Towards a taxonomy of technical debt for COTS-intensive cyber physical systems

Ye Yang

Stevens Institute of Technology

CSER 2019

conference on systems engineering research

SYSTEMS ENGINEERING RESEARCH CENTER

Towards a taxonomy of technical debt for COTS-intensive cyber physical systems

Ye Yang, Jon Wade, Dinesh Verma, Turki Alelyani (Stevens Institute of Technology)
Ronald Michel (RDECOM CERDEC)
Martin Törngren (KTH Royal Institute of Technology)

17th Annual Conference on Systems Engineering Research (CSER'19)
April 3, 2019 • Washington, DC

Outline

- Background
- Research Methodology
- Software Technical Debt
- COTS Technical Debt Taxonomy
- Conclusions

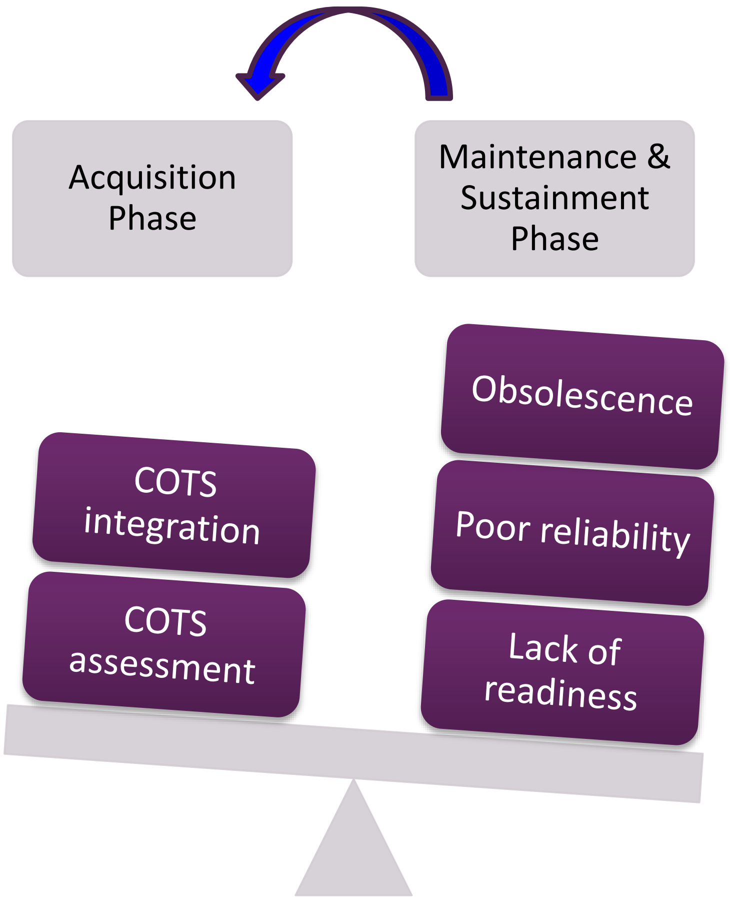
Background

- Obsolescence is a complex mix of engineering, economic, and business issues with many associated uncertainties.
- Obsolescence is the inevitable consequences of dependence on **COTS** components in many **Cyber-Physical-Systems (CPS)**
  - Long lead time of CPS, tightly-coupled components, shorter upgrade cycle of COTS, no control over COTS evolution, etc.

  - *"Future Combat System had 153 relevant systems to deal with. If every one updated once a year, that would be a change every other day!"*
    ---- Barry Boehm, USC
  - *"70 percent of electronics are obsolete prior to system fielding, and one component may become obsolete five to ten times during the weapon systems life cycle."* ---- Anthony Haynes, AMRDEC

Motivations

- **Problem Statement:**
  - — Obsolescence is the consequence of COTS technical debt that can be possibly captured and managed in early CPS life cycle activities, i.e. COTS acquisition.
    - exemplar forms for debt repayment
      - — planned systems upgrade, systems replacement costs, or in the worst case, defaulted systems



- **Motivations:**
  - — The compelling need for a systems engineering technical debt metaphor grows as well
  - — To increase awareness of COTS technical debt
  - — To support early identification, assessment, and management of COTS technical debt

Outline

- Background
- Research Methodology
- Software Technical Debt
- COTS Technical Debt Taxonomy
- Conclusions

Research Methodology

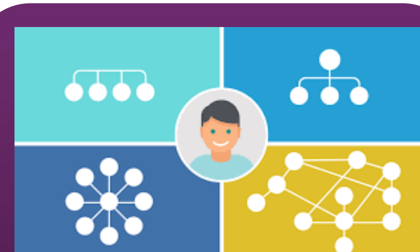
Literature Review

- Understanding trend in COTS related CPS Obsolescence studies

Afternoon talk: A Literature Review on Obsolescence Management in COTS-Centric Cyber Physical Systems

Mapping Framework

- Align existing MPTs
- Identify gap



COTS Technical Debt

- Taxonomy
- Guidelines
- Meta attributes

Outline

- Background
- Research Methodology
- Software Technical Debt
- COTS Technical Debt Taxonomy
- Conclusions

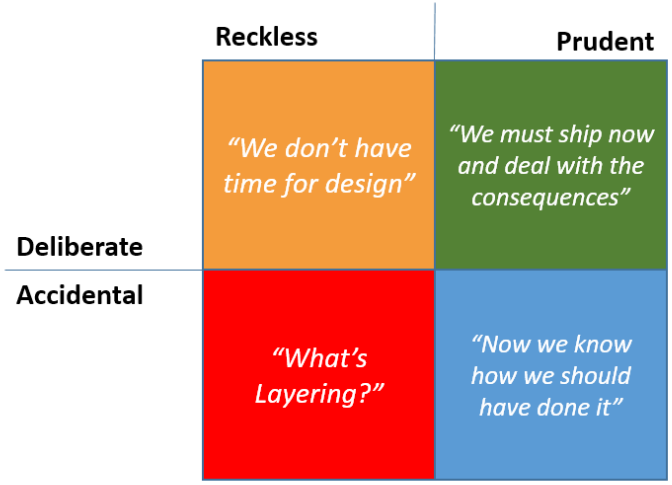
The Notion of Technical Debt

- Originated in software engineering field, coined by Ward Cunningham in 1992
  - Immature work, compromising in one dimension in order to get benefits in other dimensions
  - Initially concerning ”refactoring” at code level (i.e. implementation) in agile software development

- Evolved to span across all life cycle phases
  - a metaphor reflecting technical compromises that can yield short-term benefit but may hurt the long-term health of a software system

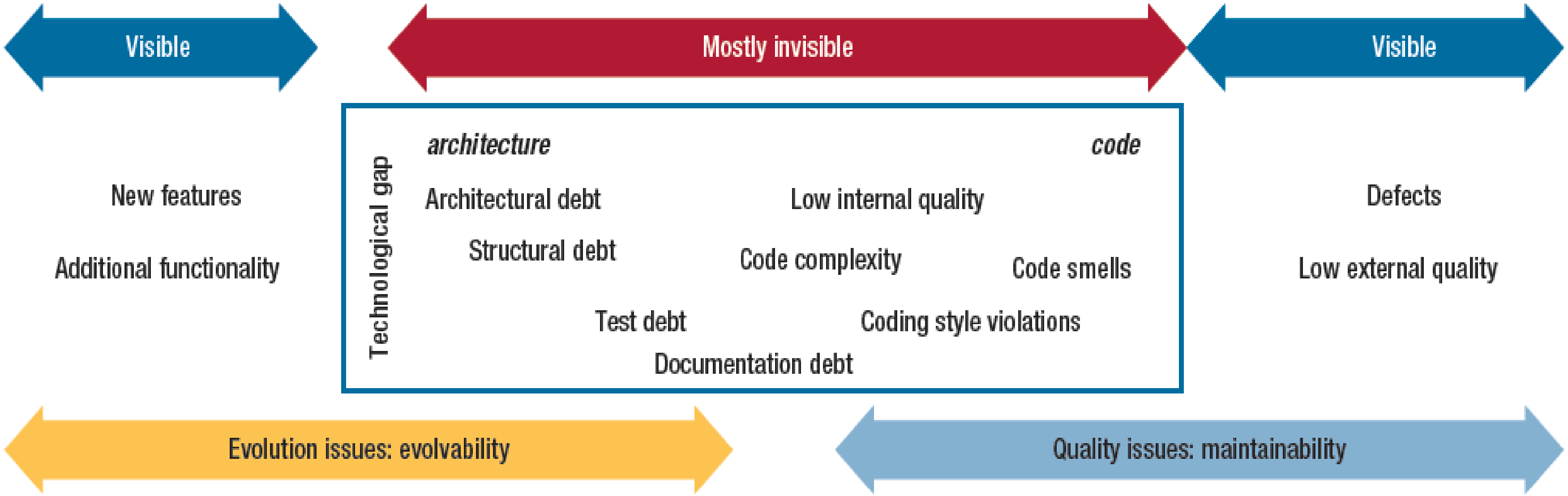
- Technical Debt Quadrants [Martin Fowler, 2009]

Technical Debt Quadrants



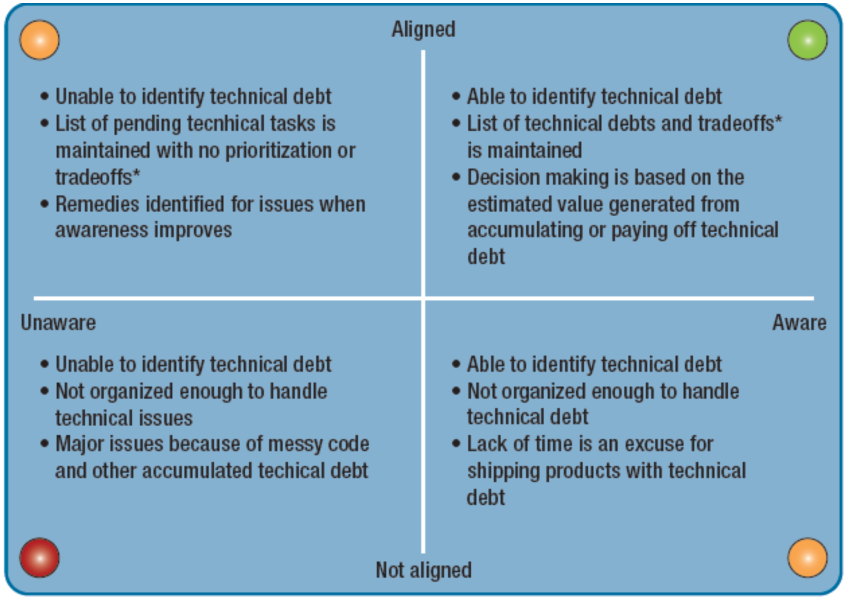
What Constitutes Technical Debt?

- Technical Debt Landscape (Ozkaya, Nord, Kruchten, 2012)
  - Differentiate visible elements from invisible elements



Existing Taxonomies on Technical Debt

- Rubin's Taxonomy
  - Context: within Agile team
    - Naïve technical debt: irresponsible behaviours or immature practices
      - sloppy design, poor engineering practices, and insufficient testing
    - Unavoidable technical debt: usually unpredictable and unpreventable
      - Design evolution, component API changes
    - Strategic technical debt: tool for organizational level trade-off:
      - e.g. quality vs. time-sensitivity
- Clark's Taxonomy
  - Context: Riot Games (*League of Legends)*
    - Local debt: standalone debt within blackbox
    - MacGyver debt: temporary, short-cut solutions, but not reliable in the long run
    - Foundational debt: future change or rework required on fundamental design assumption
    - Data debt: accumulated ripple effect of TD over time
- Bavani's Taxonomy
  - Context: distributed teams & agile testing
    - Degree of awareness of technical debt across distributed teams
    - Degree of alignment in managing technical debt across distributed teams



*Tradeoff can be an alternate approach to lessen the impact of technical issues or debt

“COTS Technical Debt” Analogy in CPS

COTS Benefits	COTS Implications	COTS “Technical Debt”
Avoids expensive development & maintenance	Up front license fees	Long term, system level: maybe more expensive to maintain
Predictable license costs & performance	Recurring maintenance fees	Yes. Incurred COTS upgrading cost and system re-evaluation/re-testing cost
Rich in functionality	Reliability often unknown/ inadequate; Unnecessary features compromise usability, security, performance	Yes. Incurred cost to take care of functional/non-functional requirement mismatch and additional verification & validation
Broadly used, mature technology	Functionality, efficiency constraints	Yes. Incurred cost to tailor to specific CPS context; increased limitation over system evolution
Frequent upgrades often anticipate organization’s needs	No control over upgrades/maintenance	Yes. Increased obsolescence risk due to life cycle mismatch between CPS system and COTS components
Dedicated support organization	Dependency on vendor	Yes. Increased obsolescence risk due to documentation and support dependency
Hardware/software independence	Integration not always trivial; incompatibilities among different COTS	Yes. Incurred cost to evaluate and enhance COTS interoperability in COTS-intensive CPS.
Tracks technology trends	Synchronizing multiple-vendor upgrades	Yes. Increased obsolescence risk due to life cycle mismatch between CPS system and COTS components

Outline

- Background
- Research Methodology
- Software Technical Debt
- COTS Technical Debt Taxonomy
- Conclusions

COTS TD Taxonomy in CPS Context

TD Category	Description	Analogy to existing work
Function	The degree of functionality mismatch between COTS capabilities and system needs.	Local TD; Data TD
Performance	The degree of mismatches between COTS capabilities and system needs, w.r.t. performance properties.	MacGyver TD; Data TD
Interoperability	The degree of interface/ assumption mismatches among various interdependent COTS components, as well as among COTS and system custom components.	MacGyver TD; Data TD
Configuration Version	CPS configuration version planning needs to address solution availability plan. Greater tendency of COTS version upgrade/refresh may lead to more obsolete COTS.	Unavoidable TD; Local TD; MacGyver TD; Foundational TD; Data TD
Documentation & Support	Lack of documentation and vendor support will seriously impact on issue resolution related to obsolete COTS.	Unavoidable; Data TD
System Evolution Limitations	Requirements imposed by COTS may place great limitation on system evolution.	Unavoidable TD; Foundational TD; Data TD
Organic	People-centric perspective of TD focusing on organizational decision-making, behaviours, and practices associated with those personnel responsible for introductions of new technologies & systems and/or the sustainment of existing systems	Local TD; Naïve TD; Strategic TD

COTS TD Management Activities

- TD identification
- TD representation
- TD communication

- TD measurement
- TD prioritization
- TD Monitoring

- TD repayment
- TD prevention

Guidelines for applying the taxonomy - 1

- TD Identification:
  - —Detects TD caused by intentional or unintentional COTS decisions
    - It is a many-to-many relationship between a COTS component and a COTS TD item;
    - It is possible for a COTS TD item to be associated with multiple categories, since intensive COTS TD items in CPS systems may come from the complex interdependencies among COTS hardware and software components;
    - The identification of System Evolution Limitations TD items is the most difficult, and it is essential for offsetting COTS obsolescence risk through early involvement of user/customer/operating organizations in COTS assessment and acquisition activities;
    - It is suggested to label all applicable COTS TD categories according to its relevance and significance.
  - —Example techniques:
    - COTS assessment; modeling / simulation; prototyping; dependency analysis; checklist

Guidelines for applying the taxonomy - 2

- TD Measurement:
  - —Quantifies the benefit and cost of known COTS TD in a system through estimation techniques
    - Measure a COTS TD item whenever it is identified;
    - Function, Performance, and Interoperability TD items need to be measured based on intensive COTS assessment results;
    - Re-measure after TD repayment activities.
  - —Example techniques:
    - Timed value; NPV; Real Option, etc.

Guidelines for applying the taxonomy - 3

- TD Repayment:
  - Resolves or mitigates COTS TD
    - Establish COTS TD repayment strategies, with respect to particular COTS TD types.
      - Strategies for resolving COTS mismatches
        - Bridge
        - Wrapper
        - Mediator
        - Negotiation
      - Strategies for mitigating configuration version TD might include the following options:
        - Skipping the new COTS version;
        - Upgrade to keep up with every new COTS version;
        - Upgrade COTS every other version;
        - Upgrade on a regular basis, e.g., every 18-month.
  - Example techniques:
    - COTS version upgrade; reengineering; refactoring; incident fixing; fault tolerant; repackaging; automation

Template for representing a TD item

Attribute	Description
ID	A unique identifier for the COTS TD item.
Name	The name of a specific COTS TD item
Location	The location of the identified COTS TD item, e.g. the name of the COTS(s) with which it is associated.
Accountable Party	The party responsible to repay the COTS TD item, e.g. COTS vendor, integration team, program office, specific organization. This identifies the "accountable" debt-holder for the liability. The Accountable Party is identified at the start of a new design/development/ modernization effort, and can assign TD "tracking" and "maintenance of TD visibility" within its span of authority/control.
Type	The COTS TD type that the COTS TD item is classified into.
Description	General information on the COTS TD item.
Open date/time	The specific date/time when the COTS TD is identified.
Principle	The estimated cost of repaying the COTS TD item.
Interest amount	The estimated extra cost of tolerating the COTS TD item.
Interest probability	The probability that the interest for the COTS TD item needs to be repaid.
Contagion	The degree of spreading of the COTS TD item through the interfaces with other system components, if this TD is allowed to continue to exist.
Context	A certain implementation context of a specific COTS TD item
Propagation rule	How the COTS TD item impacts the related parts of the CPS system
Intentionality	Is the COTS TD item Intentionally or unintentionally incurred?

Outline

- Background
- Research Methodology
- Software Technical Debt
- COTS Technical Debt Taxonomy
- Conclusions

Conclusions and Future Directions

- Conclusions
  - — Compelling and critical need for a Systems Engineering technical debt metaphor grows
  - — The notions of COTS technical debts will help to inform COTS decision making practices in the acquisition process to avoid unaffordable obsolescence issues particularly in the sustainment phase
  - — Taxonomy of COTS-related technical debt can support early identification, communication, and assessment of obsolescence risks in CPS system engineering life cycles

- Future directions:
  - — Map major obsolescence issues in existing case studies to the proposed COTS TD taxonomy
  - — Modelling and Simulation of COTS changes and impact on technical debt aggregation within CPS
  - — Align COTS TD management techniques and align with existing acquisition activities

Thank you!
&
Questions?

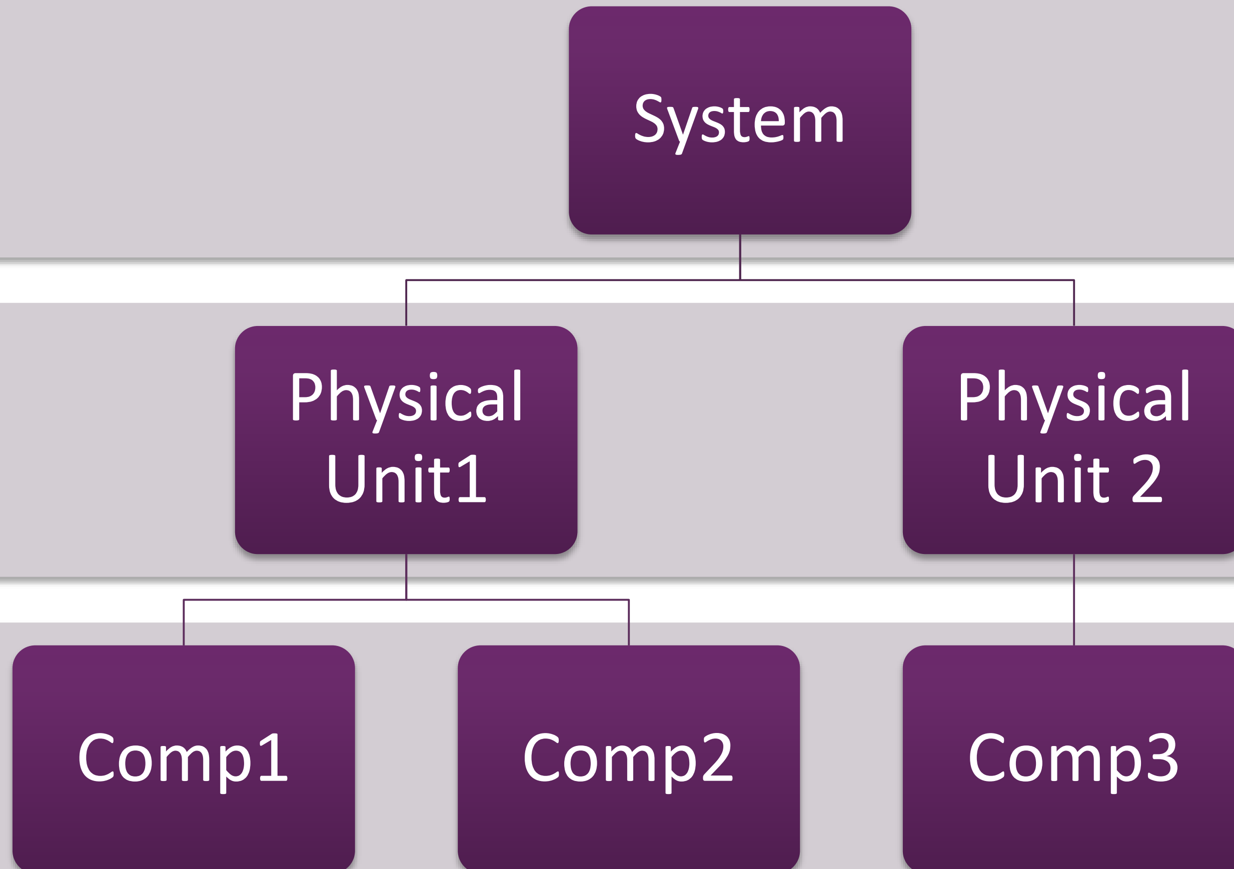
Backup Slides

Hierarchical View of a Simple Technical Debt Model for COTS-Intensive CPS

Technical Debt = f_S(changes across entire system, required work, TD management strategy)

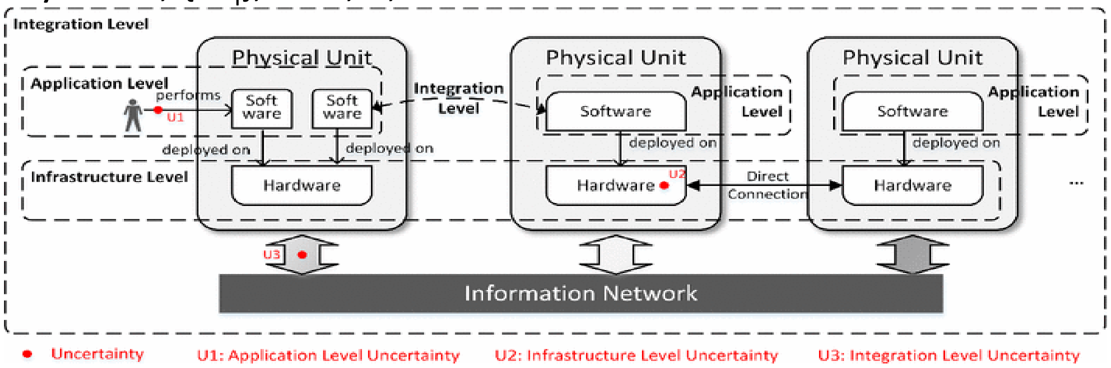
Technical Debt = f_{PU}(changes across a PU, required work, TD management strategy)

Technical Debt = f_C(changes within a component, required work, TD management strategy)



Modeling COTS-intensive CPS

- COTS-intensive CPS
  - A set of physical units, i.e. subsystems, $\{SS_i\}$, i=1, 2, …. M
  - Attributes:
    - Budget, schedule
    - %reqt's covered by COTS
    - Planned upgrade cycle
    - Acquisition cost
    - COTS technical debt



- Dependency matrix
  - Interface requirements among all components

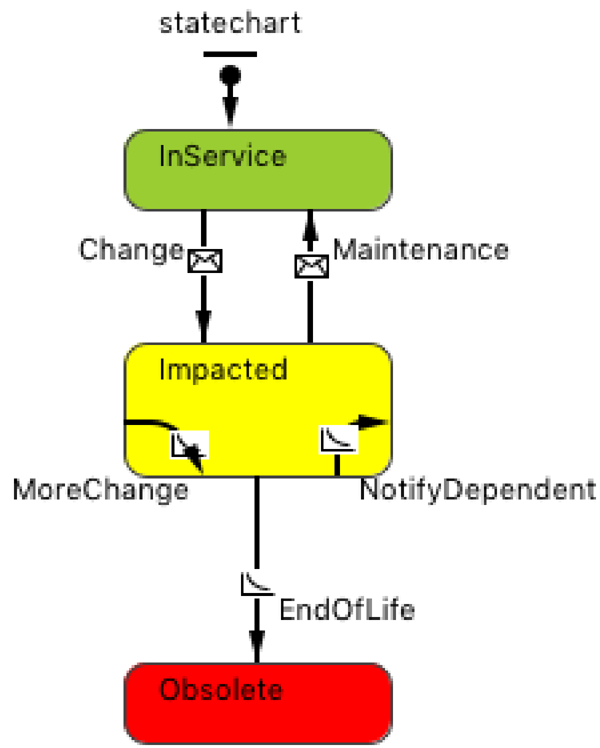
- Multi-Agent Models
  - Each physical unit, SS_i
    - A set of hardware and/or software components, $\{C_{ij}\}$, j=1, 2, …. n_i
    - Type: Application, Infrastructure, Network, other
  - Each component, C_{ij}
    - Attributes: %reqt's gap; acquisition cost, upgrade cycle, upgrading cost
    - Type: COTS h/w, COTS s/w, custom h/w, custom s/w, other

Modeling COTS Configuration Version Technical Debt

- Discrete Event Model
  - COTS change events
    - COTS change:
      - Upgrade cycle: Probabilistic distribution function: e.g. [6month, 12month]
      - Change ratio: random variable {0, 1), larger number indicating greater portion of COTS is changed
  - TD management actions
    - TD Principal Measurement
      - Component level: f_C(change ratio, required work, TD reduction strategy)
      - Physical Unit level: f_{PU}(changes across a PU, required work, TD reduction strategy)
      - System level: f_S(changes across entire system, required work, TD reduction strategy)
    - TD Reduction strategies
      - 0: no work
      - 1: upgrade every version
      - 2: upgrade every other version
      - 3: upgrade until end-of-life
    - TD Dynamic Forecasting
      - f(TD principal, probablity of TD interest, TD interest amount, t)

COTS Change Propagation and Change Impact Modeling

- COTS Change Impact Analysis
  - Dependency matrix
    - Coupling rate
  - State transition model
    - InService
    - Impacted
    - Obsolete



Examples of Decision Scenario Simulation

- Selecting different COTS-based solutions
- Dynamics of TD aggregation and reduction