Organizer's Corner: Some challenges for going beyond using Models@RT to Building Them

Kirstie L. Bellman, Ph.D.

TopcyHouse Consulting

July 18, 2016

Models@RunTime

Consider why we have System of Systems/complex distributed systems, etc.

- Local, specialized machinery -> **connected**
- A repertoire of diverse capabilities to select from -> **adaptive, context-sensitive, robust**
- Relatively open; can add new services and tools -> **agile, responsive**
- Operates at many levels of precision from ‘factory floor’ to ‘enterprise’ -> **comprehensive, powerful**
- MODELS@RT essential to some of this adaptive, context-sensitive, and robust behavior
- BUT these very properties make it difficult to design a priori sufficient, updatable models (ones that stay relevant)
- Making it harder may actually help: **Let’s put the processes into place to build models, update models and verify and validate models at Run Time.**

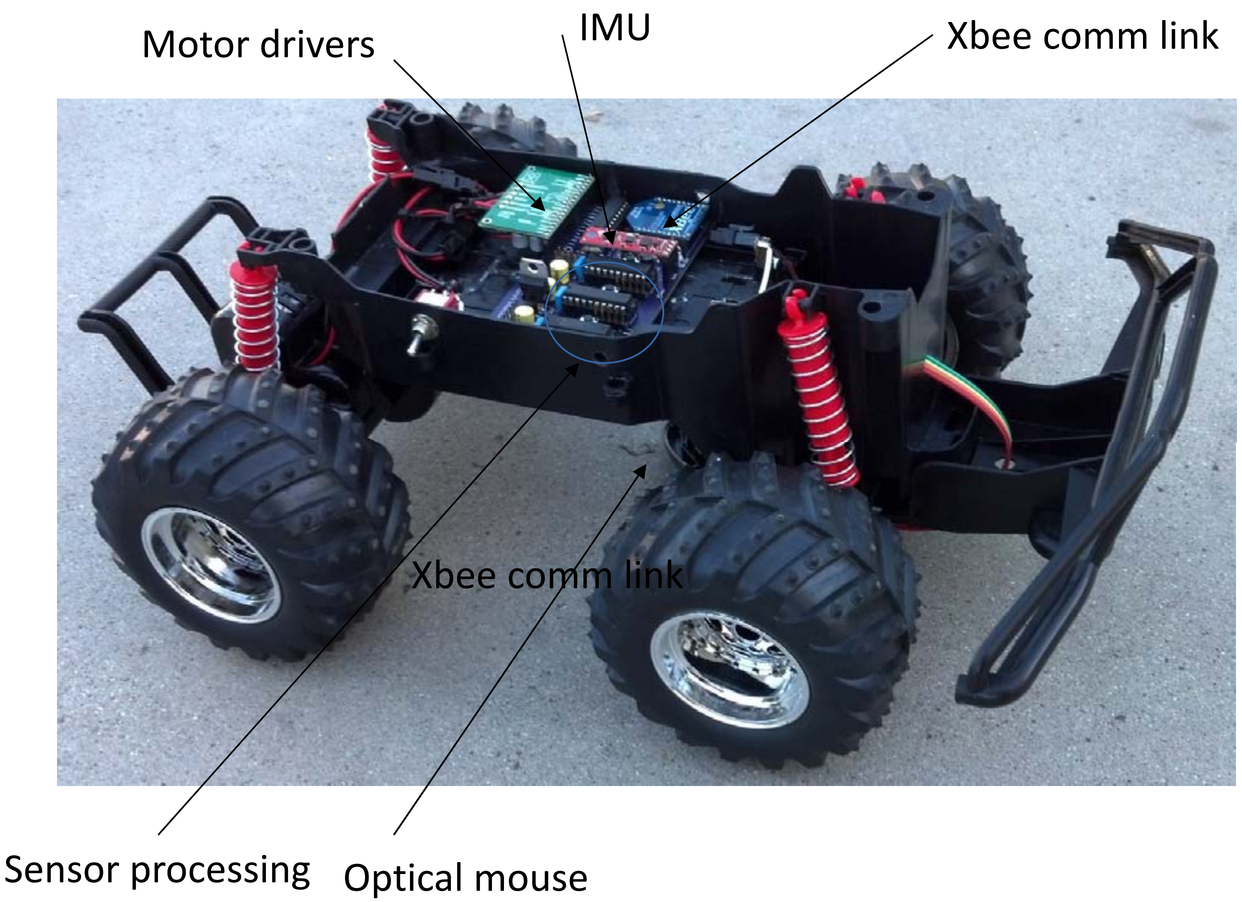
Purpose of this discussion: Stimulate our thinking on where we need to go next

- This morning we saw ongoing work over two decades for how to dynamically update models and alter the collection of additional data (Frederica Darema)
- How do we go from running models@runtime -> to having systems that build models and self-models
- Will mention three key concepts to start the discussion:
  - Active experimentation
  - The role of ‘others’ in building models
  - The role of archives, memory, and “continual contemplation” supported by new mathematical methods for pattern discovery

Context for discussion: CARS TESTBED

- At California State Polytechnic University, Pomona, directed by P. Nelson, diverse robotic cars use the reflection and adaptive capabilities of *Wrappings*
  - Studies the development of self-models, the dynamic substitution of different game-playing approaches and adjustments in the uses of sensors and other resources
  - Purposely diverse robotic cars with different sensors, different capabilities (like speed and turning ratios)
- Robotic cars participate in games with the same control infrastructure, as well as adjusting for sensor & part failures.
  - These games designed to require a broad range of different and even conflicting game-playing strategies.
  - Games: Follow the leader, tag, soccer practice, and push the box to a goal (requires several cars)

CARS Example Vehicle



Fleet of toy cars
•very non-ideal vehicles compared to the performance we desire
• they are not identical
•communication, sensing, and computation are added

CARS Testbed

Purposely DIVERSE capabilities in each car. Each has many sensors, such as video, three axis gyroscope, compass, three-axis accelerometer, a mouse sensor used for distance traveled, bumper pressure sensors, ultrasound, and infrared sensors

Each car modifies the parameters of the game-playing strategies based on their ‘experience’ with their capabilities (e.g., their own turning ratio, their maximum speed, their fuel efficiency, etc.)

A CARS robot needs to learn their self-models in a realistic "world"

- Animals invest energy and time
  - 'Exploratory behavior' (learning its environment)
  - Play (exploring one's own capabilities within that environment)
- Complex Systems need play time
  - Try out resource integration in different operational contexts
  - Learn its own **preferred** strategies, combinations
  - Learn appropriate rules for & parameterize its self models
  - Push one's performance limits in a safe place…a form of negative testing and stress testing
  - Get corrections as needed from human "coaches"
  - **Fire drills**: what to do when errors/problems/unexpected conditions

A SoS needs to learn their self models in a realistic "world"

- Complex Systems need play time…A SoS needs play even more!
  - An open system means new players need to be integrated to the 'team'
    - Make sure that new features have appropriate constraints
    - Learn signs and symptoms of failure modes
  - **Continually** develop the appropriate 'reflexes' or patterns for "safing" the system when integration & parts fails
  - **Continually** develop better specifications and constraints on the use of resources
- **This experience and reflection can/should be communicated and shared with designers and users**

This is all well and good until we see how hard it is to build models

- In current CARS, we simplify by having essentially templates for parameters that the individual cars need to fill in (like their speed going uphill, their turning ratios)…figuring out the right parameterization for the model developer is hard enough at times.
- We theoretically can figure out how systems can derive a new parameter influencing results….the example of the “breeze” during CARS soccer practice.
- But sometimes we need an ‘external view’ and the next example shows us the importance of “others” even for self-models

When another set of eyes and hands can really help!

We assume that some things will not happen, but they do!

The Role of ‘others’ in building models (1)

- Surprising need for others feedback – even for self models – learn own body, capabilities, style; learn word for states (oh you’re just tired; I know you are angry….)

The Role of ‘others’ in building models (2)

- Others add scope to our sensors, our experiences, our reasoning – correcting a swing, a proof, an attitude
- Some directly communicated; some by imitation and modeling

The Role of ‘others’ in building models (3)

- Many different styles of interaction, learning, and collaboration
- Many contexts and situations

Archives, Pattern Discovery and Continual Contemplation

- “**Continual contemplation**” (Landauer and Bellman, 95)
  - Systems uses instrumentation for lots of data about the system itself;
  - Reflective processes includes monitoring internal and external data sources, reasoning about them, and doing appropriate actions as a result
  - Self-reflective processes are used in a continual fashion to discover new things about the system, its behavior, the mapping of its capabilities into the ‘real world’ and the status of its goals;
  - The Wrappings approach already makes a step in this direction with the “continual contemplation” of the resources by *Wrappings* infrastructure
- Up to now “Continual contemplation” is a background process over all data and process behavior.
  - Tip relevant internal and external participants about anything interesting
  - Constantly check to see that the right thing is still being done and is still having the effects one expects in the operational context.
- Reflection results can be learned, archived, communicated, and integrated with other reflective processes

But much more needs to be done for continual contemplation that supports model building

- Reflection can become one of methods for pulling out relevant information from data that becomes knowledge and that knowledge into models
- If add methods for noticing unexpected connections and pattern discovery, have some basis for model changing and eventual model building
    - Need wide range of diverse old and new mathematical techniques from grammatical inference, time series analysis, manifold discovery, topological data analysis using homological methods…and much more

To support model- building means understanding more about our own human capabilities

- We need to understand then develop computational approaches to:
  - Integrating Bottom up data mining and pattern discovery methods with complementary top-down or ‘given’ knowledge
  - Trending whether some behavior is still compatible with success criteria and whether several ongoing behaviors will result in future insurmountable incompatibilities.
  - Developing new intermediate results in computational processes in order to help trend the impacts of ongoing computational processes and model performance
  - Need diverse logical methods for handling partially satisfying goals or partially meeting requirements.
- To build and evaluate models, we need to form a foundation of evidence that helps the system AND helps us decide where model-building systems are effective – and where they are not.
- **We must not build systems that we cannot understand, monitor, or trust.**