UCCM (User Centered Communications Management)

Problem:
The BAMS (Broad Area Maritime Surveillance UAS) will be the Navy's next-generation remote surveillance platform. It will be an airliner sized drone packed with advanced radars, video and IR cameras, and various signals intelligence equipment. It will fly at around 60,000' but can descend below the cloud deck for better viewing.

The BAMS will have three radio channels for sending sensor readings back to the ground station. A Ku-band satellite link, when available, provides bandwidth of up to 10 Mbps. The bandwidth is more typically 1-2 Mbps. A line-of-sight, C-band channel runs at slow DSL speeds. A VHF band channel runs at modem speeds. The bandwidth available on any channel varies quite dynamically due to environmental conditions, and does go to zero. For example, there is no guarantee an appropriate communications satellite will be in range, or that time on a commercial satellite can be purchased on short notice.

The sensor data itself causes problems. At one extreme, the omnidirectional emitter detector produces reports around 1 KB in size. If BAMS is flying over an urban area and is tuned to a cell-phone band, the sensor produces O(1000 readings/sec). The visual camera can produce images up to about 26 MB in size, and can generate them as fast as the operator can push the button. There is also a video mode.

The fundamental problem is that the BAMS can easily overwhelm all available channels with data. A FIFO strategy is clearly not optimal. It is a truism that half the video images will be useless due to clouds and fog. It is a good idea to let a 26 MB image start to send on a 400 Mbps channel, and hold up thousands of other sensor readings while it transmits? The goal of UCCM is to manage the communication flow to maximize the value of date delivered, as perceived by the operators.

UCCM Solution:
UCCM manages a prioritized queue of transmission requests. Each time the radio finishes sending its last transmission; UCCM supplies it with the next that has the highest priority for this point in the mission, for this operator. Priorities are assigned by heuristics that take into account factors like the size and type of the transmission, the age of the request, the operator's intent (either inferred or declared), and the generic importance of the type of transmission.

A second set of rules managed the queue itself. When should requests be repriortized? The age of requests increases over time and is an important factor. Depending on the type of sensor that generated the request, the age-priority factor increases for a short time and then decays. At some point, if the transmission hasn't been sent, it is probably not worth sending. We needed to implement a selective, incremental reprioritization strategy as this extra work significantly burdens the processor. Rules determine when requests should be purged. Other rules implement strategies such as, "This data need not be sent in real time. Send only on a bandwidth-available basis."

A third set of heuristics will manage the channels. Each sensor is assigned to a channel, allowing for some parallel transmission. When a channel's bandwidth drops too far, the channel's request queue can be redistributed to the other active channels. Or it might be

better to wait and see if the bandwidth comes back soon. Rules are an excellent medium for encoding multiple, competing strategies.

Implementation:

Most of the decision logic falls very naturally into the rules paradigm. Even communications with the operator's GUI is very much an event-based process (in the traditional sense, not in the CEP sense). UCCM provides a tailored developer's GUI and compiles into ordinary Drools for execution. The project started before Guvnor was announced—we might have just adopted that otherwise. Our GUI provides forms-based rule editing for subject-matter experts who do not know Java. It allows the developer to define event classes (our terminology) that are compiled into POJOs. There are facilities to start, pause, resume, and stop execution, and to examine/change the state of working memory. And finally, there are many pages to collect and display various statistics and metrics about the operation of the system.

Rules are built by hand, so there are a small number of them compared to the applications that generate 10,000s of rules. We anticipate running UCCM on a processor on board the aircraft, but we don't yet know what the CPU budget will be. We can just barely run UCCM and a data generator full out on a Windows 7 laptop. We anticipate that Drools should keep up with demand in the deployment version that strips out the developer's overhead. The worst case scenario envisions 1300 new transmission requests/sec at low bandwidth so that the priority queue requires a great deal of management.

A UCCM rule server onboard the aircraft seems like an excellent fit. We use the H2 memory based DB and Drools has a small footprint. UCCM is installed as a simple intermediary between the sensors and the radio. There are buffers between sensors and the UCCM working memory, and the aircraft's radio subsystem requests the next transmission from UCCM when it is ready. We assert UCCM need not run as a hard real-time process as is normal on aircraft, so Drools can be used as-is.

Status:

We just completed Phase II. We designed and implemented a set of prioritization rules and another for queue management. We assume a single channel for now and need expert input for how to balance multiple channels. We built a very simple operator's GUI but have mocked up a far more capable version. These extensions obviously await a Phase III.

The heuristics do a good job of adapting the flow of information to changing circumstances. Expressing them in rule form allows them to be effective and easy to change without being overly arcane. The rules prune out transmissions that have been rendered obsolete or whose value is not worth the overhead of sending them now. The operators receive data that is timely and valuable, with a good balance of large and small data transmissions. And most importantly for long term success, the rule format allows the logic to cleanly express competing strategies and special cases. Naval experts will be able to quickly express and test new strategies, so that UCCM can grow with the BAMS platform.

Principal Investigator: Allan Terry (aterry@teknowledge.com)
This work was supported in part by a grant from the Naval Air Command.