

A New “Availability-Payment” Model for Pricing Performance- Based Logistics Contracts

**A. KashaniPour, X. Zhu, P. Sandborn, Q. Cui
University of Maryland, College Park**

**AFCEA Acquisition Research Symposium
"Creating Synergy for Informed Change"
May 14-15, 2014
Monterey, CA**

Grant number: N00244-13-1-0009



Research Objectives

- Adapt and extend “availability payment” concepts currently in use for Public Private Partnerships (PPPs) to contract design and pricing for Performance-Based Logistics (PBL) contracts
- Availability-Based Contracts (a subset of PBL)
 - Decision making (and contractual obligation) with availability as the performance measure
- Contract Availability Requirement Development
 - Using public-private partnership models
 - **Goal: do not overpay the private sector, while at the same time minimizing the risk that the asset will become unsupported**



Availability

Availability is the probability that system will respond to demand;
it is the outcome of the logistics

$$Availability = \frac{Uptime}{Uptime + Downtime}$$

Increase

Decrease

- Reliability of Parts
- Hot Redundancy Architecture
- Overhaul/Perfect Maintenance
- Avoiding Failures
- Fault Tolerant Control
- Sensor Networks
- Demand Prediction
- Condition-Based Maintenance (or PHM)

Expensive

- Rapid Fault Ramification
- Optimal Switching
- Redundancy
- Maintenance Time
- Fill rate, Inventory Lead Time, CWT
- Fault Diagnosis (Detection, Isolation)

Less Expensive (?)



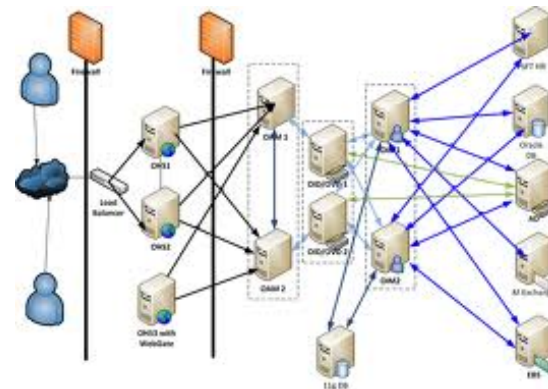
Availability-Based Contracts

Customers of availability-centric systems are entering into “availability contracts” in which the customer buys the availability of the system or process (rather than actually purchasing the system itself)

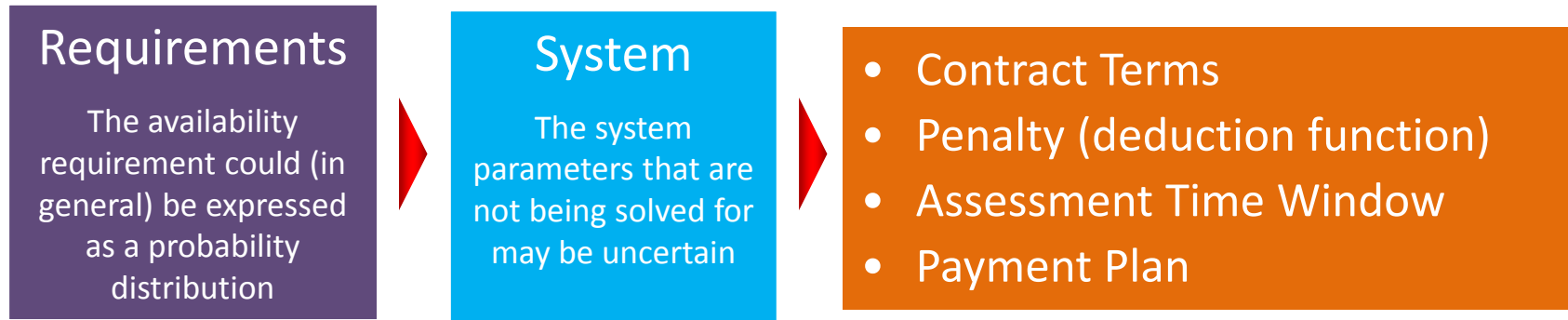
- **The Armidale Class Patrol Boat Project**
 - “...deliver up to **3600 sea days per year** **fixed cost in-service support contract** to maintain the **availability** of the specified capability requirements.”
- **High Availability Server Industry - Microsoft**
 - Availability is driven by how quickly a fault can be recovered from by **switching to a redundant system**
 - The logistics of sparing is not part of the problem
 - Design for availability means minimizing the switch over and switch back times
 - Time to repair is not part of the analysis



Austal Corporation <http://www.austal.com/>
accessed: 9/1//2011



Problem Statement



Gaps in the State-of-the-Art

- The dependency between parameters is neglected, e.g., reliability/demand and level of maintenance, or temporal dependencies in data
- End of support (EOS) date is assumed to be fixed (it is not)
- The learning process is neglected
- Scalability and risk allocation is not addressed
- The flow down of this process into the supply chain is not addressed

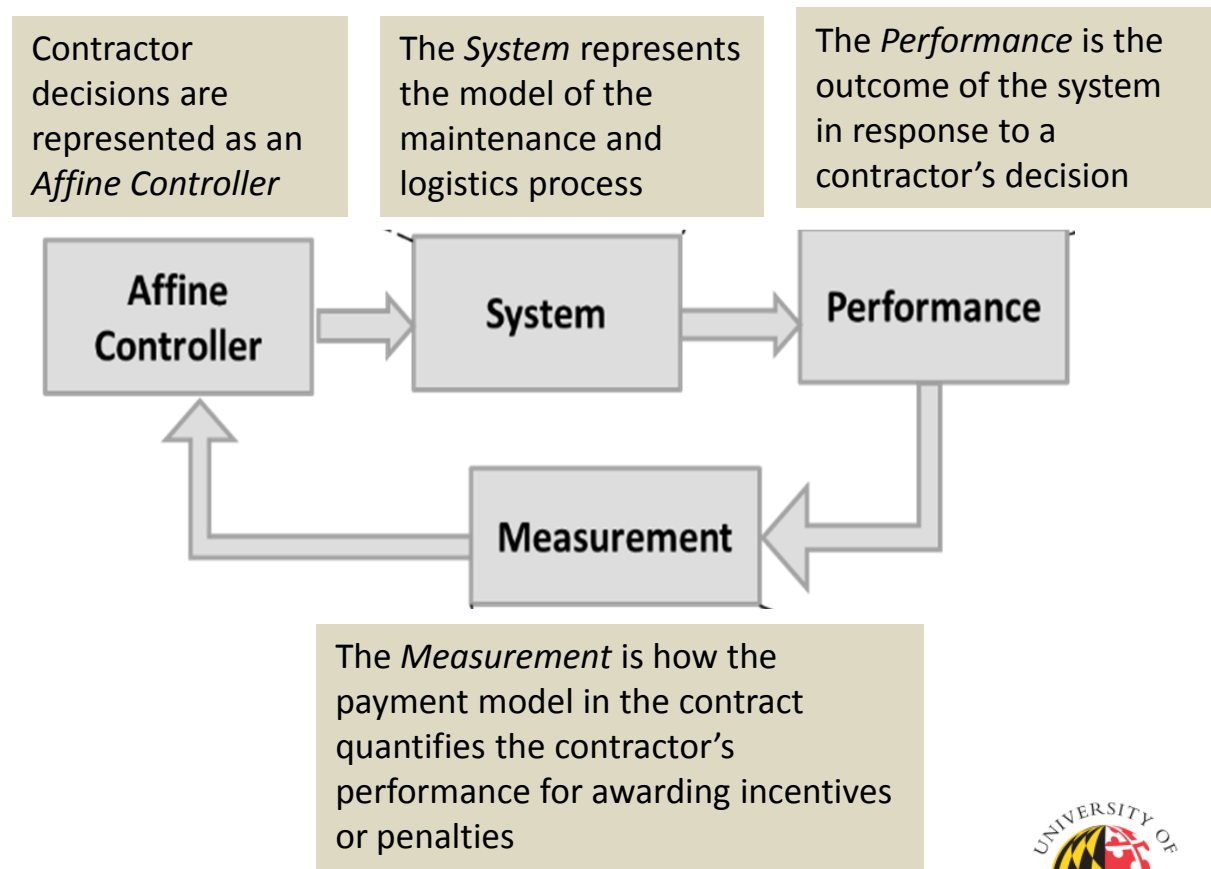
There is little science behind the determination of an availability requirement in a contract



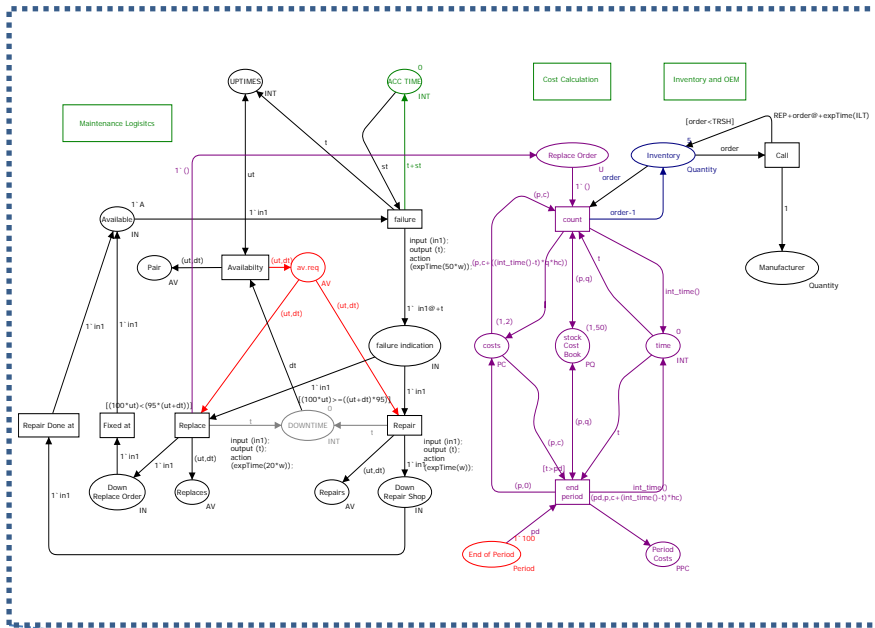
Model Design

The goal of the analysis is to insure a minimum level of availability at all times. The outcome-based orientation of our problem places an emphasize on selecting the proper time frames to evaluate the performance. Meanwhile the nature of reliability and maintenance actions are generally event based.

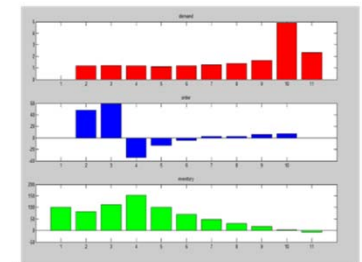
- The model involves the integration of the event-based structure (demand generation) with a time-based controller
- The time-based controller uses the historical demand data in equal periods of time to determine new order sizes
- Demands are generated by a discrete-event simulator that simulates the behavior of the system in time



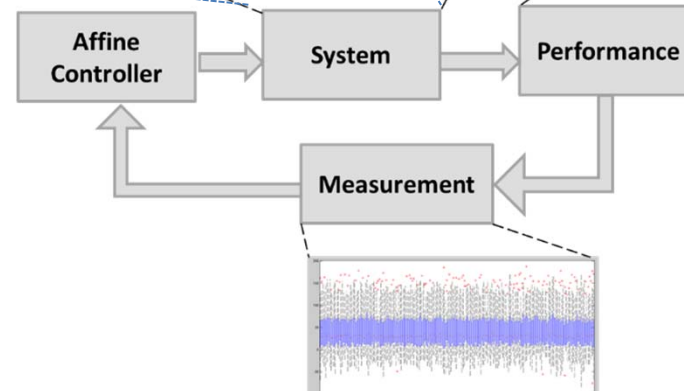
Discrete-Event Simulator



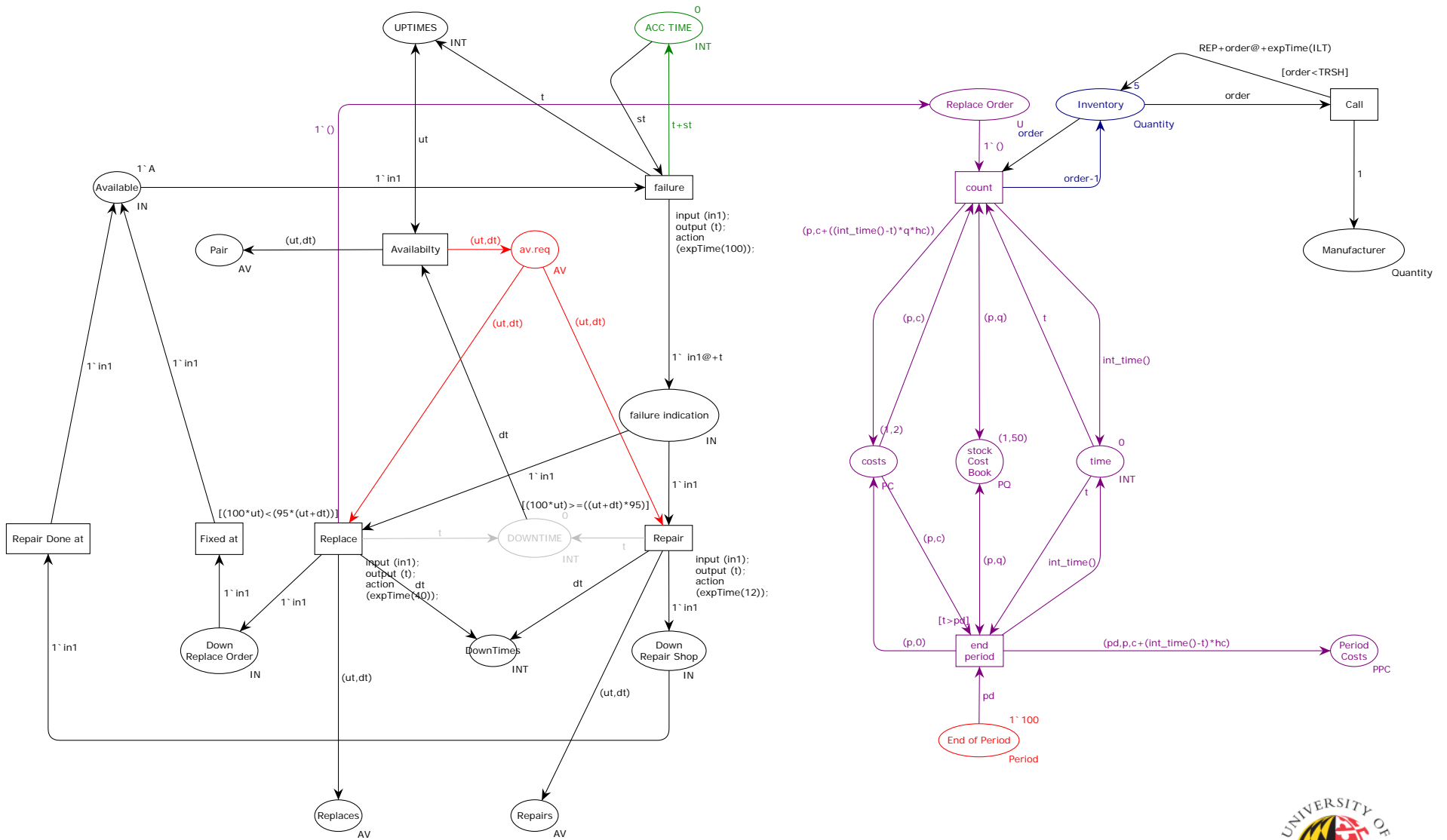
The net generates maintenance demands via sampling failure distributions for the system's parts and uses the inventory to support the system's maintenance requests. The net produces a time series (and cost) of the system's failure, maintenance, and operation.



A special type of discrete event simulator (a Petri net) is used to capture concurrency and synchronization properties of the system.



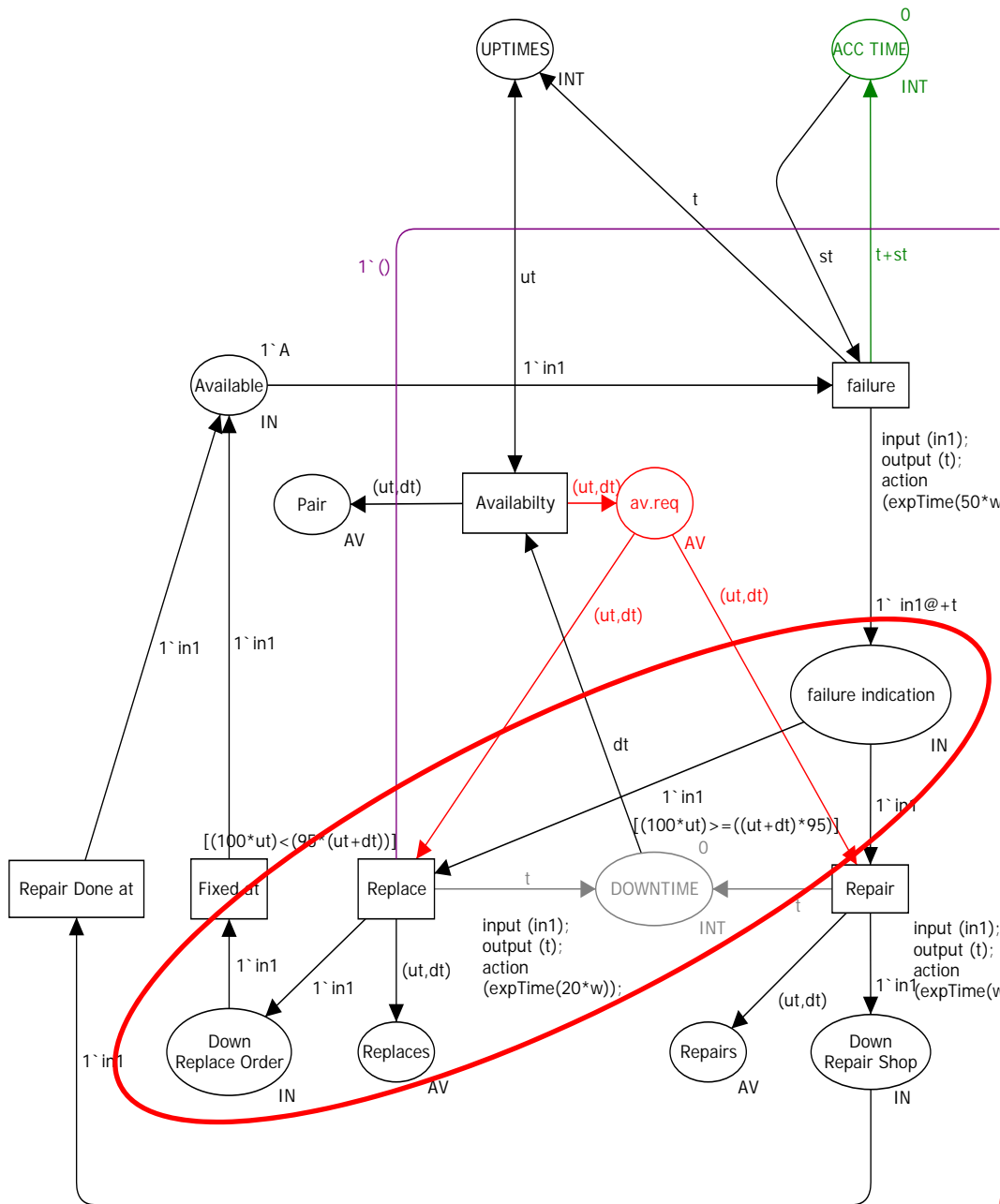
Petri Net



Replacement

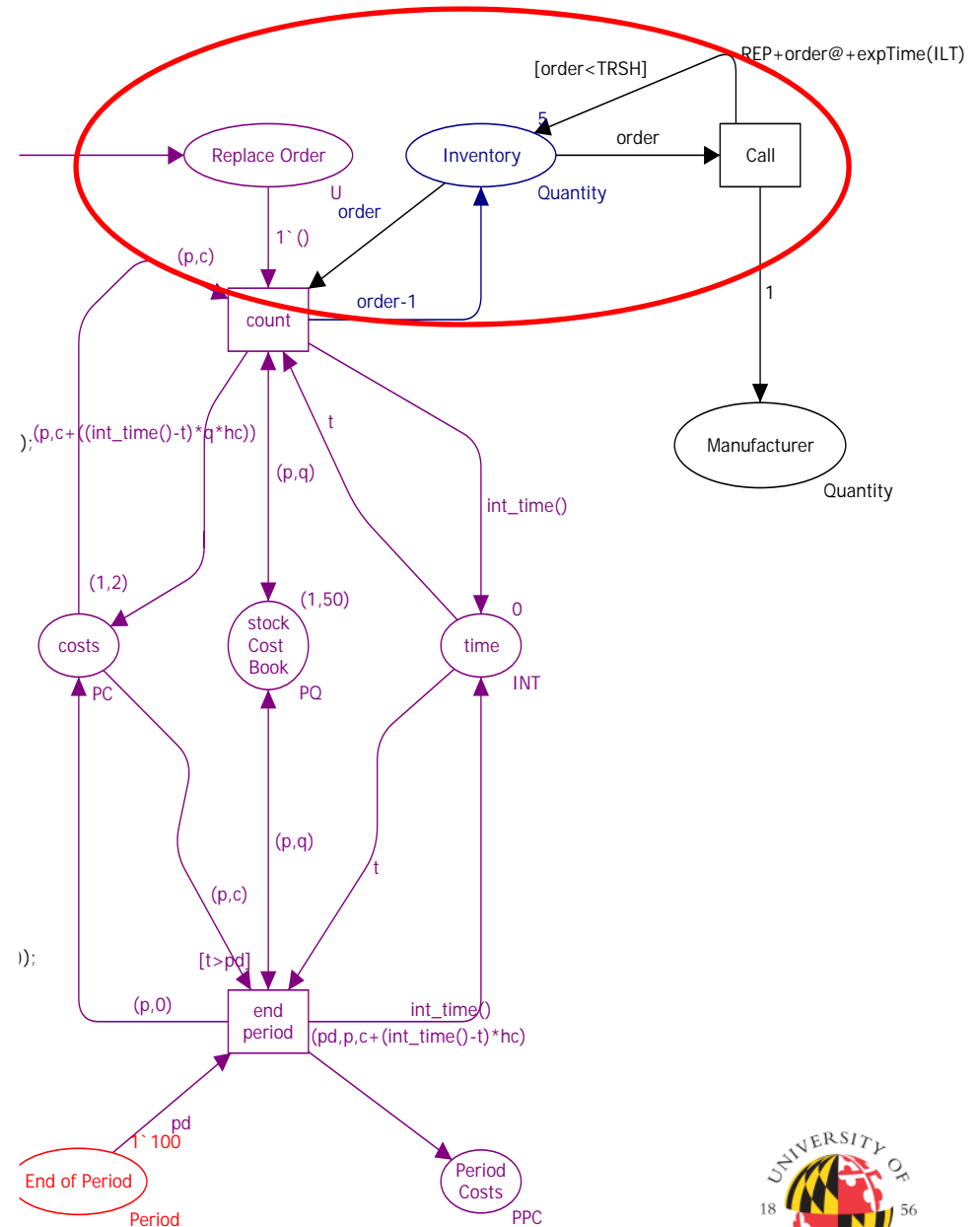
- After a failure event if the availability is below the requirement* the part is replaced
- This assumes that replacement time is always less than repair time
- After replacement, the system will be available immediately, i.e., the inventory lead time is considered in the inventory model

*The “requirement” is an operational availability constraint

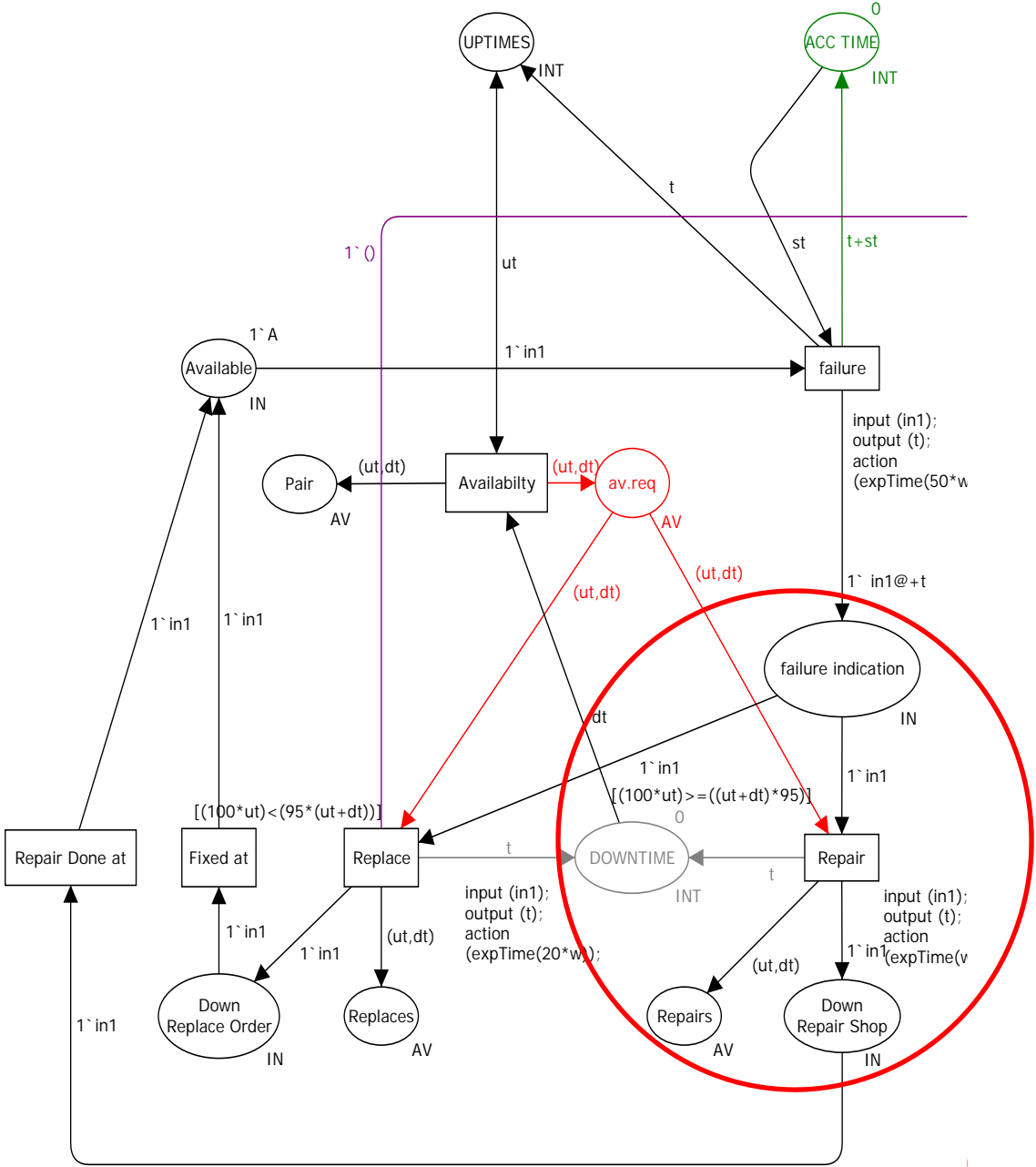


Inventory

- As an order arrives (caused by a replacement event) a new part from inventory is needed
- If the inventory level goes below a certain level, it will be replenished up to a certain level
- The inventory is connected to the manufacturer so each delivery from the manufacturer can have a different reliability



Maintenance



- After the failure event the part goes for repair if the availability exceeds the required availability (at this instant in time)
- Downtime of the system increases and a new availability is calculated after repair
- Good-as-new repair is assumed



Availability Performance and its Measurement

- Due to the accumulative nature of availability (how it is accumulated along the timeline), we need to look at the role of the time assessment window in the measurement system
- The time assessment window refers to the period of time over which the availability is **measured**, e.g., monthly, quarterly, annually, etc.
 - If the time assessment window is too long, then contractor actions near the end of the window will have little impact on the availability measurement (contractors will be inclined to “drop the ball” late in the window because nothing they do will change the result).
 - Alternatively, if window is too short, contractors are penalized for the initial condition of the system and the inventory.
 - Therefore the size of the assessment window will determine the sensitivity of contractor performance actions to different interruptions and eventually affect the contractor’s risk-taking attitude.

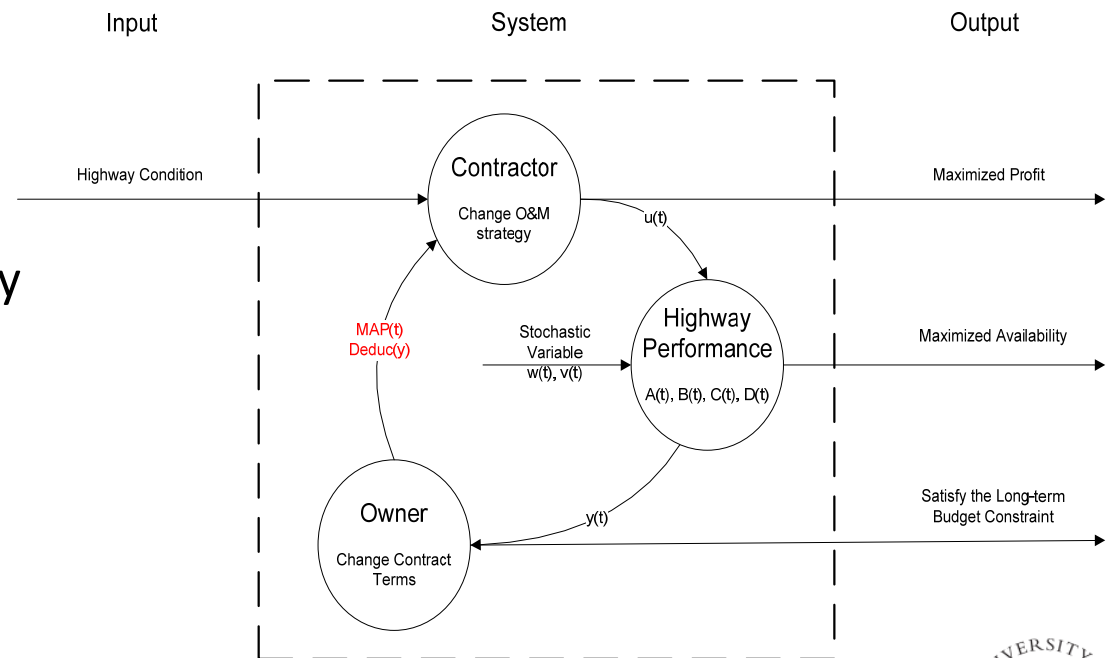


Payment Design Model

The control-feedback mechanism for availability contracts is based on an affine control model. The model aims to determine the optimal incentives/disincentives in an availability contract so that the customer can expect the best performance or availability given the long-term budget constraint while the contractor maintains a steady revenue (with profit).

Purpose of the optimization model:

- Design the contractual terms:
 - Maximum Availability Payment (MAP)
 - Deductions
- Simulate the two-layer problem (public request, private react)

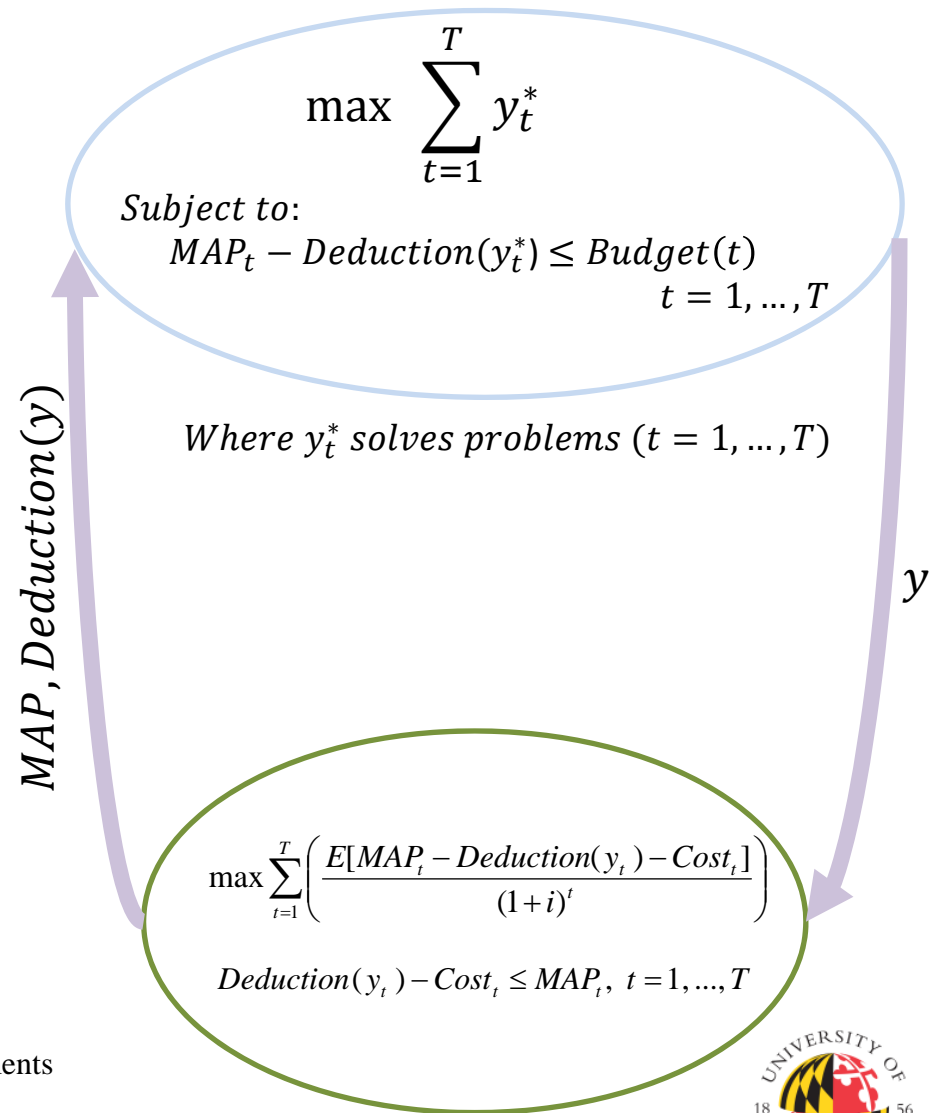


Two-Layer Optimization

- Public Sector: Design a set of Maximum Availability Payments (MAPs), and a deduction function (based on the performance) so as to:
 - Make sure the project is within budget
 - Incentivize the private sector to provide the best performance
- Private Sector: Given the MAPs and the deduction matrix, decide their strategies throughout the operation phase, such as: quality of the construction, O&M plan, and service quality, so as to:
 - Maximize their profit
 - Minimize their risk

y = availability (contractor performance)
 T = assessment time window size
 MAP = Maximum Availability Payment
 (could be affordability cap?)

i = interest rate
 $Deduction$ = penalty payments
 $Cost$ = logistics costs



Payment Model

- A model that determines what amount the contractor is paid
- The payment is based on:
 - The availability of the system, $x(t)$
 - The length of the contract, T (number of payments)
 - The penalty function, m
 - MAP (maximum availability payment)
- There is no MAP if the availability drops below some threshold

$$\max \sum_{t=1}^T \left(\frac{E[MAP_t - Deduction(y_t) - Cost_t]}{(1+i)^t} \right)$$

$$Deduction(y_t) - Cost_t \leq MAP_t, \quad t = 1, \dots, T$$

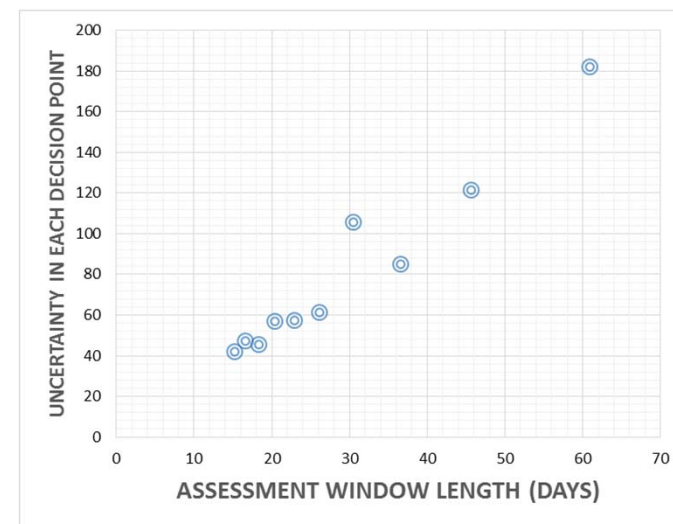
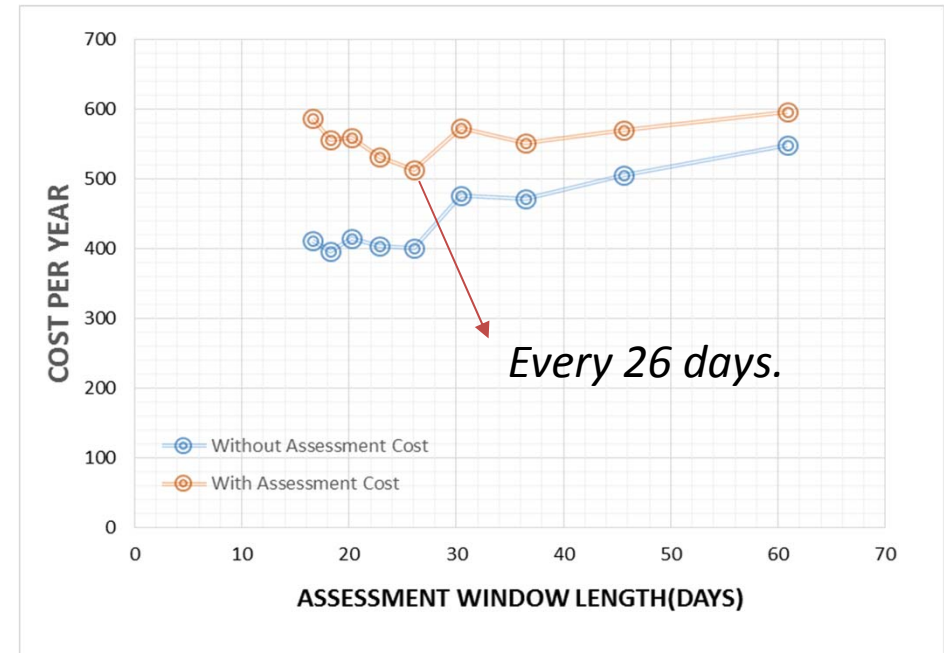
Case Study

- We tested the algorithm on an inventory model whose demand is driven by the failure of parts and subsequent maintenance requests.
- The objective of customer was to minimize the costs over fixed time window while maintaining availability. Meanwhile contractor was optimizing their revenue.
- We used Affine Controller to model contractor decision making and use convex optimization to find the global optimum
- The model will help the contractor optimize (or stabilize) the revenue while optimizing the outcome for customer
- Different inventory decision making parameters and costs are being tested to find the optimum strategy and contract
- The variance of the outcome was controlled using a Monte Carlo method



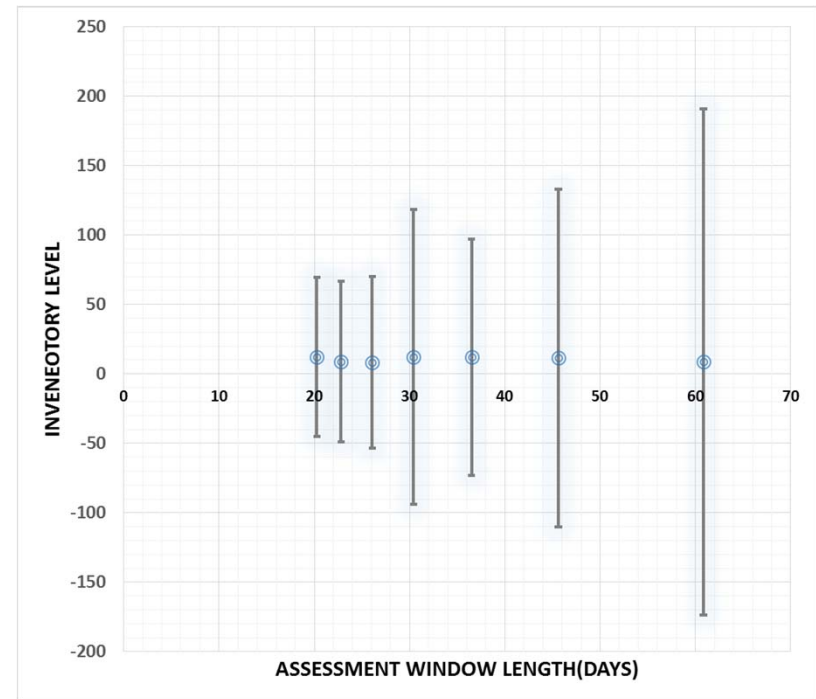
Results

- There is a assessment window that minimized the cost per period for the contractor.
- For a given limit of sampling, adding more samples helps both parties to do better cost control, however there is a limit for performance improvement.
- The amount of risk decreases when the performance is assessed more often during the contract term.
- Lower risk is highly desirable for contractors, however, a contractor might benefit from higher costs.



Unresolved Issues

- The demand/failure distribution plays a key role
 - One item or multiple items
- The integration between time-based and event-based systems is a defining part in modeling process and results
- The effect of prediction technologies (prognostics) on the contractor behavior remains unexplored
- Assessment windows results can have a feedback effect on the behavior of the contractor for the next cycle. Here we assumed the contractor works the same through the year with identical cycles in all assessment window periods.
- The effect of value at risk on contractor side remains unexplored



Conclusions

- We studied the effect of the availability assessment window on the incentives of the contractor and outcome
- We were able to demonstrate that there is an optimum size of this window
 - The number of samples we need to determine the performance of contractors has an optimum
- Different inventory decision making parameters and costs are being tested to find the optimum strategy and contract
- Contractor's decisions are also highly influenced by the amount of risk involved
 - Just focusing on expected value can be misleading especially in cases where there are only a few samples

