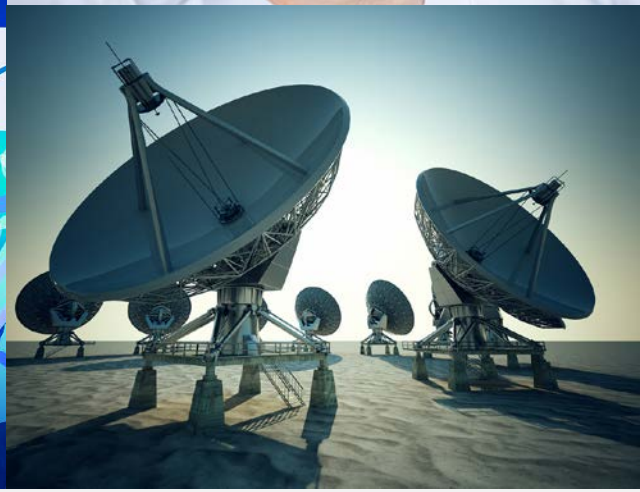
FRAUNHOFER USA

Center for Experimental Software Engineering (CESE)



Fraunhofer USA

Center for Experimental Software Engineering

Modeling Requirements of Autonomous System

Gudjon Magnusson, Madeline Diep, Tim Phillips, and Mikael Lindvall
Fraunhofer Center of Experimental Software Engineering

Presented at:
Flight Software Workshop December 2018

Fraunhofer USA
Center for Experimental Software Engineering

Motivation

- Strong interest in adopting autonomous capabilities
- Autonomous systems challenging to assure because behavior not always fully specified
  - There exists uncertainty in the environment where the system is deployed
- Better specification can lead to better assurance

→ How to do better specification for autonomous system?

Outline

- Definition of autonomy for this work
- Requirements for autonomous system
- Our modeling approach
- Identifying issues with requirements

What is autonomy

- Many levels of autonomy based on the degree of “adaptability”, e.g., automation vs autonomy

- **Autonomous feature**: a function that achieves its objective without human intervention
- **Autonomous system**: system with at least one autonomous feature

Autonomous System Requirements

High-level objective of a behavior
E.g., Navigate to destination without running into obstacle

Determination of actual behavior involves reasoning, decision making and complex evaluation of the environment

Detailed and specific behavior
E.g., Table of rules specifying a large set of possible environment conditions and desired responses

Autonomous System Requirements

- Our adopted definition:

Requirements for autonomous systems describe the system's desired behavior under **a dynamic environment based on available information** where **there exists uncertainty** that cannot be engineered away.

- Since autonomous behaviors **cannot be fully predetermined**, it can be difficult to reason about their completeness and correctness.

Why Modeling (Graphically)

- Modeling is known to be a good method for managing complexity and communicating complicated ideas.
  - Model abstracts away unnecessary details
  - Assists in understanding of dependencies and relationships through visual representation or diagrams
- The act of transforming natural language requirements to model has been shown to be capable of identifying requirements problems

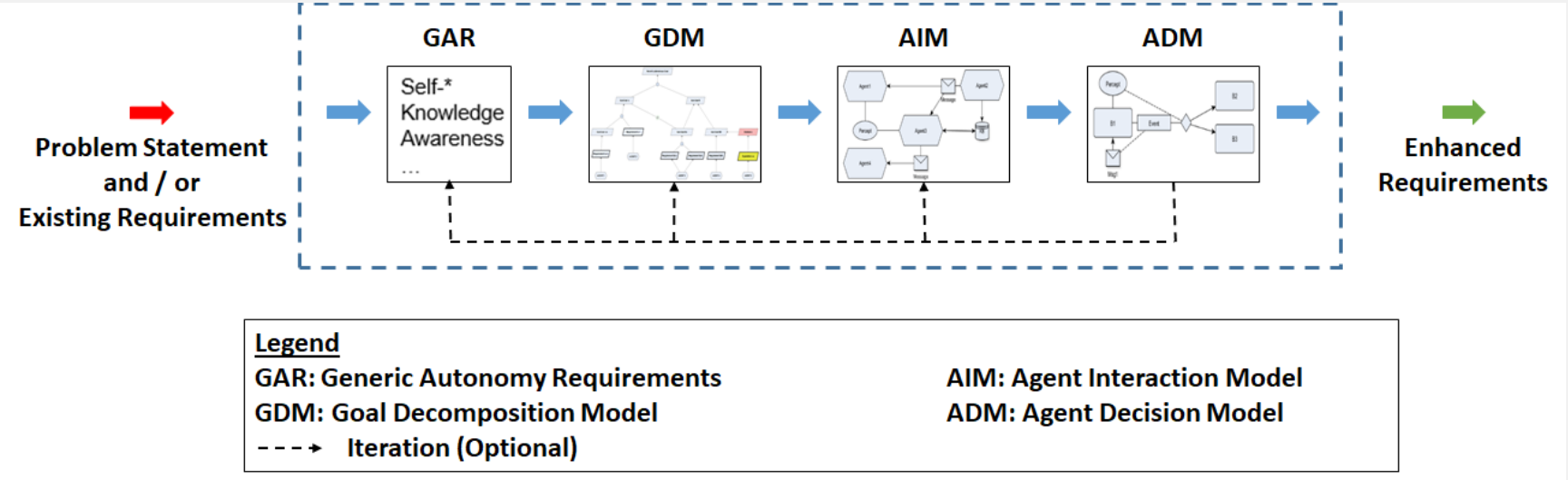
Why Modeling (Graphically)

- Modeling autonomous system requirements may provide engineers insights into nature of uncertainty and the expected behavior of an autonomous system
  - Early identification of requirements problem and reduce the risk of errors as the project moves from design to implementation
  - Good requirements provide basis for good testing

Our work

- We are interested in understanding:
  - What abstractions are useful to express behavior of autonomous systems?
  - What analysis can be performed on the models to improve requirements?

Our Modeling Approach

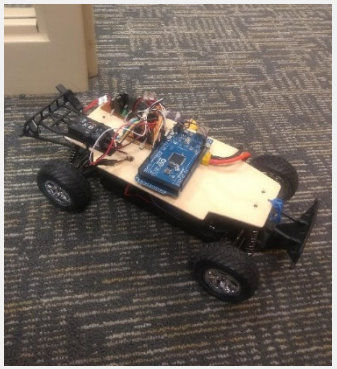


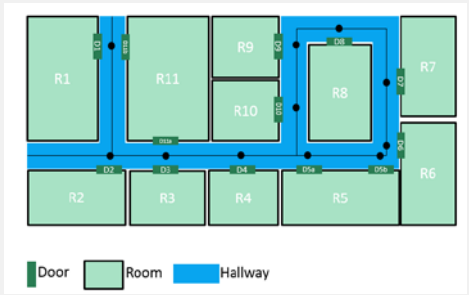
3-tiers approach (supported by GAR) → Start from high-level to more specific requirements

Our System of Analysis– Autonomous Office Rover

- The following autonomous system will be used illustrate our modeling approach:

- High-level Requirements
  - Accept payloads from persons in the office
  - Deliver payloads to a specific room in the office
  - Minimize time spent traveling
  - Wait at home base when idle
  - Avoid running out of power
  - Avoid running into obstacles
- Idea was to incorporate features that compare to NASA mission autonomy





Fraunhofer USA

Center for Experimental Software Engineering

Generic Autonomous Requirements (GAR)

- Proposed by Vasey et. al.
- Provides categorization to elicit self-* requirements and their supporting requirements
  - Self-* requirements for autonomous features, e.g., self-navigate, self-plan
  - Supporting requirements for each self-* requirement including:
    - Awareness – ability to notice change and implication of change
    - Robustness – ability to avoid and correct errors
- Useful starting point for GDM and to complement existing requirements

Goal Decomposition Model (GDM)

- Based on Goal-Oriented Requirements Engineering – especially KAOS method
- The model shows relationship between goals and requirements
- The model is of tree-graph, where the root node is a high-level system goal
  - Each lower level contains one or more sub-goals that support goals of the level above
  - The leaf-nodes represent goals that are specific enough to be expressed as requirements
  - Each leaf-node is assigned to the component (either internal or external) that will be responsible to achieve it

Agent Interaction Model (AIM)

- Highlights **how the agents interact and how they assist or potentially interfere with each other in achieving goals**
- Models:
  - Agent actions
  - Information shared with between multiple agents – utilized, trigger or triggered by actions
- Information is categorized into three types:
  - **Message:** Information explicitly exchanged between agents
  - **Knowledge (KB):** Persistent information, stored in memory and used over time. Knowledge can be given a priori (e.g. map of static obstacles) or it can be acquired at runtime and used later (e.g. generated route).
  - **Percept:** information that is observed directly from the environment in near real time through sensors.
    - Includes both raw/unprocessed data or data processed and fused together
    - A camera is an example of a sensor and the images can be processed to detect a human face or obstacles on the road, those can be considered percepts.

Agent Decision Model

- Elaborate how each agent behaves and acts based on available information.
- The models show **when** an agent performs each of the actions assigned to it, and **what** information the agent relies on to perform those actions.
- A behavior is intended to represent an action that is executed over time.
  - Action can be physical actions or computation that takes some time.
- We adapted ADM from finite state machine; however we also take into account that actions take time and the state of the world can change at any time

Example of Application of our Modeling Approach - GAR

- A few examples of relevant GAR:
- **Self-Navigate**: The rover shall autonomously...
  - Provide routes between tasks
  - Provide alternate routes to account for changes in topography
  - Provide alternate routes to account for the presence of obstacles
- **Self-Transfer**: The rover shall autonomously…
  - Receive packages from a “sender”
  - Deliver packages to a “recipient”

Example of Application of our Modeling Approach - GAR

- Supporting requirements for Self-Transfer:

Knowledge: The rover shall have knowledge of…
- Sender location
- Recipient location
- Package type

Awareness: The rover shall be aware of…
- Current rover locations

Monitoring: The rover shall monitor…
- Package stability

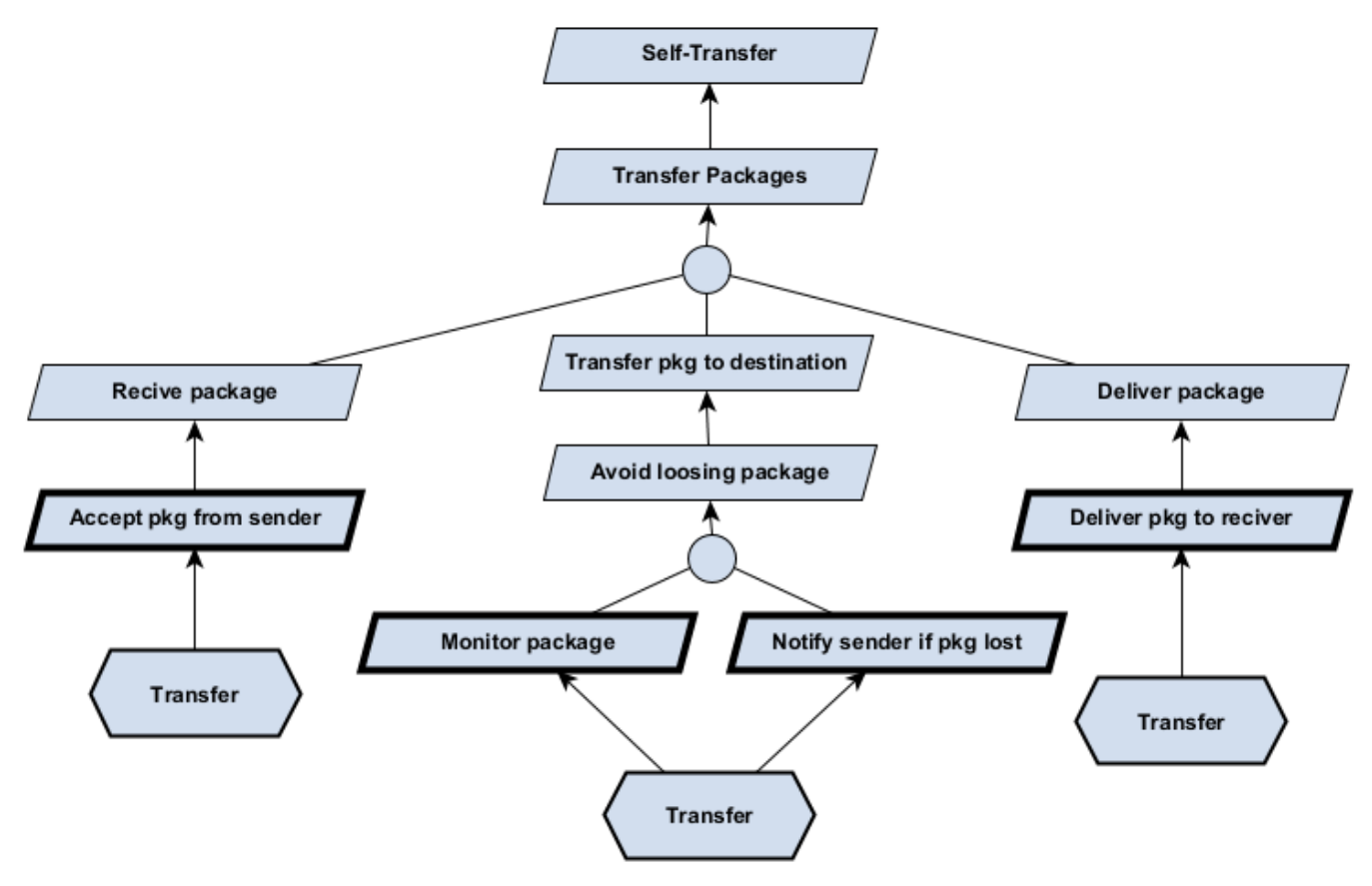
Adaptability: The rover shall adapt to…
- Oddly-shaped packages

Dynamicity: N/A

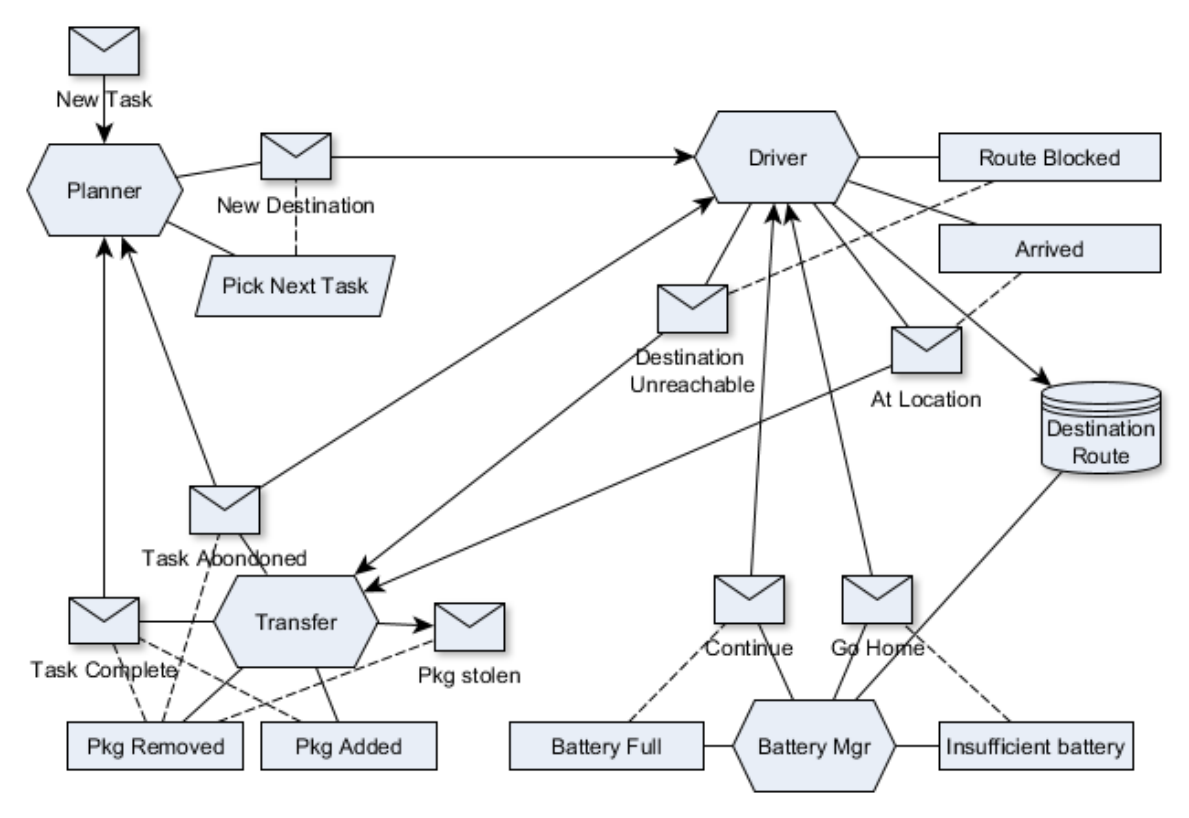
Robustness: N/A

Mobility: The rover shall be able to…
- Transit while carrying packages

Example of Application of our Modeling Approach - GDM



Example of Application of our Modeling Approach - AIM

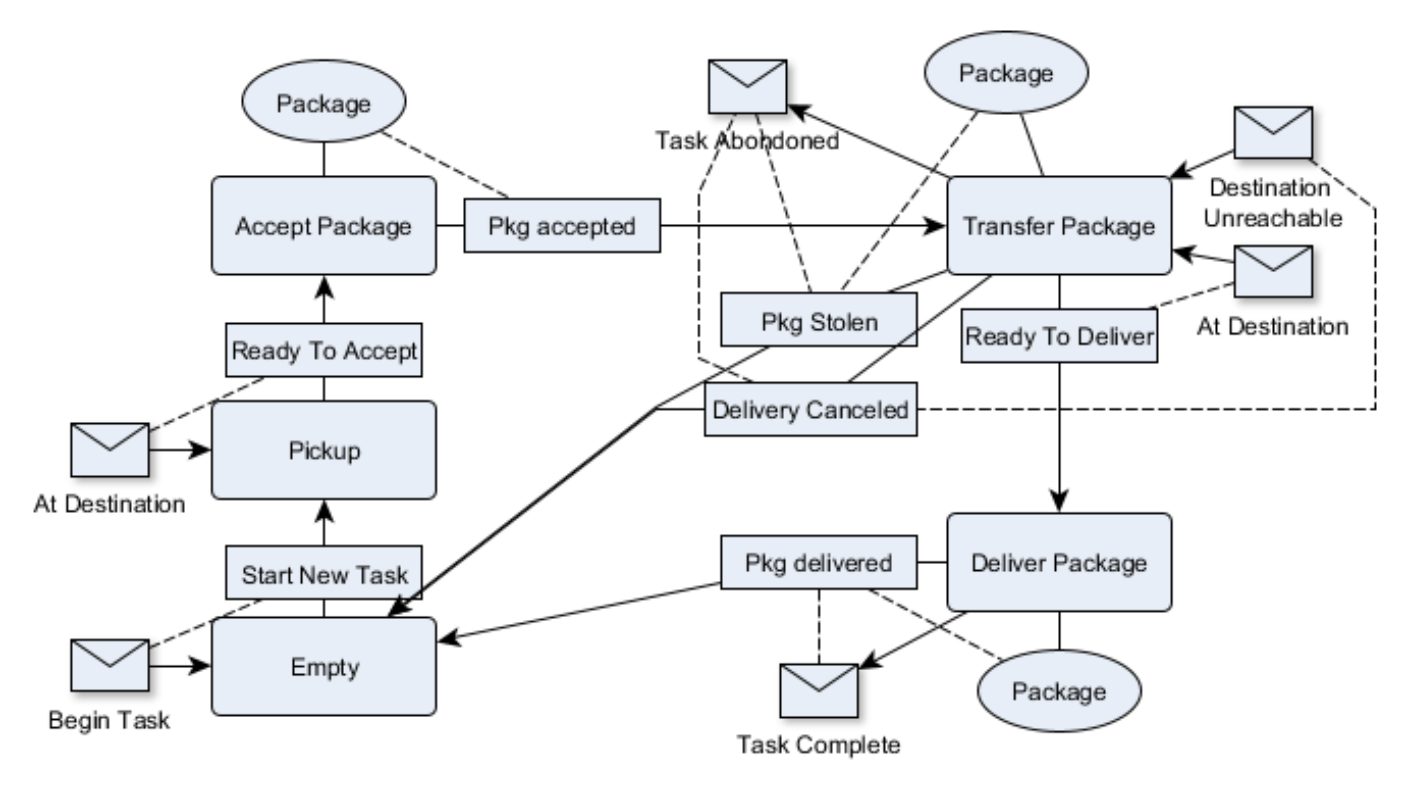


Interactions between the 4 agents identified* -- exclude human-agents such as senders and recipients

*Include more than the self-transfer requirements

Example of Application of our Modeling Approach - ADM

The transfer agent keeps track of packages from pickup to delivery.



Model analysis

- GDM focuses on goals and their relationships, and can be used to asses:
  - Completeness:
    - Have all the system's goal been enumerated?
  - Consistency:
    - For each pair of goal, can they be both satisfied at the same time? If not, they are conflicting goals -- have they been explicitly identified in the diagram?
  - Feasibility:
    - For each goal, is there at least one requirement defined to satisfy the goal?
    - For each requirements identified, has an agent been identified to be responsible to perform the requirement?
    - For each goal, have all possible obstacles to the goal been identified?
    - For each obstacle, is there at least one task identified to resolve the obstacle or to mitigate impact of obstacle?

Model Analysis (2)

- Both AIM and GDM focuses on relationships between actions and information. The models can be used to identify:
  - Completeness
    - Have all the information been identified?
    - For each requirement, is there a corresponding action identified? Reversely, for each action in the diagram, is there a corresponding requirement?
    - For each requirement that describes temporal and causal link, is there a corresponding event identified? Reversely, for each event, is there a corresponding requirement?
    - For each potential obstacle identified (from GDM), is there a corresponding percept to detect it and event to react to it?

Model Analysis (3)

- Consistency
  - If more than one agents acted upon a common information, do the agents have consistent interpretation of the information?
  - If more than one agents acted upon a common information, are their actions consistent with one another?
  - For each state that an agent can be in, is there a potential conflict with another agent's states (e.g., the two agents' states cannot occur together)? If yes, are there considerations to ensure that they cannot be in conflicted state?
  - If an agent acted upon information that is of the nature of knowledge base, is there consideration for ensuring that the knowledge is not stale?
    - What mechanism exists to update the knowledge base?
    - What triggers the update of the knowledge base?

Model Analysis (4)

- Uncertainty (from information)
  - If an agent acted upon information that is of the nature of percept, is there a consideration for possible sensor error or noise which could lead to incorrect decision?
    - If error and noise possible, what is the expected frequency of the noise?
    - What is the impact of acting upon noisy data/incorrect percept?
  - If an agent acted upon information that is of the nature of message (from other agent), is there a consideration for ensuring the integrity and authenticity of the message?

Conclusion

- We have proposed a modeling process that leverages four modeling methods, including ones that have been applied for autonomous system requirements.
- We have applied the modeling process to a case study, which though is not real, still represents non-trivial autonomous system that is relevant for NASA domain.
- The proposed modeling process is still a preliminary work which needs to be further developed.
- While the modeling method in our process accounts for uncertainties, they are mostly implicit.
- To be more useful, the uncertainties need to be made more explicit so that developers and engineers can benefit from understanding risk inherent in the requirements.

Acknowledgement

- This work is funded by NASA Software Assurance Research Program (SARP) 2018-2019