrams Main Battle Tank

- Combat Weight: 54.5 tons
- Maximum Speed: 45 mph
- Power to weight ratio: 27 HP/ton
- Length of Hull: 24.49 feet
- Height: 8.68 feet
- Total Crew: 4 soldiers
- Weapons: 120mm Howitzer, .50 Caliber Heavy Machine Gun, and two 7.62mm M60 GPMGs

Attacker Ontology



Attack Frame



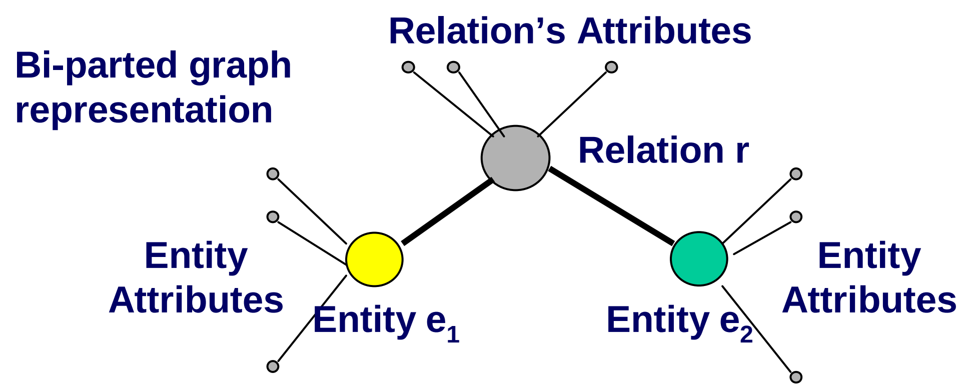
Relations

Entities are engaged in different class, structural, spatial and other domain-specific relations. Relation is a mental abstraction of linking a certain number, very often two, entities together. Mathematically, relation $R \subseteq E_1 \times \ldots \times E_m = \{(e_1, \ldots, e_m) / e_1 \in E_1, \ldots, e_m \in E_m\}$, where $E_1, \ldots, E_m \subseteq U$.

Relation R could be considered as a set of instant relationships $\{r_1, \ldots, r_q\}$. In case of binary relations the commonly used notation is $r = e_i R e_j$, where $r = (e_i, e_j) \in R$.

In several practical applications it is required to consider relations as entities, in sense that they are characterized by set of attributes $\{b_1, b_2, \ldots, b_h\}$, and all the

features that were attached to the attributes of entities.

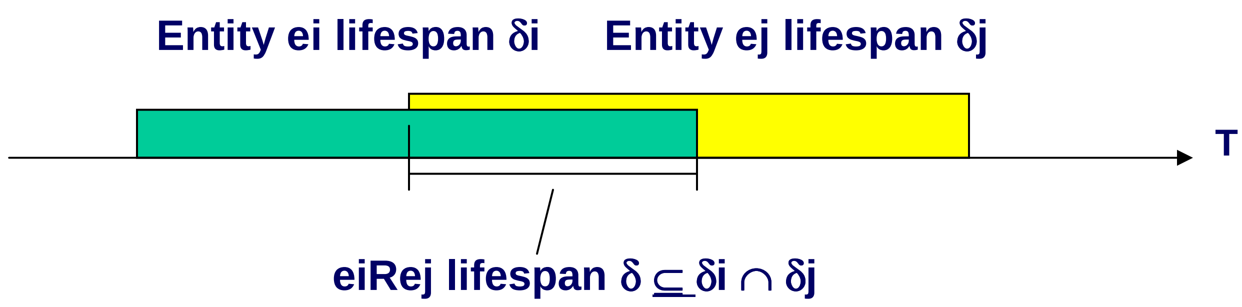


Time Dependent Features of Relations

We will consider relations as dynamic time dependent objects with their time of creation t', time of clear t", and lifespan δ=(t', t").

The following time dependency should hold for a relationship:

if $e_i R e_j$ and δ_i, δ_j are lifespans of e_i, e_j, accordingly, then for the relationship $e_i R e_j$ the lifespan $\delta \subseteq \delta_i \cap \delta_j$. Any attribute value b(t) of a relation is defined only during the existence of the relation, i.e. b(t), $t \in \delta$.



Types of Relations

- For our further discussion, it is important to consider the following types of relations between entities:
  - Class relations
  - Structural relations
  - Spatial relations
  - Domain-specific relations
- Class relation establishes a link between an entity and abstract entity class or between entity classes. Class relation is the major tool of conceptualization of entities and building conceptual frameworks of abstract concepts (ontologies).
- Structural relations Part-Of , Overlaprs-With and Similar-With are the basic construction primitives of the universe.
- Spatial relations Inside, Near, Above, etc. are used to express topological (spatial) links between the entities.
- There is large number of various domain specific relations, which semantics depends on he particualr domain. For example, Service x Supported-by Network y, Unit x Under-Fire-of unit y, Element x Connected- by Trunk-T1-to Element y.

Definition of Base Situations

1. Let $\{a_1, \ldots, a_p\}$ be set of situational attributes of entity e. Situation $S_e(d)$ on entity e during a time interval d, $d \subseteq \delta$, where δ is the lifespan of entity e is defined as

$$S_e(d) = \langle a_1(t), \ldots, a_p(t) \rangle \in v_1 \times \ldots \times v_p / \; \forall (t, t') \in d \; [\langle a_1(t), \ldots, a_p(t) \rangle = \langle a_1(t'), \ldots, a_p(t') \rangle]$$

2. Let $\{b_1, \ldots, b_q\}$ be set of situational attributes of relation R. Situation $S_R(d)$ on relation R during a time interval d, $d \subseteq \delta$, where δ is the lifespan of relation R is defined as

$$S_R(d) = \langle b_1(t), \ldots, b_q(t) \rangle \in v_1 \times \ldots \times v_q / \; \forall (t, t') \in d \; [\langle b_1(t), \ldots, b_q(t) \rangle = \langle b_1(t'), \ldots, b_q(t') \rangle]$$

3. Let $R \subseteq E_i \times E_j$, where $E_i, E_j \subseteq U$, $(e_i, e_j) \in R$, and δ_i, δ_j are lifespans of e_i, e_j, accordingly, then

$$S_{(ei,\, ej)}(d) = e_i R e_j$$

is a situation, where $d \subseteq \delta$, where δ is the lifespan of the relation R.

It is important to state that in the semantics of the relation R is defined as structural, spatial or domain specific relations.

Construction of Complex Situations

Complex situations could be constructed from other situations using set-theoretical union and inter-section operations.

1. If $S_{B1}(d_1)$ and $S_{B2}(d_2)$ are two situations, where $B_1, B_2 \subseteq U$ and d_1, d_2 are subsets of common lifespans of all entities in B_1, B_2, correspondingly, then,

$$S_B(d) = S_{B1}(d_1) \cup S_{B2}(d_2) \text{ and } S'_B(d') = S_{B1}(d_1) \cap S_{B2}(d_2)$$

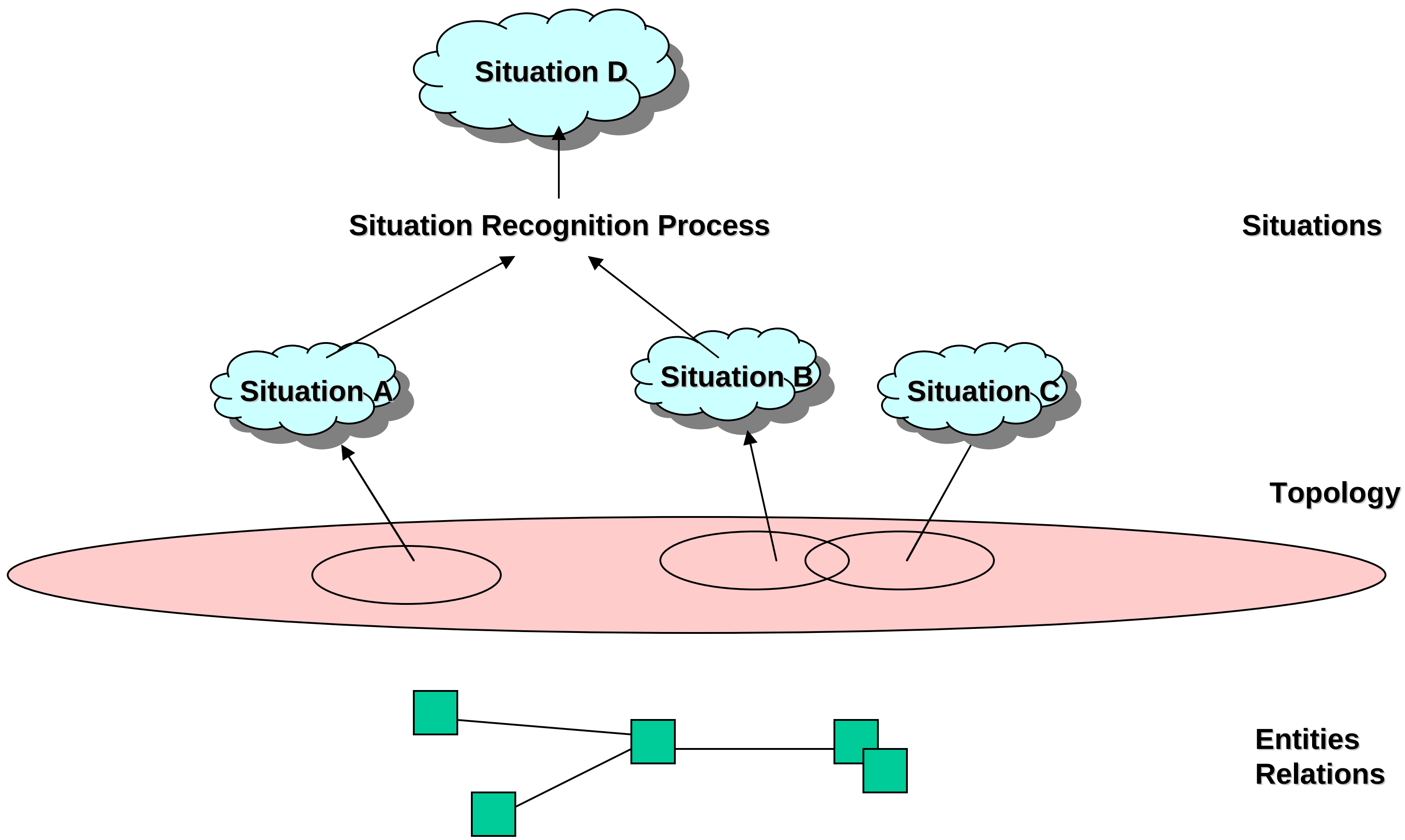
are situations, where, correspondingly

$$d = d_1 \cap d_2 \text{ and } B = B_1 \cup B_2 \quad \text{and } d' = d_1 \cap d_2 \text{ and } B = B_1 \cap B_2$$

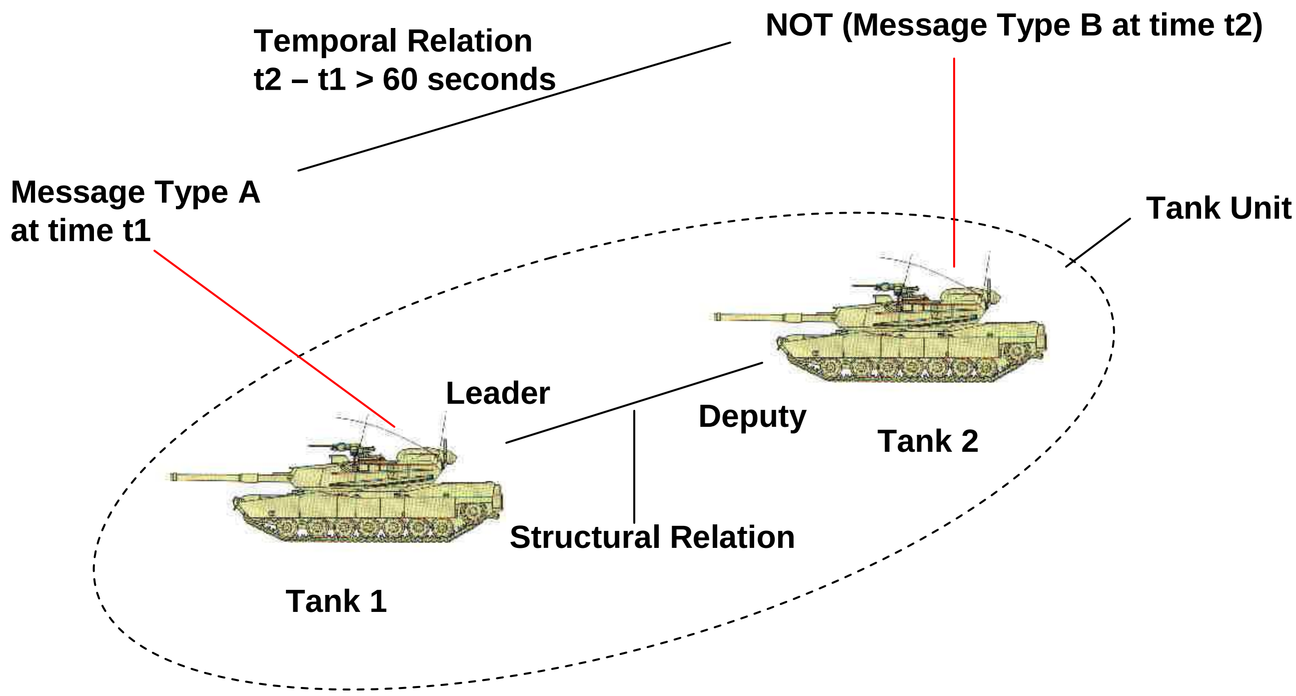
2. Due to use of active entities and situational attributes, multiple different situations can be defined on the same set of entities and relations

NB! Logical and temporal relations could be defined between situations, however only as between predicates, e.g. S' ^ S" and S' AFTER S" are predicates not situations

Recognition of Complex Situations



Situation Awareness Example



Situation Awareness (Recognition) Example

Suppose an event of type A was issued at time t1 from tank labeled *?tank1*, but during the following 60 second interval an expected event of type *B* was not issued from tank *?tank2*. It is known that tanks *?tank1* and *?tank2* form a unit, where *?tank1* is the leader and tank *?tank2* is the deputy supporting tank *?tank1*.

SituationRuleName: UNIT_CONTACT_LOST_SITUATION_RULE

Conditions:

```
MSG:  EVENT-TYPE-A  ?msg1
      TANK:  ?tank1
      TIMESTAMP  ?t1
Not   MSG: EVENT-TYPE-B ?msg2
      TANK:  ?tank2
      TIMESTAMP  ?t2
AFTER: ?msg2 ?msg1 60 TIMESENT ?t
RELATION  TANK_UNIT ?unit1
            LEADER  ?tank1
            DEPUTY  ?tank2
Actions:
Assert: UNIT_COTNACT_LOST_SITUATION ?situation1
      ATTRIBUTES
            ?msg1, ?msg2, ?unit1
```

Characteristics of the Event Correlation Process

- **Event Correlation is a time-dependent process in several aspects:**
  - **Correlated events have limited time of existence and during the correlation process they might expire**
  - **Temporal relations between events are significant**
  - **The contributing events should happen within a pre-defined time period**
  - **The latency in the event communication lines may distort the actual sequence of incoming events**
- **As a general rule – the correlation process follows the real time and should be fast, e.g. hundreds of events processed per second**

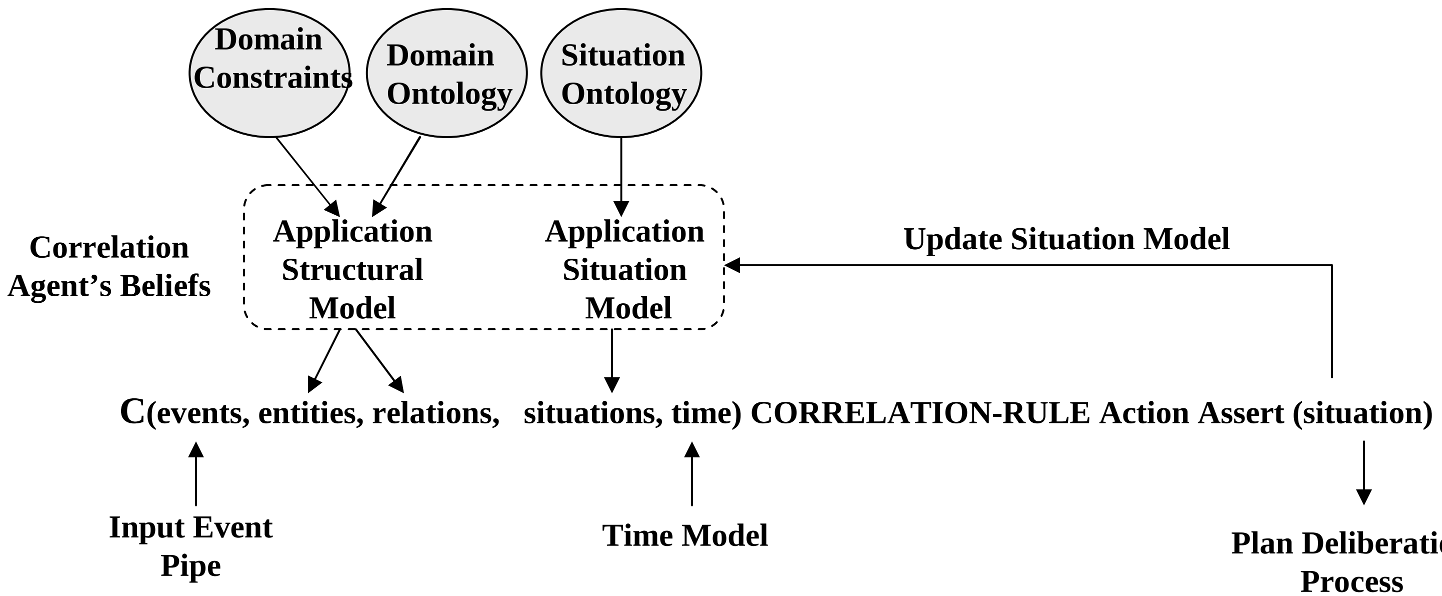
Event Correlation Classes

1. [a, a, ..., a] ⟶ [a]	Compression
2. [a, p(a) < H] ⟶ [nil]	Filtering
3. [a, C] ⟶ [nil]	Suppression
4. [n x a] ⟶ b	Count
5. [n x a; p(a)] ⟶ a, p'(a), p' > p	Escalation
6. [a; a *subclass* b] ⟶ b	Generalization
7. [a; b *subclass* a] ⟶ b	Specialization
8. [a T b] ⟶ c	Temporality
9. [a, b, ...T, *AND, OR, NOT*] ⟶ c	Logic/Temporality

Event Correlation Classes (Cont.)

1. **Compression – reduction of multiple occurrences of identical events a into one event**
2. **Filtering – event a is eliminated, if parameter p(a) of event a does not belong to the set of pre-defined values H**
3. **Suppression – event a is temporarily inhibited during the presence of context C**
4. **Counting – n occurrences of event a generates event b**
5. **Escalation – assigns a higher value to the event a parameter p(a), e.g. priority, depending on some context, e.g. number of event a occurrences**
6. **Generalization – replaces event a with its superclass event b**
7. **Specialization – replaces event a with its subclass event b**
8. **Temporality – generates a new event c depending of the existence of temporal relations, e.g. BEFORE, AFTER between the events a and b**
9. **Clustering – generation of complex event patterns c using Boolean operators over conditional (predicate) terms**

Event Correlation and Situation Recognition Process



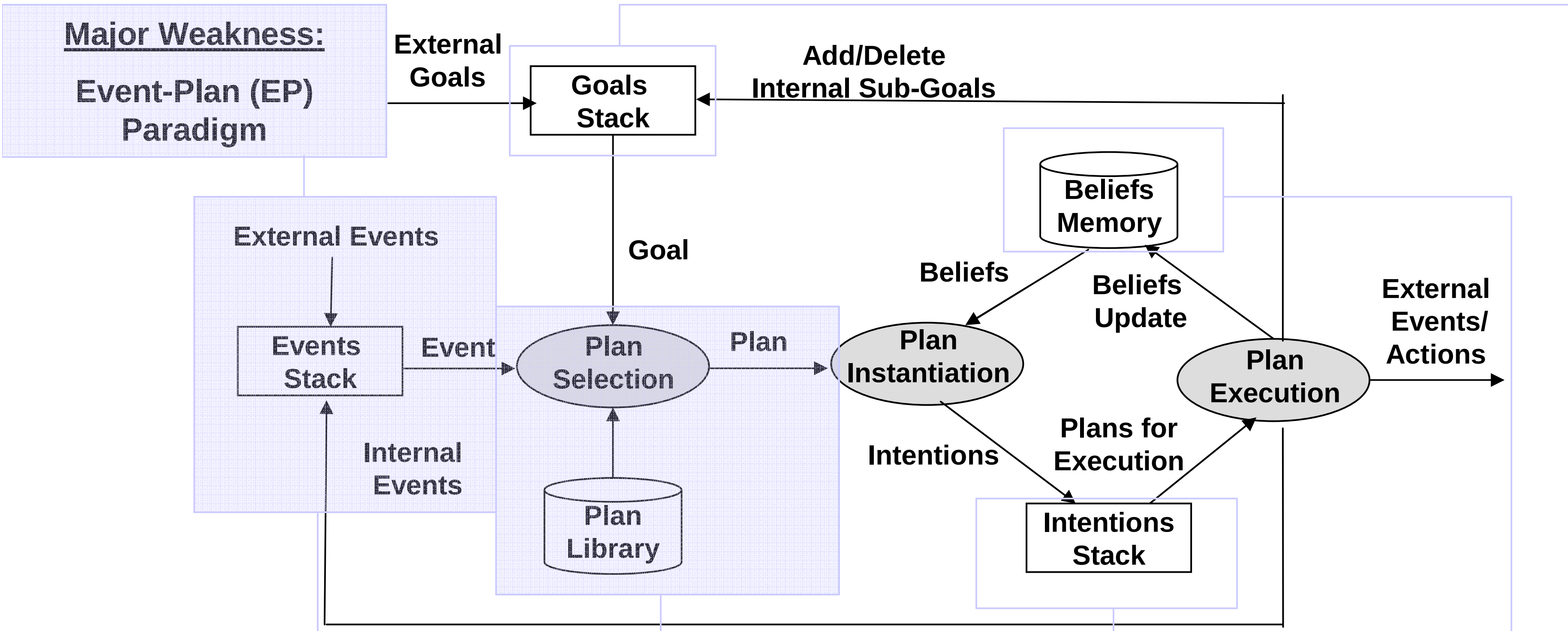
Multi-Agent Systems

- **The paradigm of multi-agent systems (MAS) has its roots in distributed artificial intelligence, object oriented systems and human team cognition.**
- **MAS is currently one of the most powerful approaches used in building distributed computing systems.**
- **MAS has several important features which correspond to our specific interests, particularly:**
  - **Adaptivity: the ability to reorganize and improve behavior with experience**
  - **Autonomy: goal-directedness, proactive and self-starting behavior**
  - **Collaboration: the ability to work with other agents to achieve a common goal**
  - **Inference: the ability to act on abstract task specifications**
  - **Mobility: migration in physical or cyber space**

Current MAS Approach to Situation Awareness

- A typical MAS solution to situation awareness, and consequently to the whole process of command and control, is based on dividing situation awareness (command and control) into several dedicated agents either across functional tasks, e.g. data detection, classification, visualization, etc., or across levels of abstraction of information, e.g. signal, data and semantic information levels.
- Most of the MAS complexity is in the internal agent architecture, the data/knowledge representation and the inference procedures, while inter-agent communication is simplified.
- More sophisticated MAS architectures establish inter-agent communication rules guiding the flow of data and control.
- A few MAS have introduced an ontology-based architecture which allows a semantically deeper data structure, and most importantly, the unifying of conceptually different data representations from different agents.

Classical Belief Desire Intension (BDI) Agent Architecture

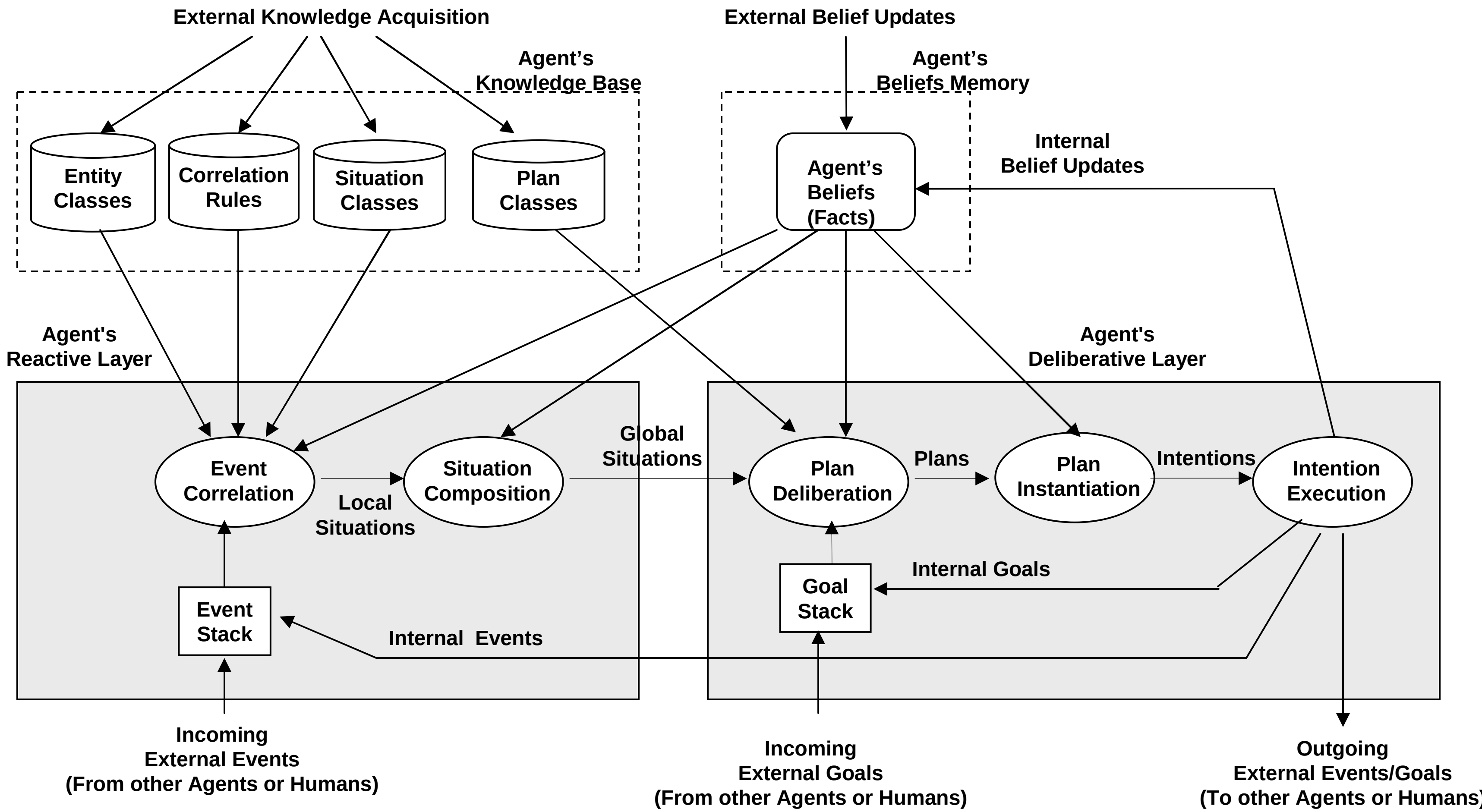


Based on the dMARS formalism – a well recognized reference model for BDI agents

5 main notions:
events, plans, intentions, beliefs & desires

d'Inverno, M., Luck, M., Georgeff, M., Kinny, D., and Wooldridge, M. The dMARS Architechure: A Specification of the Distributed Multi-Agent Reasoning System, In *Journal of Autonomous Agents and Multi-Agent Systems*, 9(1-2):5-53, 2004

Situation-Aware BDI Agent Architecture



Control Alternatives for MAS Architectures

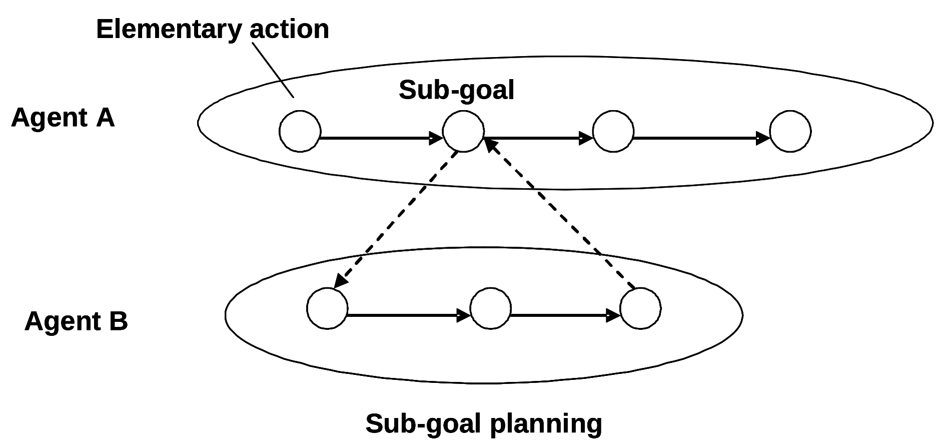
- **We are considering following MAS architectures**
  - **Hierarchical**
  - **Federated**
  - **Mission-oriented**
- **<u>A hierarchical architecture</u> is based on a pre-defined configuration and command/control structure between agents, which are positioned on multiple levels. This corresponds to a well-defined (rigid) command structure and is better suited for deterministic, predictable combat situations.**
- **<u>A federated architecture</u> defines regions of local autonomy from a higher-level central command. The central command defines tasks (goals, plans, beliefs, knowledge, etc) and reporting structure, while execution of local task is left to a region. A federated architecture is the most promising for tactical UCO.**
- **<u>A mission-oriented architecture</u>, which is the most flexible, is dynamically planned depending on the battleground situation, the mission to be executed and the available resources. This architecture fits best the unpredictable nature of battleground situations.**

Inter- Agent Collaboration

- Collaboration of agents in MAS requires that the agents possess a capability to understand each other, share a common understanding of the surrounding environment and communicate effectively. For UCO the collaborating agents should have at least the following resources:
  - An effectively organized and computable set of common goals
  - A shared ontology
  - A set of inter-agent communication channels
  - A common or transformable format for content exchange
  - A set of collaboration acts, and procedures (policies) for implementing the collaboration acts
  - Situation awareness of the urban combat environment
  - A system of constraints (resource, time, quality, etc.) affecting the collaboration acts.

Scenario-Driven Collaboration

- **A collaboration scenario is a sequence of pre-defined collaboration acts performed by two agents.**
- **The following illustrates such collaboration between two agents, Agent A and Agent B. Agent A is performing certain actions belonging to a mission. Agent A encounters a task which has as a sub-goal - a task for which Agent A is not an expert, e.g. the task of planning a physical route for tanks. Agent A, using some method to find a task expert, determines that Agent B is a task expert and initiates a collaboration process with Agent B.**



Sample Scenario” Physical Route Planning

Collaboration_Scenario “Physical Route Planning”
Scenario_Type Sub_Goal_Planning
Initating_Agent Agent A
Partner_Agent Agent B
Collaboration_Actions

- **Action_1 <Agent A initiates collaboration request>**
- **Action_2 <Agent A selects collaboration partner agent>**
- **Action_3 <Agent A passes task information>**
- **Action_4 <Agent B acknowledges collaboration act>**
- **Action_5 <Agent B resolves the sub-goal>**
- **Action_6 <Agent B passes the result to Agent A>**
- **Action_7 <Agent A and Agent B acknowledge completion of the collaboration>**

Policy-Based Collaboration

- Policies are rules for normative behavior. They describe very specific acts and constraints for collaboration between two agents.
- Unlike collaboration scenarios, policies do not describe the complete deterministic sequence of the collaboration acts.
- In addition, policy-based collaboration might involve the following elements:
  - Simple elements of negotiation, e.g. a request for more information by a collaborating agent before resolution of a sub-goal
  - Dynamic adjustment of the collaboration actions depending on the events of an unfolding situation
  - Limited mixed-initiative collaboration, where certain collaboration acts are initiated by the partner agent
- Scenario-based and policy-based collaboration are currently considered the two main collaboration strategies between agents for C2 in urban combat operations.

Thank You for Your Interest!

Situation Calculus

- The first formal specification of a situation was given by McCarthy and Hayes in their Situation Calculus, where they used first order logic (FOL) expressions to define a situation as a snapshot of a <u>complete world state</u> at a particular time.
- Since it was computationally inefficient to consider a situation as a complete state of the world, Reiter and Pirri in their approach to situation calculus defined a situation as a <u>sequence of actions</u> enabling calculation of the current state knowing the initial state and the sequence of actions transforming the initial state.
- One of the extensions of Situation Calculus involves using fluents, situation-dependent functions in describing situations

J. McCarthy and P. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence." In D. Michie, editor, *Machine Intelligence 4*, American Elsevier, New York, NY, 1969.

F. Pirri and R. Reiter, Some Contributions to the Situation Calculus. *J. ACM*, 46(3): 325-364, 1999.

References

Situation Management

Jakobson G. et al D-AESOP: A Situation-Aware BDI Agent System for Disaster Situation Management, AAMAS 2006, 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems, Hakodate, Japan, May 2006

Buford, J., Jakobson,G., and Lewis, L. Case Study of Urbain Transportation Threat Monitoring Using the AESOP Situation Manager™. The 2005 IEEE Technologies for Homeland Security Conference, Boston, MA., 2005.

Jakobson, G. Buford, J., and Lewis, L. Towards an Architecture for Reasoning About Complex Event-Based Dynamic Situations, International Workshop on Distributed Event based Systems DEBS'04, Edinburgh, UK, 2004.

Salerno, John J., Hinman, Michael, and Boulware, Douglas, "Evaluating Algorithmic Techniques in Supporting Situation Awareness", In *Proc of the Defense and Security Conference*, Orlando, FL, March 2005.

Chris J. Matheus, Mitch M. Kokar, and Kenneth Baclawski. A core ontology for situation awareness. In Proc. Sixth Int. Conf. Information Fusion, pages 545 -552, Cairns, Australia, 8-11 July 2003.

References

Situation Modeling and Management

1. John McCarthy and Patrick Hayes. Some philosophical problems from the standpoint of artificial intelligence. In Donald Michie, editor, Machine Intelligence 4, American Elsevier, New York, NY, 1969.

2. Richard D. Gilson. Introduction to the special issue on situation awareness. Human Factors, 37(1): 3-4, 1995.

3. Fiora Pirri and Raymond Reiter. Some contributions to the situation calculus. J. ACM, 46(3): 325-364, 1999.

4. Hector J. Levesque, Raymond Reiter, et al. GOLOG: a logic programming language for dynamic domains, J. Programming, 31: 59-84, 1997.

5. Reiter,Y. Lesperance, and H. J. Levesque – “A Situation Calculus Approach to Modeling and Programming Agents”. A. Rao and M. Wooldridge, (eds.), Foundations and Theories of Rational Agency. Kluwer, New York, 1997.

6. Zwaan, A.R., and Radvansky, G.A. Situation Models in Language Comprehension and Memory, Psychological Bulletin, 123, pp. 162-185, 1998

7. Jon Barwise, “Scenes and other Situations”, Journal of Philosophy, vol. 77, 1981, pp. 369-397.

8. Mica Endsley, Designing for Situation Awareness, Taylor & Francis LTD, 2003

References

Knowledegre representation and Ontology

1. R. Davis, H. Shrobe, and P. Szolovits. What is a Knowledge Representation? *AI Magazine*, 14(1):17-33, 1993.
2. http://en.wikipedia.org/wiki/Knowledge_representation#References
3. Chris J. Matheus, Mitch M. Kokar, and Kenneth Baclawski. A core ontology for situation awareness. In Proc. Sixth Int. Conf. Information Fusion, pages 545 -552, Cairns, Australia, 8-11 July 2003.
4. Barry Smith, The Basic Tools of Formal Ontology, in Nicola Guarino (ed.), *Formal*
5. *Ontology in Information Systems* Amsterdam, Oxford, Tokyo, Washington, DC: IOS Press Frontiers in Artificial Intelligence and Applications, 1998, 19–28.
6. http://www.aaai.org/AITopics/html/ontol.html
7. *Blackwell Guide to the Philosophy of Computing and Information*, Oxford: Blackwell, 2003, 155–166.
8. K. Baclawski, M. M. Kokar, P. A. Kogut, L. Hart, J. Smith, W. S. Holmes III, J. Letkowski and M. L. Aronson. *Extending UML to Support Ontology Engineering for the Semantic Web*. Proceedings of the Fourth International Conference on the Unified Modeling Language, Toronto, Canada, 2001.

References

Event Correlation

1. Gabriel Jakobson, Mark Weissman, Leonhar. Brenner, Carol Lafond, and Chris Matheus. GRACE: building next generation event correlation services. In IEEE Symp. Network Operations and Management, pages 701-714, Honolulu, HI, 10-14 April 2000.
2. Yemini, S., Kliger, S., Yemini, Y., Ohsie, D., High Speed and Robust Event Correlation. IEEE Communications Magazine, May 1996.
3. Jakobson, G., and Weissman, M., Real-Time Telecommunication Network Management: Extending Event Correlation with Temporal Constraints, Fourth IFIP/IEEE International Symposium on Integrated Network Management, May 1995, Santa Barbara, CA.
4. Jackson, P., Introduction to Expert Systems, Addison Wesley, 1999

References

General Topics

1. Ikuo Tahara, editor. Cognitive Information Processing. IOS Press, Amsterdam, The Netherlands, 1994.
2. Ronald Brachman – DARPA Cognitive Information Processing Technology Program http://www.darpa.mil/ipto/Solicitations/open/02-21_PIP.htm)
3. Jackson, P., Introduction to Expert Systems, Addison Wesley, 1999.
4. Jini Network Technology. http://wwws.sun.com/software/jini/
5. Jon Siegel. CORBA Fundamentals and Programming. John Wiley & Sons, 1996.

Troy B. Downing. Java RMI: Remote Method Invocation, IDG Books Worldwide, 1998.

6. Grady Booch: *Object-Oriented Analysis and Design with Applications*, Addison-Wesley.
7. G. Booth, J. Rumbaugh, I. Jacobson. The Unified Modeling Language Guide. Addison-Wesley.
8. Kohavi, Z., *Switching and Finite Automata Theory*. McGraw-Hill, 1978.
9. Paul E. Black, “Hidden Markov Model", from Dictionary of Algorithms and Data Structures, NIST. http://www.nist.gov/dads/HTML/hiddenMarkovModel.html
10. Charles Forgy, "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem", Artificial Intelligence,19, pp 17-37, 1982.

References

Information Fusion

1. Franklin E. White, Jr., Data Fusion Lexicon, Joint Directors of laboratories, Technical Panel for C3, Data Fusion Sub-Panel, Naval Ocean Systems Center, San Diego, 1987
2. James Llinas et al "Revisiting the JDL Data Fusion Model II", Proceedings of the 7th International Conference on Information Fusion, Stockholm Sweden, June 2004, http://www.fusion2004.org/
3. Franklin White. Data fusion sub-panel report, In Proc. 1991 Joint Service Data Fusion Symposium, volume 1, pages 335-361, 10 October, 1991.
4. Alan N. Steinberg, Christopher L. Bowman, and Franklin E. White. Revisions to the JDL data fusion model, In Proc. NATO IRIS Conf., Quebec, Canada, October 1998.

5. Data Fusion Server http://www.data-fusion.org/index.php
6. International Society of Information Fusion http://www.inforfusion.org/
7. The Journal of Advances in Information Fusion http://www.inforfusion.org/JAIF-CFP-Oct28.htm
8. Information Fusion, An International Journal on Multi-Sensor, Multi-Source Information Fusion http://www.elsevier.com/wps/find/journaldescription.cws_home/620862/description#description