



Predictable End-to-End Timeliness in Network Centric Warfare Systems

E. Douglas Jensen

781-271-2514

jensen@mitre.org; andersoj@mitre.org; barkley@mitre.org

MITRE Sponsored Research

Problem



- ❑ The DoD needs to be able to strike mobile and emerging targets in minutes
- ❑ Assessment of current kill chain capabilities
 - We can only do this now if conditions are perfect
 - Question: What is the ONE unifying measure of all combat activities as they relate to the kill chain? Time. Speed is king – focus our efforts to make F2T2EA process faster
- ❑ But “time” does not mean only “real fast” – just as importantly, it means **“on time: do the best possible things at the best possible times (not the wrong things fast)”**
- ❑ Thus, dynamic resource contention by concurrent time-critical activities (e.g., threads in kill chains) must be resolved according to their end-to-end timeliness QoS and other requirements, to maximize MOPs and MOEs
- ❑ *Dynamic systems require stronger assurances than from testing*

Background



- ❑ In addition to “real fast,” priorities are the most common mechanism for attempting to resolve contention for resources
- ❑ Priorities are relative, so dynamic systems cause a domino effect of priority changes
- ❑ Priorities cannot denote orthogonal urgency and importance, resulting in contorted priority assignments
- ❑ Urgency has to be mapped in a lossy way from actual physical times to a short sequence of small integers
- ❑ **These limitations of priorities in deployed systems have repeatedly been shown to**
 - increase test and integration time/costs
 - degrade system MOEs/MOPs**in unpredictable ways**
- ❑ **Priorities are severely deficient for *assured* “on time: do the best possible things at the best possible times”**

Objective



- ❑ **Help maximize NCW overall mission operational effectiveness**
 - by resolving resource contention among dynamic time-critical activities**
 - according to application- and situation-specific end-to-end timeliness QoS criteria**
- ❑ **Create algorithms for dynamically managing network, sensor, and processing resources,**
that provide *formal* assurances about the timeliness QoS of time-critical activities
- ❑ **Implement algorithms and simulation experiments in Java, real-time Java, and distributed real-time Java**
- ❑ **Demonstrate relevance to distributed collaborative NCW systems engagements**
 - collaborate with a DoD program**
 - publicly releasable PC- and web-based demonstration and tool**

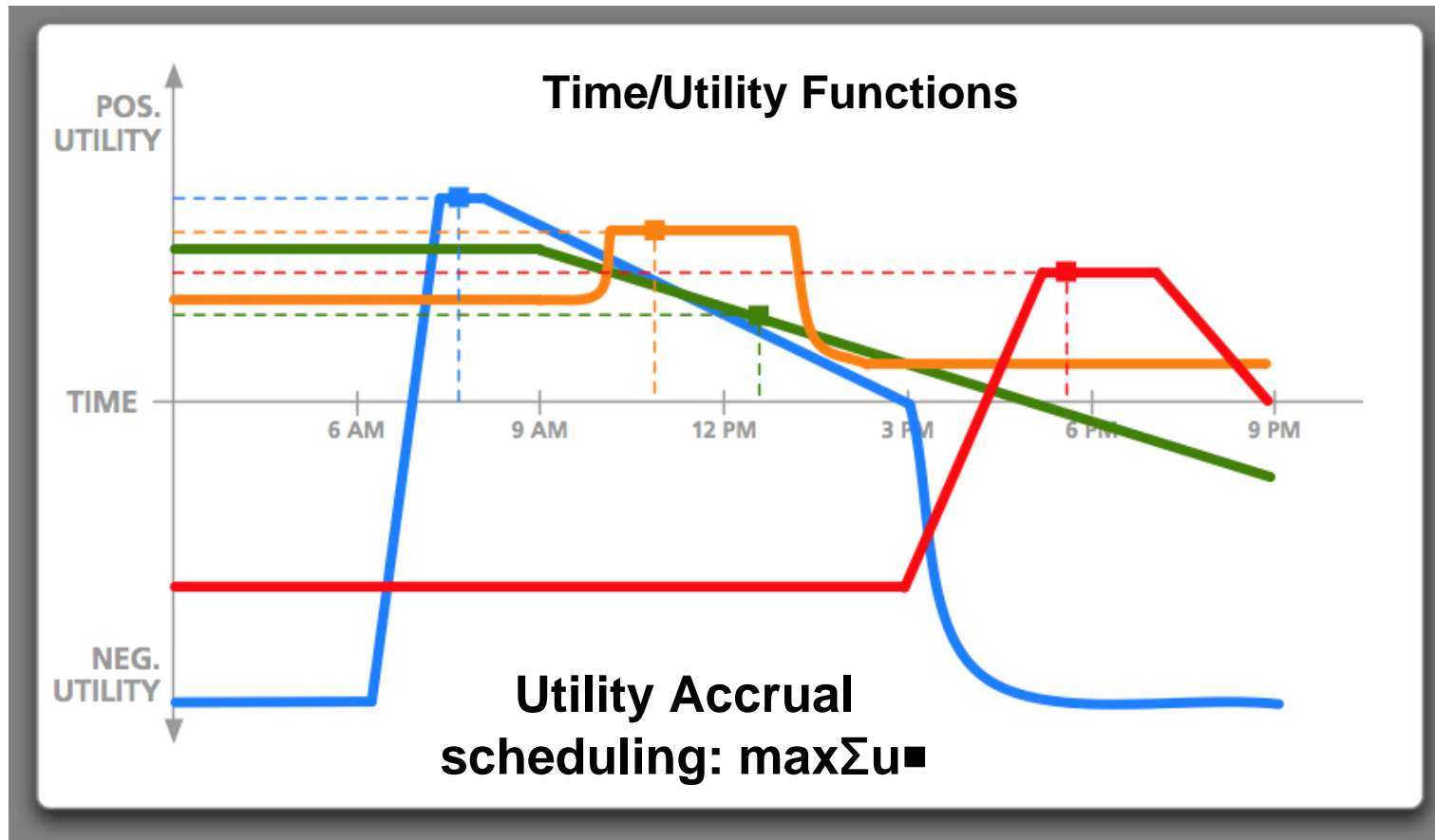
Activities



- **Invent, simulate, implement, and evaluate new resource management theories and technologies for users and designers to**
 - **formally reason about end-to-end timeliness QoS of concurrent time-critical activities (e.g., mission threads in kill chains)**
 - **manage resources accordingly**
- **Design and implement a notional “Missile Defense” video game style system that**
 - **allows a wide variety of conventional and new resource management algorithms to be developed and employed, and their effectiveness compared, using application domain MOEs/MOPs (number of leakers, etc.)**
 - **allows highly configurable coastal air defense scenarios**
 - **is in Java so runs on PCs and web sites (e.g., mitre.org)**

Highlight

In our work and in our life, we all use the concept of the *utility* gained or lost, based on when we complete an activity – we can manage sensor, network, processing, etc. resources that way

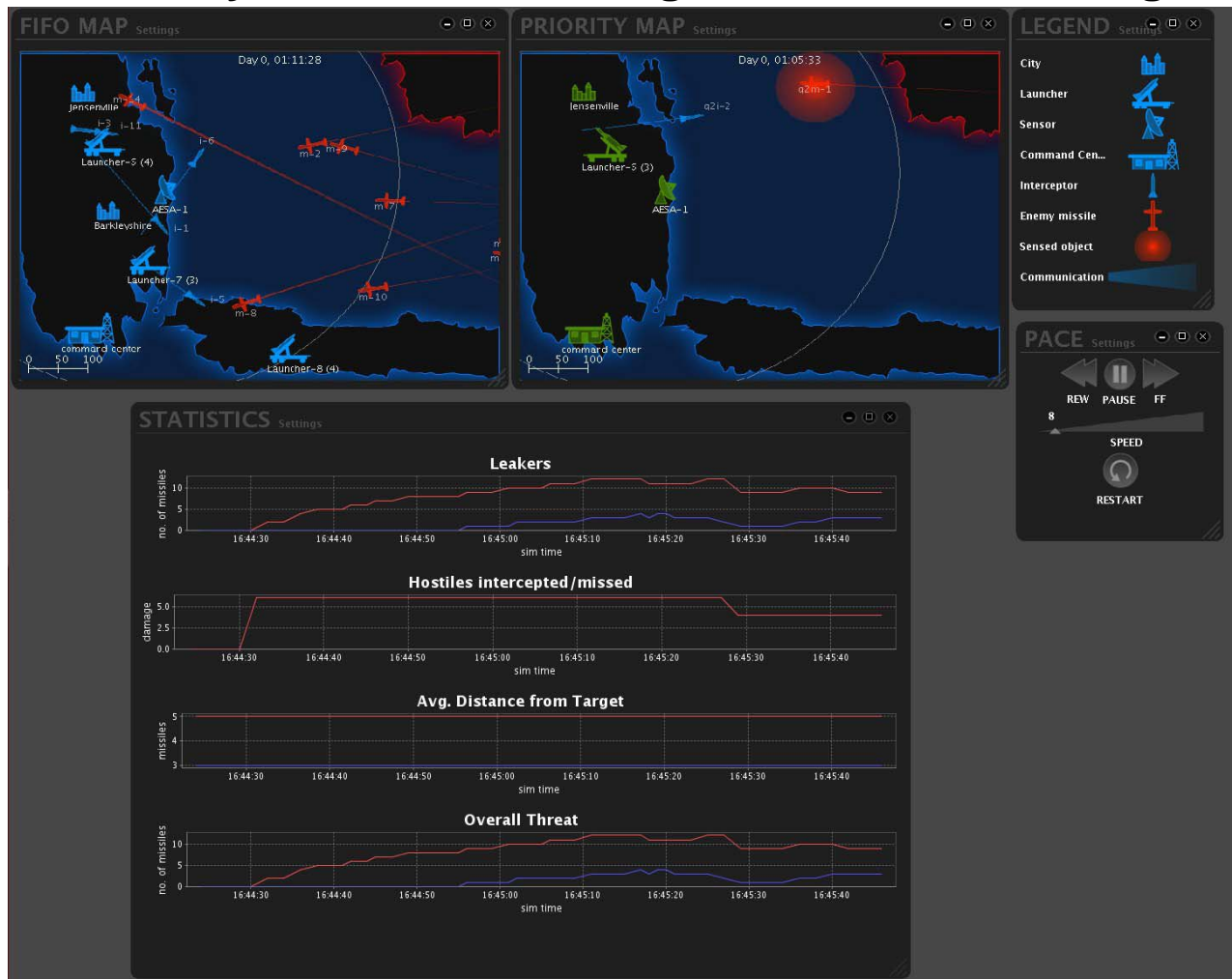


Maximize total utility by doing the right things at the right time

Highlight/Demonstration



A demo and tool for comparing the MOE/MOP of time/utility function, utility accrual scheduling, with that of other algorithms



E.g., TUF/UA algorithms have fewer leakers

Impacts



- ❑ **Users and designers will not be limited to “real fast”**
 - constrained by size, weight, power, upgrade schedules
- ❑ **Users and designers will not be limited to the artifact of priorities to represent urgency and importance**
 - thus reducing T&I time and costs, and improving MOEs/MOPs
- ❑ **Users and designers will not be limited to depending solely on testing of dynamic time-critical systems**
- ❑ **Users and designers will – for the first time – have ways of formally reasoning about MOEs/MOPs as affected by end-to-end timeliness QoS of concurrent time-critical activities (e.g., mission threads) –**
 - and thus have ways of managing resources accordingly
- ❑ **Impacted program: USAF E-10A/MEP**
- ❑ **Prospective impacts on almost all combat and surveillance systems, and systems of systems**

Future Plans



- **Continue design and development of the notional coastal air defense demo and tool for creating and evaluating algorithms to manage sensors, networks, and processing**
- **Expand the air defense scenarios and their properties**
- **Engage DoD programs to collaborate with us to experiment with that demo/tool**
- **Complete the remaining resource management theory planned for this FY**
- **Describe prospective fall-out MOIEs and follow-on MSRs**
- **Document results, and enhance MITRE technical stature, with publications in high quality refereed professional society conferences and journals – and in a final MITRE technical report**