A Sample Application: Modeling Command and Control Centers (Part 11)

Dr.Çağatay ÜNDEĞER

Öğretim Görevlisi
Bilkent Üniversitesi Bilgisayar Mühendisliği Bölümü

e-mail : cagatay@undeger.com
cagatay@cs.bilkent.edu.tr

Modeling Command and Control Centers (Outline)

- Introduction
- Generic Simulation Engine
- C4AT (C4 Analysis Tool)
  - Communication Devices
  - Environment Design
- Agent Architecture
- Conclusion & Future Work

Objective

- Developing a simulation toolkit for analyzing activities of command and control centers
- Simulation of peace time activities of the strategic and operational level command and control centers.
- In order to analyze efficiency of the organisational structure.

Modeling Complex Systems

- Compex systems that have many actors and their interactions often becomes too complex for a mathematical model.
- Therefore, agent-based modeling is a tool to study these kind of systems.

Properties of Software Agents

- Capable of acting and modifying their environment.
- Communicate with other agents in the environment.
- Have intentions.
- Control some local resources.
- Capable of perceiving their environment.
- Have only a partial representation of their environment.
- Possess skills and offers services.

Modeling Command & Control Centers

- Command and control centers are complex systems
- It is too hard to model with conventional modeling and simulation techniques
- Therefore, agent-based approach has been employed to study command and control centers
- A generic agent-based simulation engine

Generic Simulation Engine

- First, a generic simulation engine independent of the tool was developed.
- The generic simulation engine is actually a single software class (***TBtnEngine***), which is used as a base class and inherited in order to create the analysis tool, **C4AT.**

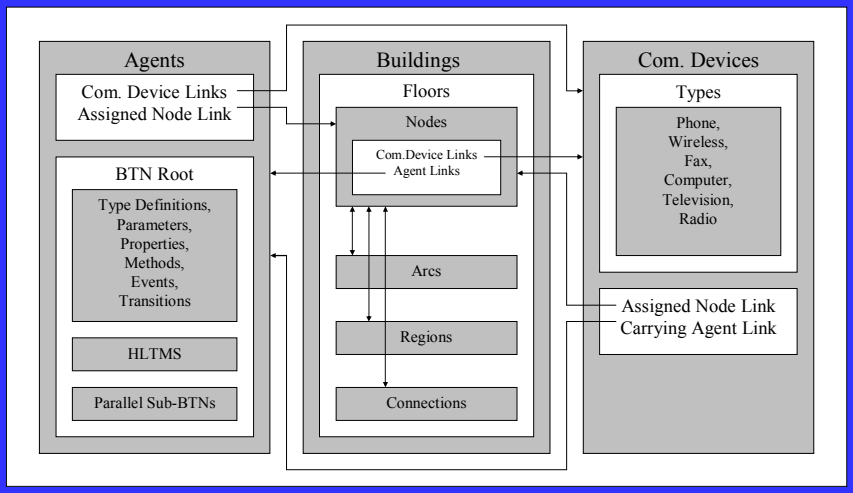
The Functions of *TBtnEngine*

- Enabling creation and modification of the project environment and the scenario.
- Managing environment and scenario files.
- Defining scenario agents including their behavioral characteristics and tasks.
- Accepting insertion of any project specific resources.
- Allowing agents to access all the resources of the project through a C-like run-time interpreter.
- Running scenario with a selected time management mode.

C4AT Simulation Engine

- With the customized engine (C4AT):
  - The geographical location of the scenario can be set.
  - The buildings of command and control centers including their interior can be designed.
  - The communication devices can be introduced.
  - The agents can access all the resources of C4AT (buildings and communication devices).

C4AT Architecture



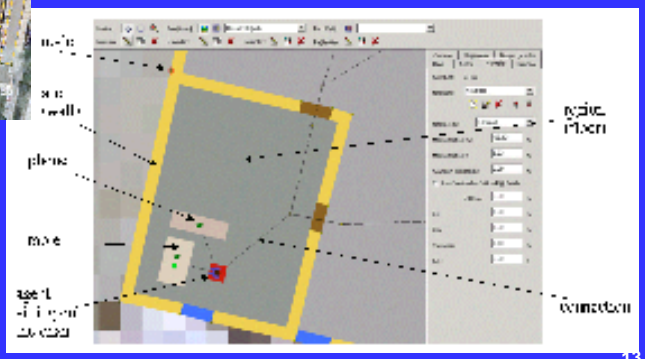
Communication Devices

- The communication devices can be categorized as:
  - phone, radio, fax, computer, and multimedia (television, radio, newspaper, etc.).
- Phone is the most preferred communication device in peace time operations.
- Consequently, we started the development of communication devices with the phone.
- Phone States: available, waiting for dial tone, ready to dial, calling, connected, busy, disconnected or ringing.

The Environment Design

- The simulation environment consists of a set of buildings and their interior.
- A building editor is developed.
- Buildings are designed by creating each floor and their connections with other floors.
- Floors contain nodes, arcs, regions and connections.

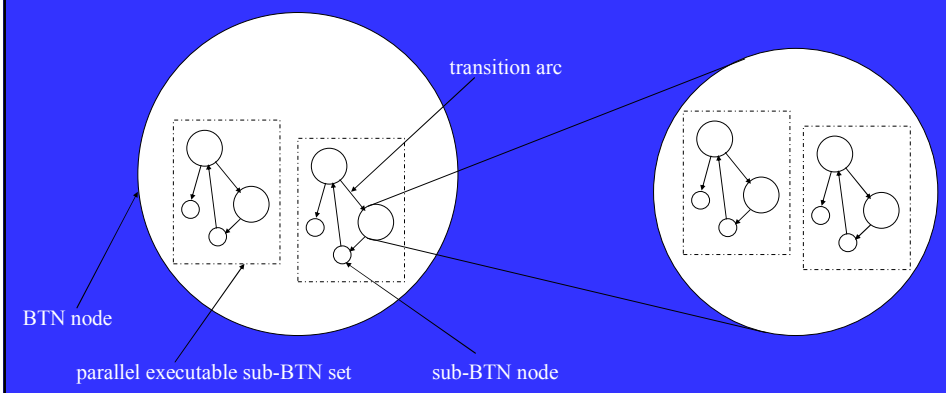
A Sample Building Design



Agent Architecture

- The behaviors of the agents are modeled using:
  - Behavioral Transition Networks (BTN)
  - A sub-feature defined within BTN structure called High Level Task Management Script (HLTMS).
- BTNs are just a specialized approach based on State Transition Diagrams.

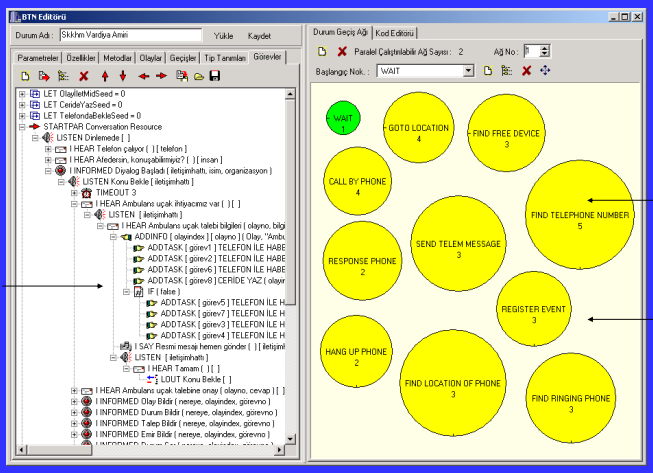
General Architecture of a BTN



Modeling Agent Behaviors

- Each agent has a behavior assigned to him, which is defined in a single BTN node called **root BTN**.
- First level BTNs, the ones directly owned by root BTN, are generally used for action modeling such as calling by phone, going to a location, etc.
- Action BTNs are fired by HLTMS.

A Root BTN Sample



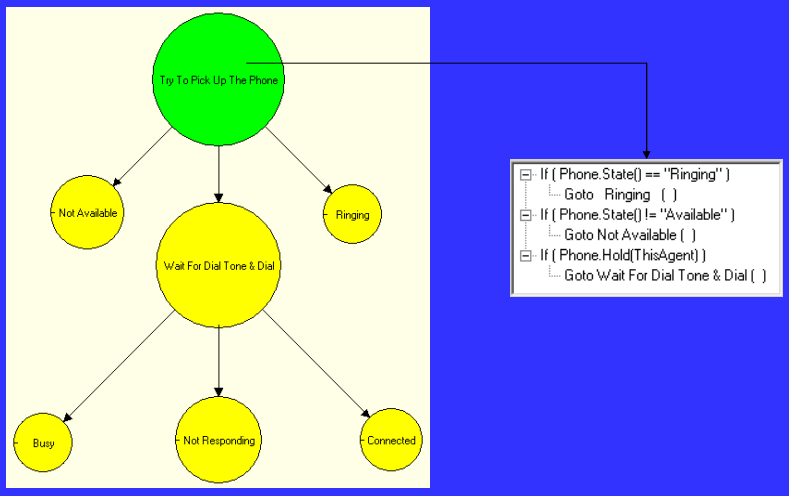
HLTMS definition window

action BTN node

parallel executable sub-BTN set

Call By Phone Action

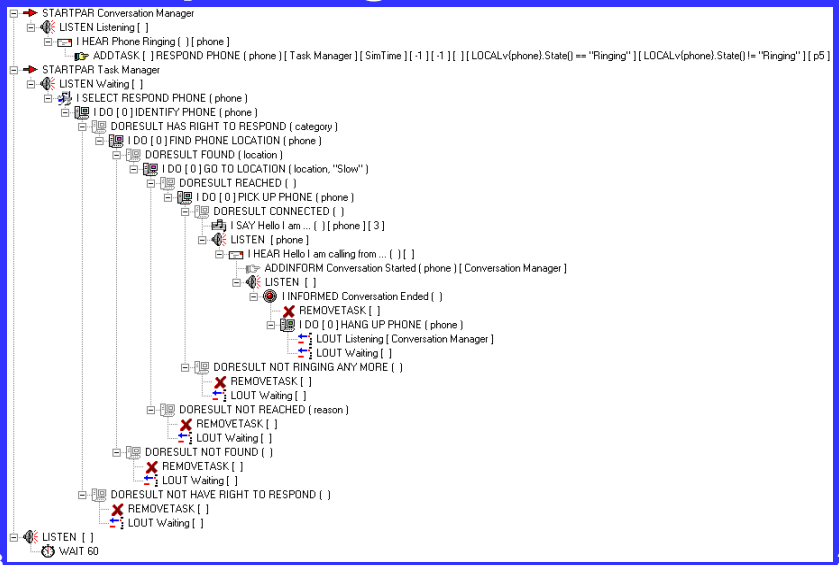
transitions of "Try To Pick Up The Phone"



High Level Task Management Script (HLTMS)

- HLTMS is a hierarchically defined script
- The statements of HLTMS can be executed sequentially or in parallel.
- Some of the statements:
  - STARTPAR: Starts a parallel execution
    - ADDINFORM : Inserts an information message into a parallel execution
    - ADDTASK : Inserts a task into a parallel execution
  - I SAY : Sends a voice message to an agent or object (phone)
  - I DO : Triggers an action
    - DO RESULT : Captures the result of an action
  - LISTEN : Starts listening for perception messages, information messages and tasks
    - I HEAR : Enables receiving voice perception messages
    - I INFORMED : Enables receiving information messages
    - I SELECT : Enables selecting from tasks. The task with highest priority is always selected
    - LOUT : Backtracks to the start of a specified LISTEN statement

A Sample HLTMS for Responding a Phone Call



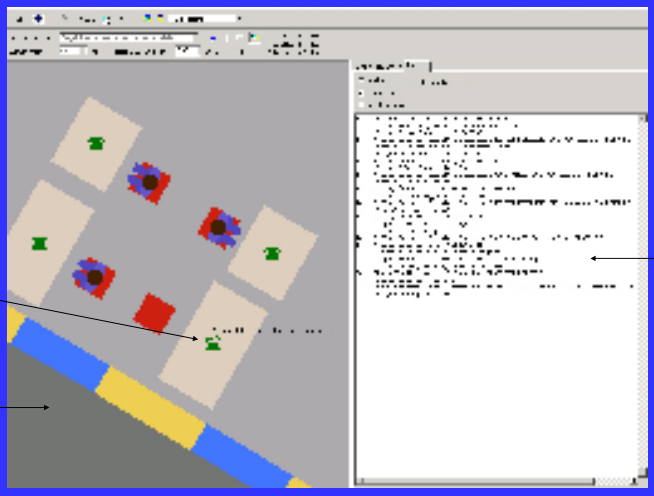
Task Distribution

- A simple but effective model, which reflects nature of command and control hierarchy.
- An agent informed of a task (event, request, order) generates a set of sub-tasks to meet the requirements of the main-task.
- Following the task decomposition, additional sub-tasks for task distribution management are inserted.

Running the Scenario

- The *TBtnEngine* allows selection of time management metodology:
  - Event based
  - Real-time
    - Unlimited time steps
    - Constant time steps
    - Upper bounded time steps
  - Constant time interval

A Snapshot of the Simulation in Execution



Conclusion & Future Work

- Proposed a simulation framework and a simulation tool for modeling command and control centers.
- Realized a generic simulation engine and a customized engine for C4AT.
- Developed an agent-based system, which uses:
  - Behavioral Transition Networks,
  - A proposed approach called High Level Task Management Script .
- The first version of our implementation has given promising results.
- Currently studying on the tool for defining agent behaviors and tasks to improve the system.