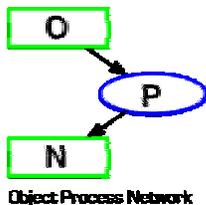


# Towards Automatic Systems Architecting

From abstract concept to  
architecture

Gustavo Pinheiro, Felipe Simon and Geilson Loureiro  
July, 2007



# Agenda

---

- Intro to the Project
- The Current Approach vs. The New Approach
- Applications
- The Hierarchical OPN
- Further Development / Conclusion

# Introduction

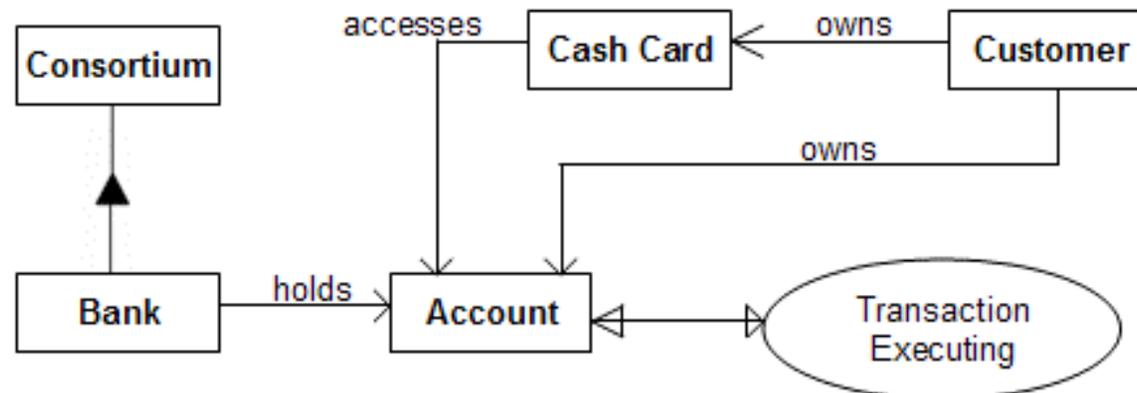
---

- Systems Architecture Domain
  - To Consider **many options** when designing a system
  - To be able to model a **specific solution/architecture** in an efficient manner
  - Available Languages / Tools :
    - **OPM**, Structured Analysis, UML, SysML ...
    - **OPN** (Decision-Support tool)
- Goal:
  - To define a **new approach** that allows **both perspectives** to be considered

# The Current Approach

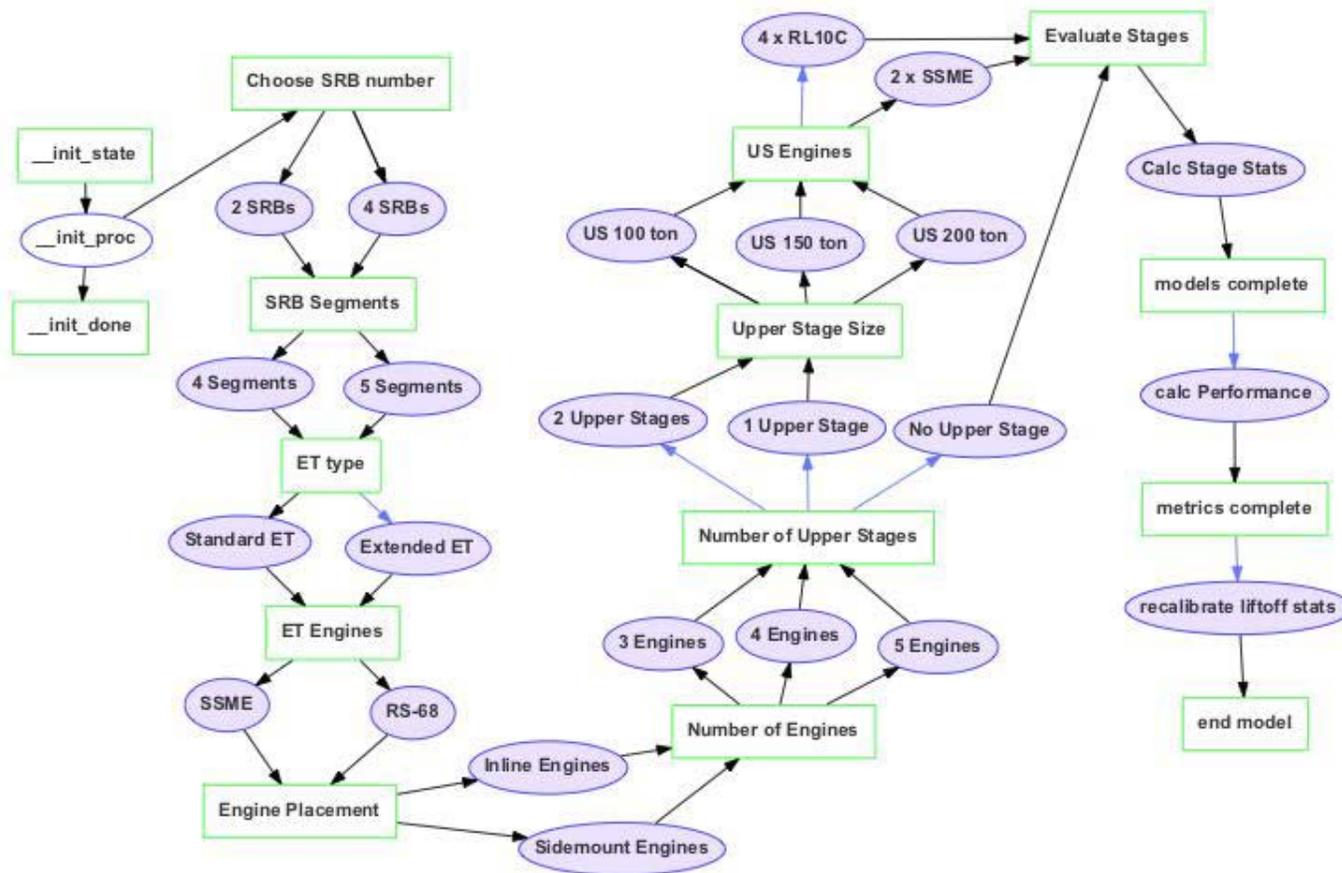
---

- Currently, decision-support tools are **completely separated** from system architecture modeling tools:
  - When **deciding**: you do not have instruments for a common visual understanding of the system
  - When **modeling**: you do not have instruments for deciding which way to move forward
- OPM Model:



# The Current Approach

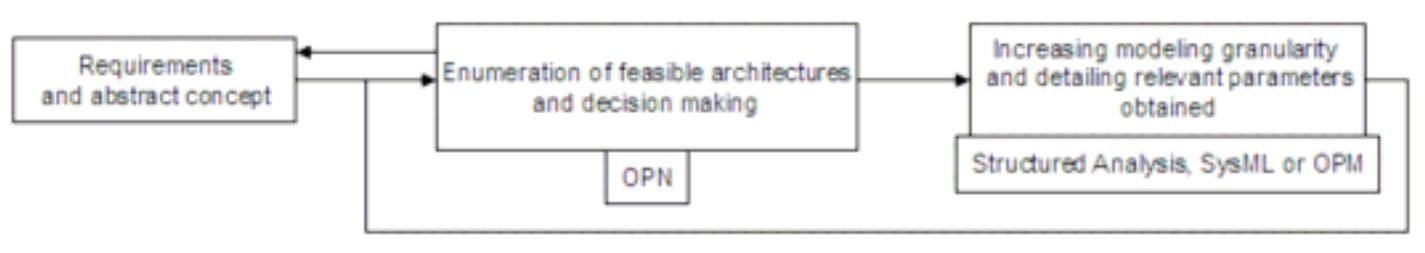
- OPN Graph (Higher-Level of Abstraction) :



# The New Approach

---

- Association OPM-OPN
  - Model the **Space of Options** with OPM
  - **Systematic Translation** to OPN
  - **Results** presented using **OPM notation** (for each architecture)

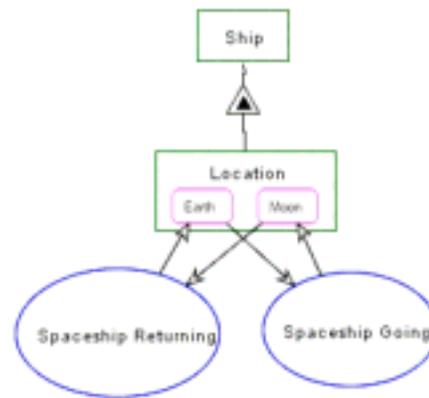


- That's what we call "Automatic Systems Architecting"

# The New Approach – Step-by-Step

---

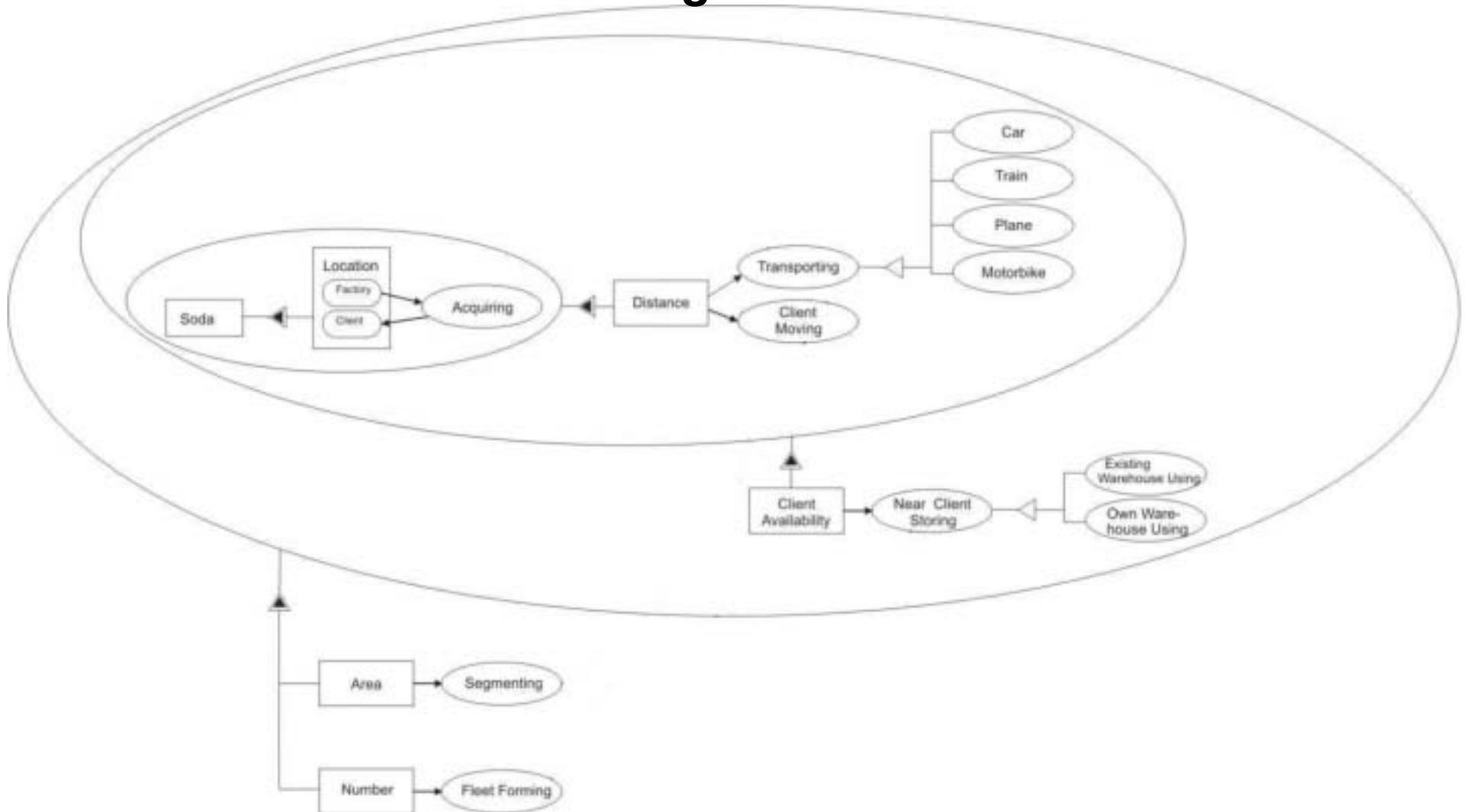
1. Define the **function** to be performed by your system.



2. Define **Boundary Conditions** (BC) to your Problem (they hinder the change from an initial state to a final state).
3. Assign **Functions** that “solve” these boundary conditions and the parameters which are important for making a decision to “feed” the OPN model.
4. Show **Specialization** possibilities for this solution and how they alter the parameter’s values.
5. If necessary, define **New BC** to these functions (iterate between 2, 3 and 4).
6. Check **Architectures** selected by OPN

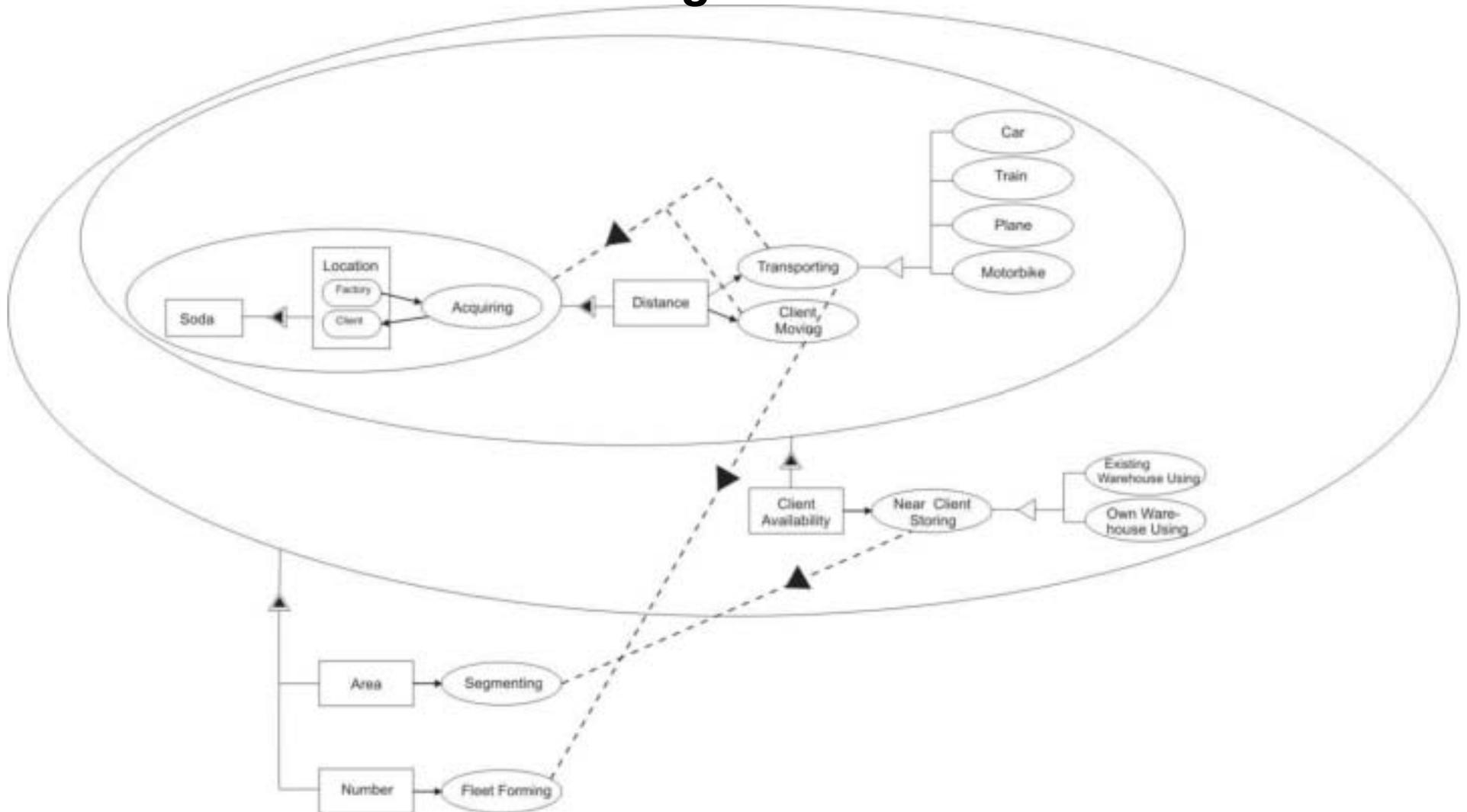
# Application 1

– Market of Sodas – *Logistic issues*



# Application 1

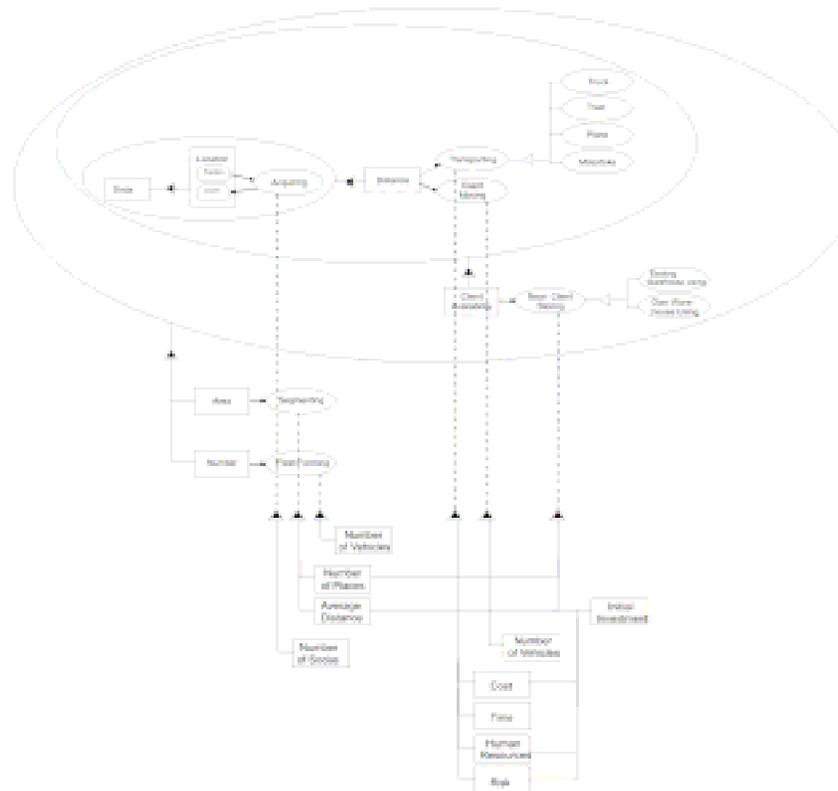
– Market of Sodas – *Logistic issues*



# Application 1

---

- **Adding parameters** that will feed the OPN model



The functions are “described” in terms of these parameters

# Application 1

- **Translating** the decisions to be made to OPN

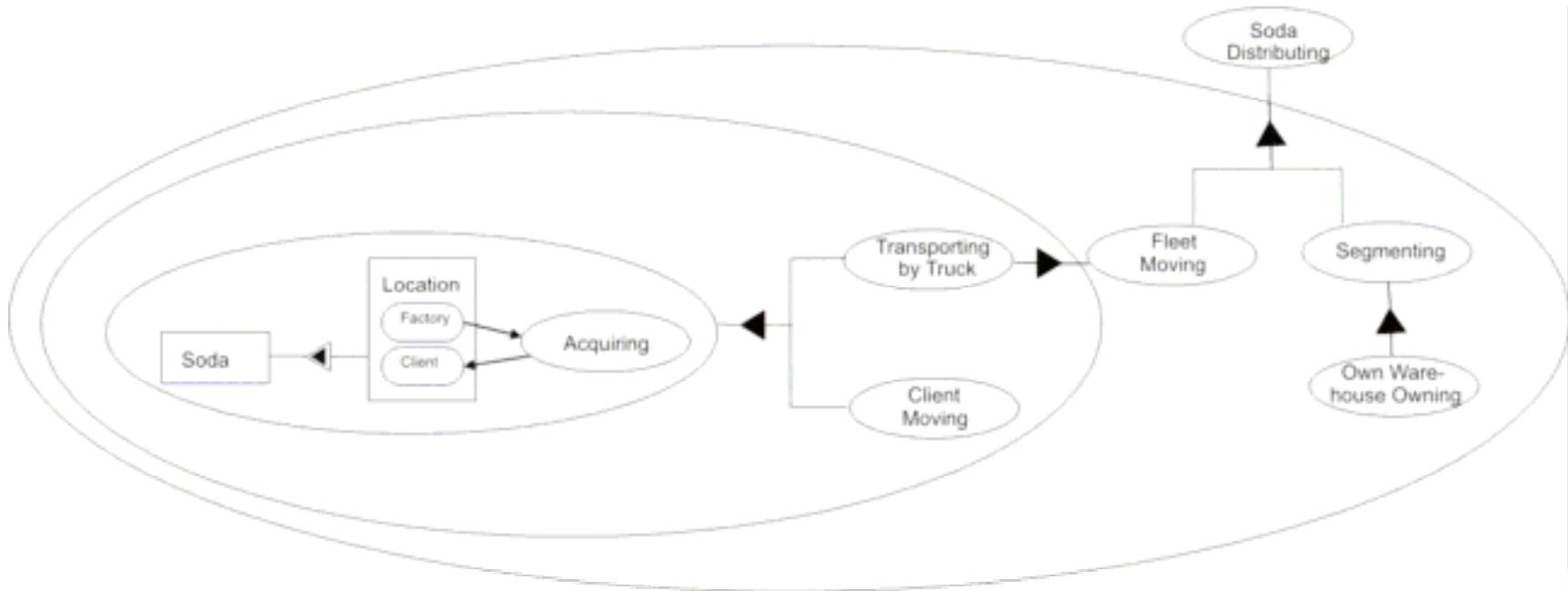


- Modeling the **decisions made** using **OPM** notation



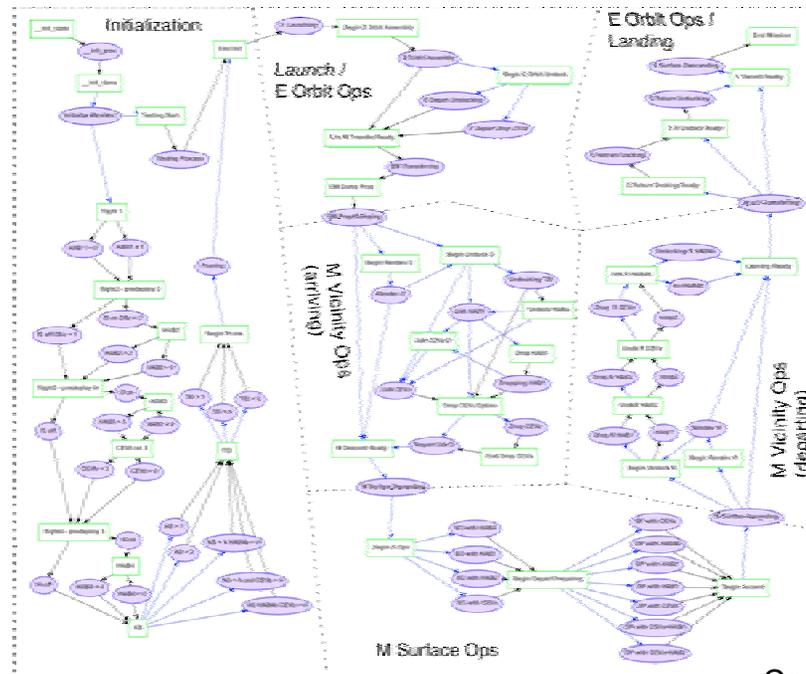
# Application 1

- Eventually, the decisions made can be modeled in the form of the **actual system** with OPM  
(This OPM model represents an architecture pointed out by OPN)



# Applications - Questions

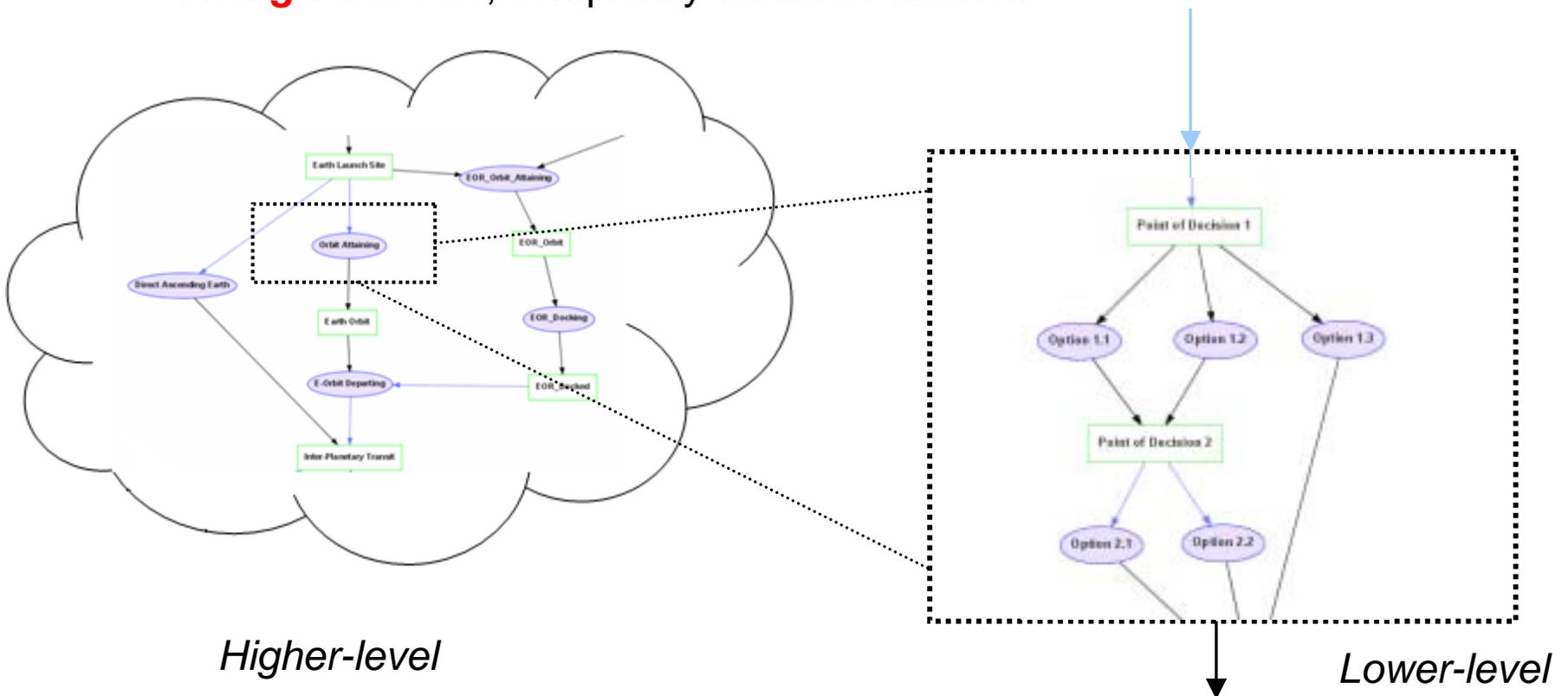
- For a simple application, a **single** OPN model may be sufficient to model all the “points of decision” to be considered.
- The **more complex** our system becomes, the more difficult it is to integrate knowledge from different domains (i.e. decisions in different areas) in a single OPN model.
- To try to model such a system using OPN current features leads to a model with **low cohesion** and **high coupling**.



Courtesy of Simmons, Koo, Crawley

# The Hierarchical OPN

- But what we need is **high cohesion** and **low coupling!**
- **Solution:** The Hierarchical OPN
- The recursivity presented in the new approach would be defined in **lower levels** OPN models.
- At **higher levels**, complexity would be hidden.



# The Hierarchical OPN

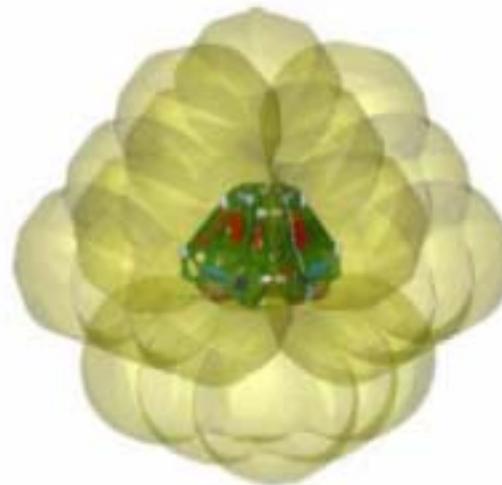
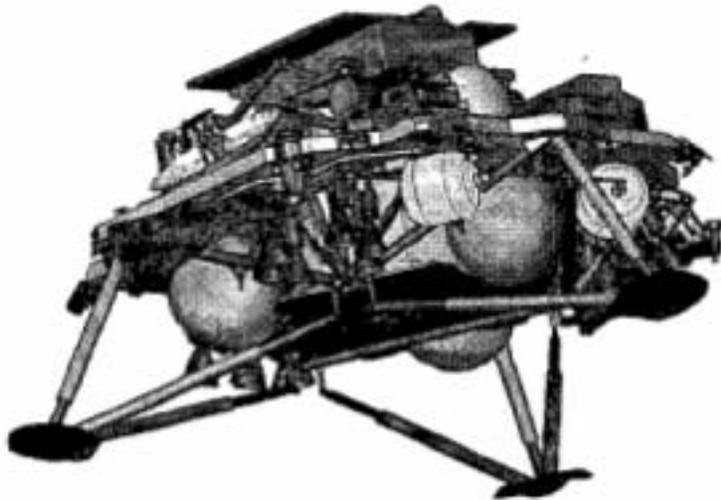
---

- Pros:
  - To allow the *design of complex systems* (that involves experts from multiple domains) using OPN tool.
  - To provide *higher cohesion* and *lower coupling*
- The New Approach:
  - Tool that integrates OPM with Hierarchical OPN
  - Note: When *different notations* are been used, one should to *translate* all of them to *OPM*. We've verified it's quite easy translation between SA, OPM, SysML, UML

# Application 2

---

- ***Lunar Lander*** (Under Development)
  - This example will show:
    - Exactly how the ***different levels*** will ***communicate*** with each other during simulation process; (parameters from higher levels modifying lower levels parameters and vice-versa).
    - That this “better organized” approach (high cohesion and low coupling) will lead to ***model reuse***



***Legged and Air-bag concepts – Two of the options considered for the Lunar Lander***

# Further Development / Conclusions

---

- How to figure out to which extent we should model? We could spend effort ***modeling a solution that will never be developed!***
- How can the models evolve?
- Conclusion of current study case (Lunar Lander)
- Implementation of an *user-friendly* tool able to mechanize the proposed approach (Association OPM + Hierarchical OPN)