systems

MDPI

Article

Constructing Models for Systems Resilience: Challenges, Concepts, and Formal Methods

Azad M. Madni *, Dan Erwin and Michael Sievers

Department of Astronautical Engineering, Viterbi School of Engineering, University of Southern California, Los Angeles, CA 90089-1192, USA; erwin@usc.edu (D.E.); michael.sievers@usc.edu (M.S.)

* Correspondence: azad.madni@usc.edu; Tel.: +01-213-821-1001

Received: 22 October 2019; Accepted: 20 January 2020; Published: 24 January 2020

Abstract: As systems continue to grow in scale and complexity and have to operate safely in challenging disruptive environments, system safety and resilience has become a critical requirement. This recognition has drawn attention to the concept of resilience, which has different definitions and several different interpretations that tend to be domain specific. For example, resilience in health care clinics means something quite different than resilience in self-driving cars, or energy grids. This paper reviews the different characterizations of resilience and assesses their value proposition in realizing engineered resilient systems. This paper emphasizes the importance of systems modeling in engineering resilient systems and presents an overarching methodology that employs different modeling approaches for operational tasks as a function of problem context. This paper specifically focuses on systems modeling in partially observable and potentially hostile environments. It discusses the need for system model verification, which is key to safety, and system flexibility and adaptability, which are key to resilience. It introduces a formal, probabilistic modeling construct called the "resilience contract." This construct employs a state-based representation that formalizes the concept of resilience while enabling system model verification and affording requisite flexibility for adaptation and learning. The key findings of our research are that different system modeling approaches and algorithms are needed based on mission tasks and operational context; adaptive capacity and continual adaptability are the two promising characterizations of resilience that can be cost-effectively realized in real-world systems; and the resilience contract construct is an effective means for probabilistic verification of system model correctness while affording flexibility needed for adaptation and learning. Collectively, these findings contribute to the body of knowledge in both model-based systems engineering (MBSE) and engineered resilient systems.

Keywords: engineered resilience; resilience modeling; resilience definitions

1. Introduction

As systems continue to grow in scale and operational missions become increasingly more complex and disruption prone, system resilience has become essential for assuring safe and successful mission accomplishment in the face of systemic faults and unexpected environmental conditions. System resilience has been characterized variously as the ability to rebound, resist/absorb, dynamically adapt capacity, and continually adapt in the face of ongoing change [1–3]. These different interpretations have inadvertently become a source of confusion to the systems engineering community. Compounding the problem is the fact that system resilience tends to be achieved through a combination of domain independent methods (e.g., physical redundancy), and domain-specific techniques (e.g., functional redundancy; adapting to a patient surge in a health care clinic, ensuring perimeter security of a military asset at an outpost). While the definitions of resilience abound in the literature, some are more useful than others. Some of the more useful definitions are:

- resilience is the capability of a system to maintain its function and structure in the face of internal and external change and to degrade gracefully when it must [4];
- resilience is "the ability of a system to withstand a major disruption within acceptable degradation parameters and to recover within a suitable time and reasonable costs and risks [5];
- given the occurrence of a particular disruptive event (or a set of events), the resilience of a system to that event (or events) is that system's ability to efficiently reduce both the magnitude and duration of deviation from targeted system performance levels [6];
- resilience is the ability of a system to anticipate and adapt to potential surprise as well as failure —this definition is rooted in the notion that when prevention is impossible, the system's ability to cope with disruptions becomes paramount [7];
- a resilient system can adapt to internal and external errors by changing its mode of operation without losing it ability to function [8].

Collectively, these definitions inform the kinds of system modeling approaches that are needed. They also help with determining which of the common resilience viewpoints offer productive lines of inquiry in the sense that they can be realized in engineered systems in the real world.

Over the last decade, several innovative concepts have appeared in the literature to realize resilience in systems [1,8–15] these include complexity management; context-sensitive topology selection; on-demand addition of various resource types; reversion to a safe mode, when failing or compromised; preemption of cascading failures through proper design of links between nodes to minimize likelihood of failure propagation; provision of buffers (i.e., safety margins) and buffer management to absorb impact of disruptions; functional redundancy, whereby a function can be performed using different means; and dependable means for evaluating trust and reputation especially in system-of-systems.

The common resilience concepts [4] include (1) *identification of critical functions* or services and protection of not only critical functions but also those functions that could potentially interfere with or break those critical functions (a simple example might be a sneak circuit in which a benign function suffers a fault which propagates to a critical function through a circuit path caused by the fault); (2) *redefining decision thresholds* based on context; (3) *maximizing achievable reorganization* given memory constraints; and (4) *incorporating safety margins through* pre-engineered or adaptive capacities, or facilities for continual adaptation in the face of ongoing changes.

Understanding critical functionality is the key to pre-planning responses to known disruptions. Decision thresholds come into play in determining whether or not a system is able to absorb a shock without having to restructure or reconfigure. It is important to note that recovery time is an important metric in assessing system resilience post-disruption when a decision threshold is not exceeded. Memory constraint determines the degree to which a system can reorganize in response to a disruption. Adaptive capacity is key to dealing with both surprise surges and downturns. It is also the key to exploiting resilience opportunities and conforming to limits in a safe-to-fail manner. Another relevant metric is coverage, which includes detection and recovery, and whether or not it occurs before sustaining permanent damage, or mission loss. The time dimension is integral to all conceptualizations of resilience [1,5].

This paper reviews various characterizations of resilience and their value proposition in formalizing the concept of resilience and demonstrating its use on a representative real-world problem. This paper is organized as follows. Section 2 reviews the many facets of resilience and the challenges they pose, along with specific criteria to evaluate their potential usefulness for a particular problem domain. Section 3 presents exemplar resilience strategies for different domains and appropriate resilience strategies to cope with them. Section 4 presents the overall methodology. It begins by presenting probabilistic and adaptive modeling using Partially Observable Markov Decision Process (POMDP) and introduces "resilience contract," an innovative modeling construct which combines key concepts from traditional contracts with POMDP and heuristics to enable system (model) verification while enabling adaptability through its state-based representation. Section 5 summarizes the main findings from this work and its key contributions to both model-based systems engineering and resilience engineering.

2. The Different Characterizations of System Resilience

Resilience is a complex concept with multiple characterizations and definitions [1,2]. Therefore, it is important to evaluate these different characterizations in terms of their potential to provide useful lines of inquiry (i.e., those that can lead to successful realization of resilient systems).

At the outset, we note that when developing resilience strategies and mechanisms, there is the likelihood of unintended consequences that can lead to dramatic sudden failures without warning [1,5]. These undesirable outcomes are invariably the result of *unexpected side effects* [1] arising from unknown interactions; *undesirable change cascades* [1,4] arising from interdependencies and interactions among subsystems; *unexpected conflicts* [4] that can arise among resilience mechanisms that are introduced in the system at different levels; and *flawed assessments* of what is wrong [4]. The problem is further exacerbated by the fact that researchers from different disciplines and communities do not share a common vocabulary about resilience.

Against the foregoing backdrop, Table 1 presents domain-independent characterizations of resilience which were derived from a review of the resilience literature [1,2,5].

Table 1. System Resilience Characterizations.

- Resilience = restoration of pre-disruption state ("capacity to rebound").
- Resilience = withstand disruption within performance envelope ("capacity to resist").
- Resilience = extend resources to fit a surge-type disruption ("capacity to adapt").
- Resilience = continue to meet performance requirements in the face of ongoing changes ("capacity to continually adapt").

The first characterization, which is also the most intuitive (i.e., "capacity to rebound"), implies that a system can fully or partially restore its previous state (i.e., operational performance and trajectory) following a disruption. Reduced to its basic implication, it is akin to saying, "if it is broken, fix it," with no guidance or clue on how to do that. In other words, there is no "how to" implied in this characterization.

The second characterization unfortunately makes a false equivalence between resilience and system robustness and consequently has led to some confusion in the engineering community. Robustness is a well-defined concept in dynamics and control. It implies that a system can absorb and survive disruption without undergoing any structural change or reorganization. It does not address system performance outside the system's designed performance envelope. For example, Avizienis et al. [16] defined robustness as "dependability with respect to external faults." Resilience, on the other hand, addresses system adaptability which invariably requires some form of structural change. Unlike robustness, resilience also implies graceful degradation when the disruption is outside the system's designed performance envelope.

The third characterization is that of "adaptive capacity." Woods has defined a special case of this concept as "graceful extensibility" in his recent publication [2]. This characterization implies that a system can sustain performance and/or bring to bear dynamic change in available resources or capability when disruption magnitude and duration challenge designed performance boundaries.

The fourth characterization of resilience equates resilience with continual (i.e., sustained) adaptability. This characterization implies that a system can continue to function by adapting as needed without getting trapped in "unsafe" states as conditions evolve over time. In practice, it is difficult to build a system that adapts continually in a variety of ways to circumvent, survive or recover from disruptions.

Based on the resilience literature, the right resilience characterization for a particular problem domain depends on how well that characterization can advance inquiry leading to realizable resilience in that domain, and whether that characterization can enable quantification of resilience. The application of these criteria to the different characterizations in Table 1 provides insights into which definitions are worth pursuing.

Characterizing resilience as the *capacity to rebound* (Figure 1) has thus far not been particularly useful in that it offers little insight into possible implementation. In other words, this interpretation

does not shed light on possible resilience mechanisms that can be realized in engineered systems. As a result, this characterization has not yielded fundamental findings, foundational theories, or new engineering techniques [1,2]. However, being one of the first and most obvious characterizations of resilience, it did manage to stimulate interest in resilience research. Therefore, its value is primarily historical.

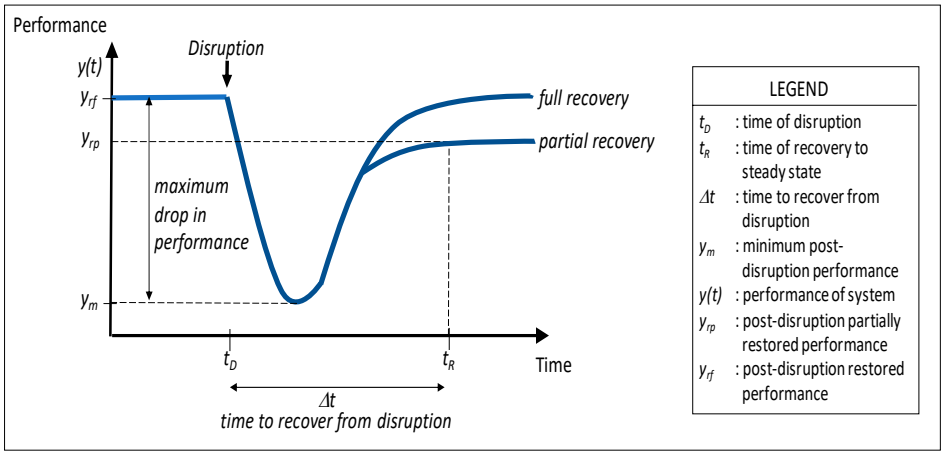

Figure 1. General form of resilience curve for resilience defined as rebound.

Characterizing resilience as *robustness* creates confusion and is in fact erroneous. In dynamics and control, robustness is defined as the ability of a system to withstand internal/external threats or changes without requiring structural modification or reconfiguration and without experiencing degradation in system performance [9]. In other words, robustness is focused on preventing a system from degrading and maintaining functionality within acceptable levels post-disruption. Resilience on the other hand is the ability of a system to adapt to change or disruptions by multiple means including structural modification and dynamic reconfiguration. Creating a false equivalence between resilience and robustness is both incorrect and unhelpful. Specifically, it does not help answer how systems adapt in the presence of unmodeled or poorly modeled disruptions. The point here is that there are different degrees to which systems can be protected from disruptions. Robustness is a valid approach but is limited in what it can do.

Characterizing resilience as *adaptive capacity* is prescriptive and therefore useful. It is concerned with a system's ability to dynamically extend capacity on demand to counter disruptions. It implies elasticity in system behavior (i.e., the system has the ability to "stretch" or "shrink" to handle a surge or drop in demand for a capability or a resource). It specifically addresses systems with finite resources in dynamic environments having to extend capacity to respond to disruptions that challenge performance boundaries (i.e., envelope). Without this ability to extend, a system becomes brittle, leading to sudden collapse in performance. Of course, the magnitude and duration of the surge determines the most appropriate resilience strategy. Imagine a health care clinic or trauma center which experiences a sudden surge in patients [17,18]. These clinics need to decide how to respond in a way that exhibits resilience. This response, in turn, requires an analysis of the disruption. In this example, the clinic needs to assess the magnitude and expected duration of the surge. If the surge is determined to be a temporary spike, then the clinic can get by with paying overtime to existing employees. If the surge is likely to be for a longer period, it pays to add temporary personnel and possibly more beds for the duration of the surge. If the surge is determined to be a trend (i.e., expected to last for an indefinite period), the clinic needs to hire permanent personnel, invest in more beds, and possibly add more space. In this regard, graceful extensibility, a term coined by Woods [2] is applicable. Graceful extensibility is a form of adaptive capacity. A more general concept than "graceful degradation," it means being prepared to adapt to handle disruptions that are at or beyond the designed performance boundary.

Characterizing resilience as *continual or sustained adaptability* builds on the recognition that adaptability is a central concept in resilience. However, a resilient system may need to continually adapt, hence the concept of continual or sustained adaptability. This characterization is not meant to imply that the adaptation is always going to be in the same dimension. Adaptation can vary (e.g., draw on reserve physical resources, replenish fuel or parts inventory, optimize coverage of an area with scarce resources by repositioning the limited available resources). Layered networks and complex adaptive systems typically exhibit this property. However, most systems are not equipped to exhibit continuous adaptation in multiple dimensions. In fact, most systems fail to adapt completely or in time resulting in sudden collapse when confronted with a new change event, or a new requirement. The key issues that arise when engineering for sustained adaptability pertain to: governance constraints and architectural properties that facilitate sustained adaptation; design principles and techniques that enable engineering systems and system-of-systems (SoS) for sustained adaptability; and ensuring that the engineered system/SoS has the ability to continually adapt as needed over time and in response to change precipitating events.

3. Exemplar Domain-Specific Resilience Strategies

The discussion in Section 2 set the stage for discussing resilience strategies for different domains. Table 2 presents examples of resilience strategies for four different domains. For each exemplar domain, resilience strategies are presented for specific disruptions.

Table 2. Exemplar Resilience Strategies for Different Domains.

- Health Care Clinic
  - Disruption: patient surge
    - Modifiers: magnitude and duration of surge
  - Resilience strategies: function of modifiers
    - Short-duration surge ("spike"): overtime pay for personnel;
    - Extended surge ("pulse"): add temporary personnel, rent additional beds;
    - Long-duration surge ("trend"): hire new staff, invest in infrastructure.
- Multi-Unmanned Aerial Vehicle (UAV) Operations
  - Disruption: drop in perimeter coverage due to loss of a UAV resource
    - Modifiers: degree to which coverage is lost
  - Resilience Strategies: function of reduction coverage (measured by value of fitness function)
    - Launch backup UAV to replace grounded UAV ("adaptive capacity");
    - Restore coverage by optimizing location and attitude of remaining good UAVs ("continual adaptability").
- Self-Driving Vehicles
  - Disruption: notification of accident ahead on freeway
    - Modifiers: distance to accident ahead, severity (time to clear freeway)
  - Resilience Strategies (adaptive behaviors"):
    - Get off freeway and take surface streets (if long clear-up time);
    - Stay on freeway in rightmost lane to exit fast if needed (fender-bender);
    - Hand-off task assignment to vehicle beyond the accident (if major pileup).
- Spacecraft Swarm
  - Disruption: untrustworthy spacecraft (Byzantine fault)
    - Modifiers: density of inconsistent observations sent to swarm (impacts SA)
  - Resilience Strategies:
    - Guaranteed message delivery protocols ("preemptive strategy");
    - Unique spacecraft identifiers ("pre-emptive strategy");
    - Formal trust and reputation evaluation resulting in ignoring faulty spacecraft ("adaptive behavior");
    - Formal mechanism for rehabilitating a spacecraft if fault is cleared ("adaptive behavior").

The first problem domain is a *health care clinic*. A particular disruption that a health care clinic is susceptible to is a patient surge. The resilience strategy for patient surge depends largely on the duration and magnitude of the surge. For a constant magnitude surge, the response strategy is a function of surge duration as shown in Table 2.

The second problem domain is *multi-unmanned aerial vehicle* (*UAV) operations*. The specific mission is maintaining perimeter security of a parked transport aircraft. Typical disruptions are loss of a UAV due to low battery power or system malfunction. The key modifiers are degree to which coverage is lost as a result of loss of a UAV. The resilience strategy is intended to restore coverage in the shortest time. Coverage is quantified by a fitness function (i.e., higher the fitness function, better the coverage).

The third problem domain is *self-driving vehicles*. Disruptions can take a variety of forms. Let us say that the disruption is an accident ahead on the freeway and the autonomous vehicle is notified about the accident. The modifiers in this case is the distance to the accident location and severity of the accident that translates into time to clear up the accident. The resilience strategies are a function of the time to clear up the accident and the magnitude of the accident (i.e., fender-bender to major pileup). The resilience strategies for this case are: exit the freeway and take surface streets (if time to clear freeway is significant; most navigation systems provide this recommendation); stay on the freeway in the rightmost lane to exit fast if needed (if a fender-bender); hand off mission/task assignment to another vehicle beyond the location of the accident (if a major pileup).

The fourth problem domain is *spacecraft swarm*. The disruption is the existence of an untrustworthy spacecraft in the swarm (i.e., a Byzantine fault). The key modifier is the density of inconsistent observations sent to the swarm. These observations impact situation awareness within the swarm. The resilience strategies in this case are: guaranteed message delivery protocols ("preemptive strategy"); unique spacecraft identifiers ("preemptive spacecraft strategy"); formal trust and reputation evaluation resulting in ignoring faulty spacecraft ("adaptive behavior"); formal mechanism to rehabilitate spacecraft if fault is cleared ("adaptive behavior").

4. Methodology

Our overall methodology employs a structured prescriptive framework for choosing the right modeling approach for different problem contexts (Tables 3,4 and 5). Problem context is defined by the *mission objective*, *a priori knowledge of system states*, *state of the environment* (e.g., observability, existence of threats), *status of own assets* (e.g., number of UAVs available to pursue mission, their battery state, etc.), and *tasks* that need to be performed within the overall planning and decision-making rubric to achieve mission objectives. Examples of such tasks are navigation/route planning, perimeter surveillance around a high value asset, and responding to environmental disruptions and enemy actions.

The *key heuristic* used in model selection is "*choose the simplest modeling construct/approach that achieves objectives with minimal computational load*." Achieving objectives includes accounting for the state/status of all relevant variables (i.e., those that bear on decision making) and system and environment constraints. The model should also allow the system to respond to systemic malfunctions and external disruptions (including those caused by an intelligent adversary). Table 3 presents how the modeling approach varies with operational context for planning and decision-making tasks.

Table 3. Modeling Approach Selection for Planning and Decision Making as a Function of Context.

Context	Modeling Approach/Algorithm
Full observability, perfect sensors	Rule-based Logic, Decision Tree, Finite State Machine (FSM)
Full observability, noisy sensors	Markov Decision Process (MDP)
Partial observability	Partially Observable Markov Decision Process (POMDP)
Partial observability with intelligent adversaries (i.e., threats)	Re-initialized POMDPs (where re-initialization occurs with appearance of new threats)

In the same vein, Table 4 illuminates how the modeling approach changes as a function of context for waypoint navigation tasks.

Table 4. Modeling Approach Selection for Waypoint Navigation as a Function of Context.

Context	Modeling Approach/Algorithm
Full observability, perfect sensors	Finite State Machine (FSM)
Full observability, noisy sensors	Markov Decision Process (MDP)
Partial observability	Partially Observable Markov Decision Process
Partial observability with pop-up threats	Re-initialized POMDPs

It is equally important to recognize that not all tasks require a state-based approach. Table 5 presents a modeling approach for surveillance tasks that require an entirely different modeling approach (e.g., adaptive optimization using fitness function).

Table 5. Modeling Approach Selection for Surveillance as a Function of Context.

Context	Modeling Approach/Algorithm
Full observability, no resource constraint	Perimeter Coverage Optimization
Full observability, resource constrained	Adaptive Optimization of Fitness Functions

The foregoing three tables convey three key points. First, for certain tasks, as the context gets more complex, the state machine representation also becomes more complex. This is shown through the progression from Finite State Machine (FSM) to Markov Decision Process (MDP) to Partially Observable Markov Decision Process (POMDP) for planning and decision-making and waypoint navigation tasks. Second, state machine representations can be employed at different levels of abstraction (e.g., planning and decision-making and waypoint navigation are two different levels). Third, not every task requires a state machine representation. For example, a surveillance task requires adaptive optimization of the fitness function.

The following subsections elaborate on specific modeling approaches that we have employed in our ongoing research.

4.1. Probabilistic and Adaptive Modeling Using POMDP

Today, ad-hoc safety-net functions are commonly used to increase system resilience. Safety-net functions comprise transitioning a malfunctioning system to safe and sustainable operation thereby enabling time for human intervention. Researchers have explored a variety of promising, formally checkable representations that show promise in realizing more rigorously defined resilience [19–24]. In this spirit, we are pursuing a variant of Contract-Based Design (CBD) [25–28], because it is intuitive, rigorous, and extensible in dealing with unknown-unknowns. CBD is a means for defining system requirements, constraints, behaviors, and interfaces by a pair of assertions, $C = (A, G)$, in which A is an assumption made on the environment and G is the guarantee a system makes if the assumption is met. Assumptions are system invariants and preconditions while guarantees are system post-conditions. More precisely, invariant contracts describe a system that produces an output $o \in O$ when in state $s \in S$ for an input $i \in I$, where O is the set of all outputs, S is the set of all system states, and I is the set of all inputs. An implementation, M, satisfies a contract if it satisfies all contract guarantees when their associated assumptions hold.

Assuming no disruptions, an invariant contract is one that must always be satisfied when the assumption is true. Invariant contracts are readily expressed and may be represented by deterministic Büchi automatons and by temporal logic [29]. However, most systems are at least partially non-deterministic. Moreover, invariant constructs are not well matched with unknown and unexpected disruptions that might arise from unpredictable environments, internal faults, prolonged system usage, or previously undiscovered interactions with the operational environment. This recognition spurred our research with resilience contracts (RCs).

To address partial observability and need for resilience, we define a mathematical construct called the "resilience contract," which extends the concept of a traditional "contract" from Contract-Based Design (CBD) to address uncertainty and partial observability that contribute to non-deterministic system behavior [13,14]. In a traditional contract, an implementation is said to satisfy a design contract if it fulfills guarantees when the assumptions are true. This is the "assert-guarantee" construct that underlies traditional contracts in CBD. What makes CBD attractive is the fact that statements in the contract are mathematically provable. However, the limitation of a traditional contract in CBD is that the assertions are invariant. This property limits its use in characterizing system reliance. With a resilience contract (RC), the assert-guarantee pair is replaced by a probabilistic "belief-reward" pair. This characterization affords the requisite flexibility while assuring probabilistic verifiability of the model.

The RC is a hybrid modeling construct that combines invariant and flexible assertions and is represented as a Partially Observable Markov Decision Process (POMDP). A POMDP is a special form of a Markov Decision Process that includes unobservable states and state transitions. POMDPs introduce flexibility into a traditional contract by allowing incomplete specification of legal inputs and flexible definition of post-condition corrections [13,14]. The solution of a POMDP determines the optimal action for each probability distribution (belief) over S when taking action $a \in A(s)$.

A POMDP models a decision process in which system dynamics are assumed to be a belief Markovian Decision Process (MDP), a memoryless decision process with transition rewards. A belief state, b_t, is the probability distribution over all states that represents the history of actions and observations up to t.

A belief MDP comprises:

S	Finite set of visible states;
Σ	Set of hidden states;
b_t	Probability distribution representing the history of actions and states;
$A(s)$	Set of actions possible in state s;
$\mathrm{T}(s, s', a)$	Probability of transition from s to s' when taking action $a \in A$;
$\tau(b, a, b')$	The belief state transition function;
$R(s, s', a)$	Expected reward for taking action, a, when transitioning from s to s';
$O(s, a)$	Set of observations.

Let $r_t \in R$ be the reward received at time, t, and define a discount factor, γ, where $0 \le \gamma < 1$ penalizes future rewards and is based on the "cost" for not taking immediate action. The goal of the POMDP is to maximize the expected discounted reward, $E[\sum_{t=0}^{\infty} \gamma^t r_t]$. A policy, π, describes how to select actions for a belief state based on the utility, $U^{\pi}(s)$ of executing π when starting in state s. An optimal policy $\pi^* = argmax U^{\pi}(s)$ maximizes the expected *utility of an action.*

Expected utility may be computed iteratively using a number of methods including a dynamic programming in which the k_{th} value is computed iteratively using Equation (1). The optimal policy is determined by finding the policy that maximizes $U^{\pi}(s)$ for each policy π_i.

$$\begin{gathered} U_0^{\pi}(s) = 0 \\ U_1^{\pi}(s) = R(s, \pi(s)) \\ \cdots \\ U_k^{\pi}(s) = R\big(s, \pi(s)\big) + \gamma \sum_{s' \in \mathrm{S}} \mathrm{T}(s' | s, \pi(s)) U_{k-1}^{\pi}(s') \end{gathered} \tag{1}$$

The updated belief state, b', is computed from the probability of transitioning from s to s' when a given action, a, is taken that results in observation, o.

$$b'(s') = p(s'|o,a,b) = \frac{p(o|s',a)\sum_{s_i \in S} T(s'|s,a)b(s)}{\sum_{s' \in S} p(o|s',a)\sum_{s \in S} p(s'|s,o)b(s)} \quad (2)$$

Simply stated, a RC extends a deterministic contract for stochastic systems. A RC introduces the flexibility needed for the resilience "sense-plan-act" cycle that comprises iteratively sensing the environment and system status (Sense ≡ assumption); sequencing actions that maximize the likelihood of achieving a goal (Plan) and executing those actions (Act ≡ guarantee). The environment sand system health are sensed and assessed after each action. The planning function determines whether to continue with the current plan if the actions accomplish the desired outcome, or otherwise make changes.

As noted earlier, in a POMDP model, some states are unobservable (hidden) because of uncertainties about system state or the outcomes of actions due to incomplete or imperfect information (e.g., noisy sensors). Therefore, the resilience contract approach begins with a naïve model of system behavior comprising known (designed) states and transitions and predicted anomalous states and transitions. New states are added when an observation does not fit any of the existing states, for example, to accommodate unknown-unknowns, and the emission and transition probabilities updated. Note too that an action could involve making another observation, invoking a function, selecting another model, and so forth.

4.2. Example of a Resilience Contract Using POMDP

Figure 2 is a visual, graph-based representation of resilience contracts in a POMDP model. The model comprises belief states, visible states, and hidden states. The belief state is updated using Equation (2) in which τ represents the probability that the system will transition from belief b to b' when an action from A is taken having the highest utility (Equation (1)) and an observation O is made. An observation may result from the current visible state but also might be created from a hidden state. The action also influences transitions in the visible and hidden states. In a resilience contract the assertion is defined by belief state and the guarantee results from a results-based utility function. Significantly, this differs from an invariant contract in which visible states define assertions and the guarantees are invariants.

A POMDP is seeded with visible states, and transition and emission probabilities determined during system design. Hidden states are added that connect to some number of the visible states and to each other. Initially these states are associated with very low, but non-zero transition and emission probabilities. During system operation, a POMDP is trained by comparing its predictive behavior to the behavior of the executing system. As needed, the transition and emission probabilities to hidden and visible states are updated to improve the model. Moreover, as needed, additional hidden states can be added when the initial set of hidden states is used and future observations warrant new hidden states.

Once trained, the model can be analyzed as a conventional Markov Model. For example, it can be used to answer questions such as "what is the probability that the system is in a particular state," or "what is the probability that the system will fail given where it is now?"

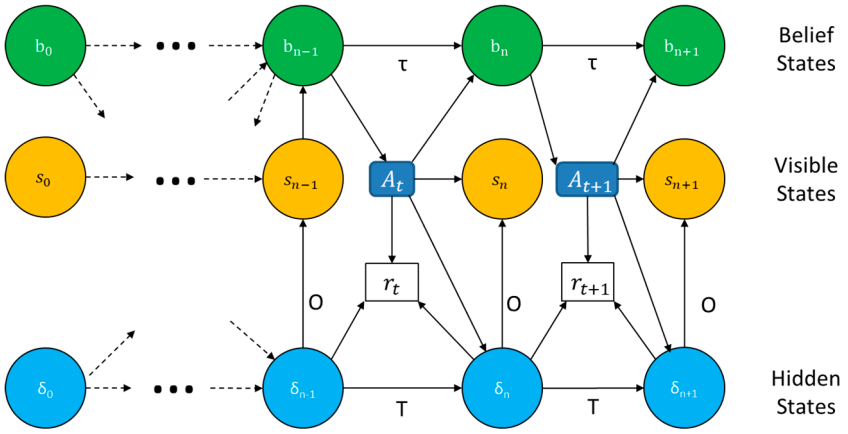

Figure 2. Resilience Contract Based on Partially Observable Markov Decision (POMDP).

Figure 3 is a partial instantiation of a hypothetical state progression for an unmanned aerial vehicle (UAV) mission that begins with nominal health and waiting for a ready indication. The model uses Moore Machine notation, i.e., *s*/*a* means that when in state, *s*, perform action *a* and the transitions between states is determined by Equation (2). Left transition functions, $\tau^{l}_{i,j}$, represent the transition probabilities from state i to state j in the UAV Health model. Similarly, right transition functions labeled $\tau^{r}_{3,0}$ represent the transition probabilities in the UAV mission model.

The UAV is tasked to image a designated reconnaissance target. The *UAV Health* POMDP (on the left) shows states associated with battery health and sends status to the *UAV Motion* POMDP. There are three visible states: Healthy, Battery Yellow and Battery Red, and n hidden states. The n hidden states represent conditions inconsistent with the visible states. The first hidden state sends "Battery Degraded" to the *UAV Motion* POMDP while the n^{th} hidden state sends "Unknown." The distinction highlights that the next state and actions taken depend on the current state and observations.

The right-side of Figure 3 is the Mission POMDP that determines the motion action to take for moving the UAV from its current position to its next position. The right-side also has visible, invariant states, e.g., "Over Target," as well as hidden states, described further below, that represent uncertainty in the UAV position or health.

Figure 3 assumes a routing function (an external input to the UAV Motion POMDP) guides the UAV to its target.

The path from the UAV launch base to the target is along predetermined waypoints. Each waypoint is associated with one or more meta-actions that inform the UAV controller about its next step. A meta-action is a generalized action instantiated at each waypoint and assigned specific parameters determined by analyzing conditions at each waypoint.

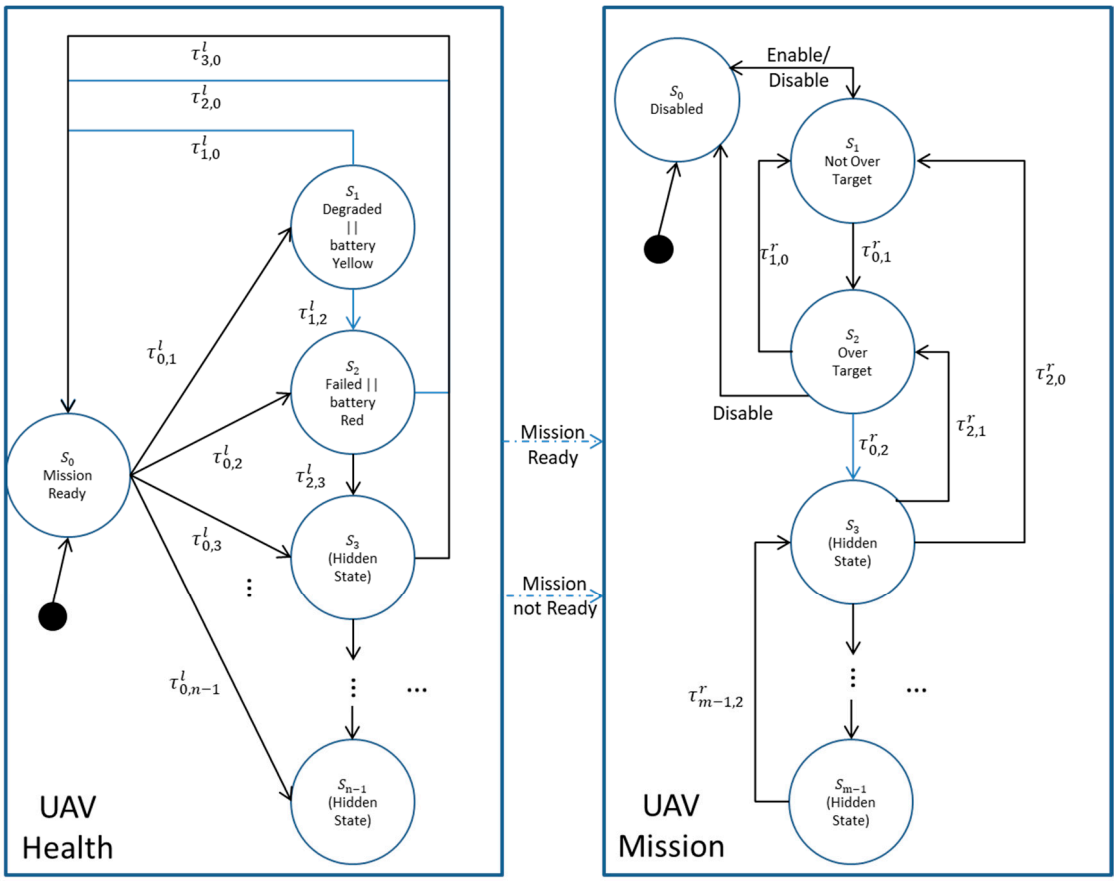

Figure 3. POMDP implementation of unmanned aerial vehicle (UAV) resilience contracts (RCs) for a reconnaissance mission.

We further assume limitations in camera ground sampling distance that result in camera field of view (FOV) quadrants as shown in Figure 4. Consequently, the UAV is certain of the target position if the target falls in the center of a quadrant. However, there is uncertainty when the target falls on or near a quadrant boundary or the UAV is not near the target. The UAV is considered over the target when the target is entirely within the camera FOV. The UAV is deemed to be not over target if the target is not fully within the bounding box of the camera FOV.

Figure 4 shows the UAV positioned over a grid relative to the target position (red star). The UAV camera FOV is shown in blue. The UAV positions itself over the target area and takes images while adjusting its position so that the target is centered under it.

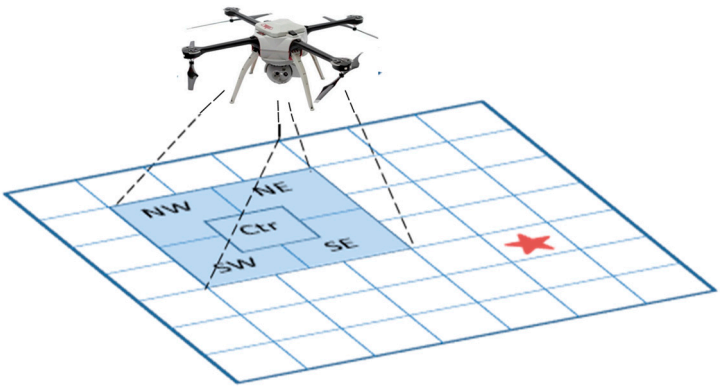

Figure 4. UAV position relative to a reconnaissance target (**red star**) and FOV (**blue**).

4.3. Exemplar Resilience Contracts

Let us assume that *V1* and *V2* are battery voltage thresholds that bear on operational mission decision making. For *V1* > *V2*, the UAV returns to base when its battery voltage drops below *V1* or has a non-critical fault condition. The UAV lands if the battery voltage falls below *V2*, or if the UAV experiences a failure.

Representative RCs for the UAV mission are shown in Table 6. Although the assumptions shown in Table 6 have the appearance of invariants, in the RC construct they represent a decision made on the highest probability $b'(s')$, in the POMDP belief state vector (Equation (2)). For example, the first RC should be interpreted as, "if (the highest probability is that the UAV is not over the target && the highest probability is that the UAV is healthy && the highest probability is that the battery is green) then take the action goto_target."

Table 6. Partial List of Contract for Exemplar Problem.

1.	¬overTarget && healthy && batteryGreen → goto_target
2.	¬batteryRed && degraded || batteryYellow → goto_base
3.	batteryRed || failed → land
4.	unknownHealth || unknownBattery → goto_base
5.	overTarget && CTR && healthy → do_mission & hover
6.	overTarget && NW && healthy → do_mission & goto SE
7.	overTarget && NE && healthy → do_mission & goto SW
8.	overTarget && SW && healthy → do_mission & goto NE
9.	overTarge && SE&&health → do_mission & goto NW

The RC invokes combinations of twelve meta-actions: goto_target, goto_base, land, do_mission, goto_SE, goto_SW, goto_NE, goto_NW, goto_Up, goto_Down, continueAction, and hover. Meta-actions goto_target and goto_base are generalizations comprising multiple directional moves that respectively guide the UAV to its target and return to base. There are multiple propositions defined by the most probable state, e.g., overTarget, healthy (all subsystems working), degraded (one or more subsystems operating below specification), failed (one or more subsystems are non-functional), NW (the target is inside the camera FOV and located north-west of UAV nadir), NE, SW, SE, CTR, batteryGreen (battery voltage is ≥ V1), batteryYellow (battery voltage is < V1 && battery voltage ≥ V2), batteryRed (battery voltage is < V2), unknownHealth, and unknownBattery.

5. Conclusions

In this paper, we investigated the concept of resilience to identify those definitions and interpretations that have contributed to productive lines of query and successful realization of engineered systems. We also formalized the definition of resilience using a rigorous state-based probabilistic modeling construct that enables system (model) verification while having the requisite flexibility to allow the system to respond to disruptions. We offered a prescriptive framework for system modeling approach selection as a function of task and context (i.e., environment characteristics) and provided an illustrative example of modeling based on flexible resilience contracts.

The key points made in this paper are that different system modeling approaches and algorithms are needed based on mission tasks and operational context; adaptive capacity and continual adaptability are the two promising characterizations of resilience that can be cost-effectively realized in real-world systems; and the resilience contract construct is an effective means for probabilistic verification of system model correctness while affording flexibility needed for adaptation and learning. Collectively, these findings contribute to the body of knowledge in both model-based systems engineering (MBSE) and engineered resilient systems.

In conclusion, this paper is intended to stimulate interest in the model-based systems engineering and resilience engineering communities and motivate researchers to explore different

approaches leading to the realization of engineered resilient systems in different domains. Future work, especially in adaptive cyber–physical–human systems, can be expected to benefit from the use of these concepts [30].

Author Contributions: Conceptualization, A.M.M., D.E., and M.S.; methodology, A.M.M.; Validation, A.M.M. and D.E.; formal analysis, A.M.M., D.E. and M.S.; investigation, A.M.M., D.E., and M.S.; resources, A.M.M.; data curation, D.E.; Writing—original draft preparation, A.M.M.; writing—review and editing, A.M.M. and M.S.; visualization, D.E.; supervision, A.M.M.; project administration, A.M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Madni, A.M.; Jackson, S. Towards a conceptual framework for resilience engineering. *IEEE Syst. J.* **2009**, *3*, 181-191.
2. Woods, D. Four Concepts for resilience and the implications for the future of resilience engineering. *Reliab. Eng. Syst. Saf.* **2015**, doi:10.1016/j.ress.2015.03.018.
3. Neches, R.; Madni, A.M. Towards Affordably Adaptable and Effective Systems. *Syst. Eng.* **2013**, *16*, 224–234.
4. Allenby, B.; Fink, J. Social and ecological resilience: Toward inherently secure and resilient societies. *Science* **2000**, *24*, 347–364.
5. Haimes, Y.Y. On the definition of resilience in systems. *Risk Anal.* **2009**, *29940*, 498–501.
6. Vugrin, E.D.; Warren, D.E.; Ehlen, M.A.; Camphouse, R.C. Sustainable Infrastructure Systems: Simulation, modeling, and intelligent engineering. *A Framework for Assessing the Resilience of Infrastructure and Economic Systems*; Gopalakrishnan, K., Peeta, S., Eds.; Springer-Verlag, Inc.: Berlin/Heidelberg, Germany, 2010.
7. Linkov, I.; Kott, A. Fundamental Concepts of Cyber Resilience: Introduction and Overview, I Cyber Resilience of Systems and Networks; Kott, L., Ed.; Springer: Berlin/Heidelberg, Germany, 2018.
8. Gao, J.; Barzel, B.; Barabasi, L. Universal Resilience Patterns in Complex Networks. *Nature* **2016**, *530*, 307–312, doi:10.1287/orsc.2.1.71.
9. Doyle, J.; Alderson, D.; Li, L.; Low, S.; Roughan, M.; Shalunov, S.; Tanaka, R.; Willinger, W. The "robust yet fragile" nature of the internet. *Proc Natl. Acad. Sci. USA* **2005**; *102*, 14497–14502.
10. Folke, C. Resilience: The emergence of a perspective for social-Ecological systems analyses. *Glob. Environ. Chang.* **2006**, *16*, 253–267, doi:10.1016/j.gloenvcha.2006.04.002.
11. Manyena, S.B. The concept of resilience revisited. *Disasters* **2006**, *30*, 433–450.
12. Sievers, M.; Madni, A.M. Agent-Based Flexible Design Contracts for Resilient Spacecraft Swarms. In Proceedings of the AIAA Science and Technology 2016 Forum and Exposition, San Diego, CA, USA, 4–8 January 2016.
13. Sievers, M.; Madni, A.M. Contract-Based Byzantine Resilience for Spacecraft Swarm. In Proceedings of the 2016 AIAA Science and Technology Forum and Expo, Grapevine, TX, USA, 9–13 January 2017.
14. Madni, A.M.; D'Ambrosio, J.; Sievers, M.; Humann, J.; Ordoukhanian, E.; Sundaram, P. Model-Based Approach for Engineering Resilient System-of-Systems: Applications to Autonomous Vehicle Network. In Proceedings of the Conference on Systems Engineering Research, Redondo Beach, CA, USA, 23–25 March 2017.
15. Longstaff, P.H.; Koslowski, T.G.; Geoghegan, W. Translating Resilience: A Framework to Enhance Communication and Implementation. In Proceedings 5th Symposium on Resilience Engineering, Soesterberg, Netherlands, 24–27 June 2013.
16. Avizienis, A.; Laprie, J.-C.; Randell, B.; Landwehr, C. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Trans. Dependable Secur. Comput.* **2004**, *1*, doi:10.1109/TDSC.2004.2.
17. Miller, A.; Xiao, Y. Multi-Level strategies to achieve resilience for an organisation operating at capacity: A case study at a trauma centre. *Cogn. Technol. Work* **2007**, *9*, 51–66.
18. Mili, L. Making the concepts of robustness resilience and sustainability useful tools for power system planning, operation and control. In Proceedings of the ISRCS2011: 4th International Symposium on Resilient Control Systems, Boise, ID, USA, 9–11 August 2011,

19. McCabe-Dansted, J.; Dixon, C. CTL-Like Fragments of a Temporal Logic of Robustness. In Proceedings of the 2010 17th International Symposium on Temporal Representation and Reasoning, Paris, France, 6–8 September 2010.
20. Pokorny, L.R.; Ramkrishnan, C.R. Modeling and Verification of Distributed Autonomous Agents Using Logic Programming. In Proceedings of the Declarative Agent Languages and Technologies II, New York, NY, USA, 19 July 2004.
21. Baltrop, K.J.; Pingree, P.J. Model Checking Investigations for Fault Protection System Validation. In Proceedings of the 2003 International Conference on Space Mission Challenges for Information Technology, Pasadena, CA, USA, 13–16 July 2003.
22. Cimatti, A.; Dorigatti, M.; Tonetta, S. OCRA: A tool for checking the refinement of temporal contracts. In Proceedings of the IEEE/ACM 28th International Conference on Automated Software Engineering (ASE), Palo Alto, CA, USA, 13–15 November 2013.
23. Wong, W.C.; Lee, J.H. Fault Detection in Process Systems using Hidden Markov Disturbance Models. In Proceedings of the 8th International Symposium on Dynamics and Control of Process Systems, Cancun, Mexico, 6-7 June 2007.
24. Modgil, S.; Faci, N.; Meneguzzi, F.; Oren, N.; Miles, S.; Luck, M. A Framework for Monitoring Agent-Based Normative Systems. In Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary, 10–15 May 2009; Decker, Sichman, Sierra, Castelfranchi, Eds.
25. Sangiovanni-Vincentellik, A.; Damm, W.; Passerone, R. Taming Dr. Frankenstein: Contract-Based Design for Cyber-Physical Systems. *Eur. J. Control* **2012**, *18*, 217–238.
26. Meyer, B. Towards More Expressive Contracts. *J. Object Oriented Program. July* **2000**, 39–43.
27. Le Traon, Y.; Baudry, B. Design by Contract to Improve Software Vigilance. *IEEE Trans. Softw. Eng.* **2006**, *32*, 571–586.
28. Cimatti, A.; Tonetta, S. A Property-Based Proof System for Contract-Based Design. In Proceedings of the 2012 38th Euromicro Conference on Software Engineering and Advanced Applications, Izmir, Turkey, 5–8 September 2012.
29. Büchi, J.R. On a Decision Method in Restricted Second-order Arithmetic. In *The 1960 Congress on Logic, Methodology and Philosophy of Science*; Stanford University Press: Stanford, CA, USA, 1962.
30. Madni, A.M.; Sievers, M.; Madni, C.C. Adaptive Cyber-Physical-Human Systems: Exploiting Cognitive Modeling and Machine Learning in the Control Loop. *Insight* **2018**, 87–93, doi:10.1002/j.2334-5837.2018.00534.x.