# Assurance Framework for Autonomy-Capable Tactical Systems (AFACTS)
## Technical Areas (TAs) 1, 2, and 3
# Volume 1 – Technical and Management Proposal

**Prepared For:**
Defense Advanced Research Projects Agency
Information Innovation Office
675 North Randolph Street
Arlington, VA 22203-2114

| Assurance Framework for Autonomy-Capable Tactical Systems (AFACTS) | |
|---|---|
| BAA Number | HR001117S0045 |
| Technical Area(s) | TA1, TA2, and TA3 |
| Proposal title | Assurance Framework for Autonomy-Capable Systems (AFACTS) |
| Lead organization name | Sierra Nevada Corporation (SNC) |
| Type of organization | Large Business |
| Technical point of contact (POC) | Dr. Jeff Smith; Sierra Nevada Corporation, 3076 Centreville Rd., Herndon, VA 20171; (703) 464-6434 jeff.smith@sncorp.com |
| Administrative POC | Diana Artemis; Sierra Nevada Corporation, 3076 Centreville Rd., Herndon, VA 20171; (703) 464-6442 diana.artemis@sncorp.com |
| Award instrument | Cost Plus Fixed Fee |
| Total amount | $10,351,767 |
| Place and period of performance | SNC's office located at: 3076 Centreville Rd., Herndon, VA 20171 [Cage: 5DHT3]; POP: 4/2/2018 – 3/31/2022 |
| Primary subcontractors | Securboration, Inc. (S), Lee Krause, Small Business, 1050 West NASA Blvd, Suite 155, Melbourne, FL 32901, lkrause@securboration.com, (321) 591-9836<br>Vanderbilt University (VU), Doug Schmidt, University, 2201 West End Ave, Nashville, TN 37235, schmidt@dre.vanderbilt.edu, (615) 322-2631<br>Northeastern University (NU), Mitch Kokar, University, 360 Huntington Ave., Boston, MA 02115, kokar@coe.neu.edu, (617) 373-4849 |
| Proposal validity period | 120 days |
| DUNS number | 094373495 |
| Taxpayer ID No. | 88-0094415 |
| CAGE code | 8X691 |
| Reference no. | A9792 |

**October 19, 2017**

# Table of Contents

I.A Executive Summary 1

I.B Innovative Claims and Deliverables 5

I.B.1 Technology Transition 5

I.B.2 Life Cycle and Sustainment Risks 6

I.B.3 Proprietary Claims and Fundamental Research 6

I.C Technical Rationale, Technical Approach, and Constructive Plan 7

I.C.1 Detailed Technical Rationale 7

I.C.1.1 Detailed TA1 Technical Rationale – MODel intEgrated fRamework for autoNomous, hIgh aSsurancE Design (MODERNISED) 7

I.C.1.2 Detailed TA2 Technical Rationale – Real-time Operation VErified Reconfigurator (ROVER) 9

I.C.1.3. Detailed TA3 Technical Rationale – Dynamic Assurance Inferencing System (DAIS) 11

I.C.2 Detailed Technical Approach 12

I.C.2.1 Detailed TA1 Technical Approach – MODel intEgrated fRamework for autoNomous, hIgh aSsurancE Design (MODERNISED) 13

I.C.2.1.1 Overview 13

I.C.2.1.2 Modeling and Reasoning Approach 14

I.C.2.1.3 MODERNISED Approach Applied to Formation Classification Example 16

I.C.2.2. Detailed TA2 Technical Rationale – Real-time Operation VErified Reconfigurator (ROVER) 17

I.C.2.2.1 Overview 18

I.C.2.3.2 ROVER Internal Operation 18

*I.C.2.3.2.1 Ingesting TA1 Models into ROVER Monitoring and Control Services* 18

*I.C.2.3.2.2 Lightweight Statistical Monitors (LSM)* 19

*I.C.2.3.2.3 Adaptive Reconfiguration Management Service (ARMS)* 20

*I.C.2.3.2.4 Alert Management Service (AMS)* 21

I.C.2.3. Detailed TA3 Technical Approach – Dynamic Assurance Inferencing System (DAIS) 21

I.C.2.3.1 Overview 21

I.C.2.3.2 DAIS Internal Operation 22

*I.C.2.3.2.1 Ratiocination Engine* 22

*I.C.2.3.2.2 Confidence Engine* 23

I.D. Management Plan I.D.1 Team Organization 25

I.D.2 Program Management 25
I.D.3 Risk Management 26
I.E Personnel, Qualifications, and Commitments 27
I.F Capabilities 30
I.F.1 Previous Accomplishments and Related Work 30
I.F.2 Facilities 31
I.G Statement of Work, Schedule and Milestones 32
I.G.1 SoW 32
I.G.2 Metrics 35
Appendix A 36
Appendix B – Bibliography 38

# List of Figures

**Figure 1. Architectural Components and Interactions in AFACTS……..1**
**Figure 2. Agents Running in an AFACT-based LE-CPS……7**
**Figure 3. Formations are Separable in 2D Angle Space……12**
**Figure 4. GSN Representation of Formation Classification……..13**
**Figure 5. Reduces Risk via Designs Supporting Multi-level Modeling, Verification, Abstraction & Languages……15**
**Figure 6. A Top-level GSN Ontology with Property Names……16**
**Figure 7. A Simple Domain Ontology with Property Names……16**
**Figure 8. TA2 interaction with TA1 & TA3……..17**
**Figure 9. Example of Ratiocination……22**
**Figure 10. AFACTS Team Organization……25**
**Figure 11. Commercial Imagery Lab Facilities at the MACE……..31**
**Figure 12. Schedule, Milestones, and Deliverables……34**

# List of Tables

**Table 1. How AFACTS Technologies Address BAA Evaluation Objectives and Impact LE-CPS** ... 3
**Table 2. AFACTS Innovations** ... 5
**Table 3. The AFACTS Approach to Mitigating Software Life-cycle & Sustainment Risks** ... 6
**Table 4.Properties from the Domain Ontology** ... 16
**Table 5. AFACTS Addresses Technical Risks Through Mitigation Options & Decision Points in the Project Plan** ... 26
**Table 6. AFACTS Combines Top Leaders in key Model-Integrated Computing, Generative Programming, QoS-aware Middleware, and Formal Methods technologies and standards** ... 27
**Table 7. Time Commitments for Key Individuals on the AFACTS Team** ... 29
**Table 8. The AFACTS Team Builds on a Solid Foundation of Research and Technology Transition Success** ... 30
**Table 9. The AFACTS Statement of Work** ... 32
**Table 10. Assured Autonomy Program Metrics for AFACTS Evaluation Measure Progress Against All Thrusts** ... 35

## I.A Executive Summary

Developing and certifying mission-critical *Learning-Enabled Cyber-Physical Systems* (LE-CPS) is essential to meet current and planned DoD mission needs as in cooperative autonomous system control for situational awareness, ground and air-based UAVs, cognitive workload reduction, and force protection missions. Due to variations in mission operating environments, it is unrealistic to expect LE-CPS to behave deterministically and statically under all conditions.

A key problem facing developers, integrators, and certifiers of LE-CPS is that the techniques (e.g. convolutional neural nets and support vector machines) that are commonly used to create learning-enabled components are hard to analyze formally due to the large state space as they exhibit nonlinearity [Kuroe] and are expected to operate and learn safely even in previously unseen environments. These properties makes any reachability based techniques for verification intractable for these systems [Adams][Cavalcanti]. Additionally, the software environment in which these system operate, might itself exhibit failures. For example, conventional run-time platforms upon which LE-CPS perform computations do not monitor and enforce *quality-of-service* (QoS) constraints [Gray1] in a resilient manner that operate dependably in the face of (partial) failures and cyber-attacks [Zhu].

To overcome limitations with conventional approaches, Sierra Nevada Corporation, teamed with Northeastern University, Vanderbilt University, and Securboration, proposes a new integrated framework called the *Assurance Framework for Autonomy-Capable Tactical Systems* (AFACTS). AFACTS provides **probabilistic QoS assurance that combines advances in design-space analysis and run-time platforms to safely and dependably handle variations in environmental conditions during mission-critical operations**, as shown in Figure 1. This figure shows how AFACTS (1) integrates design--time risk analysis methods, techniques, and tools

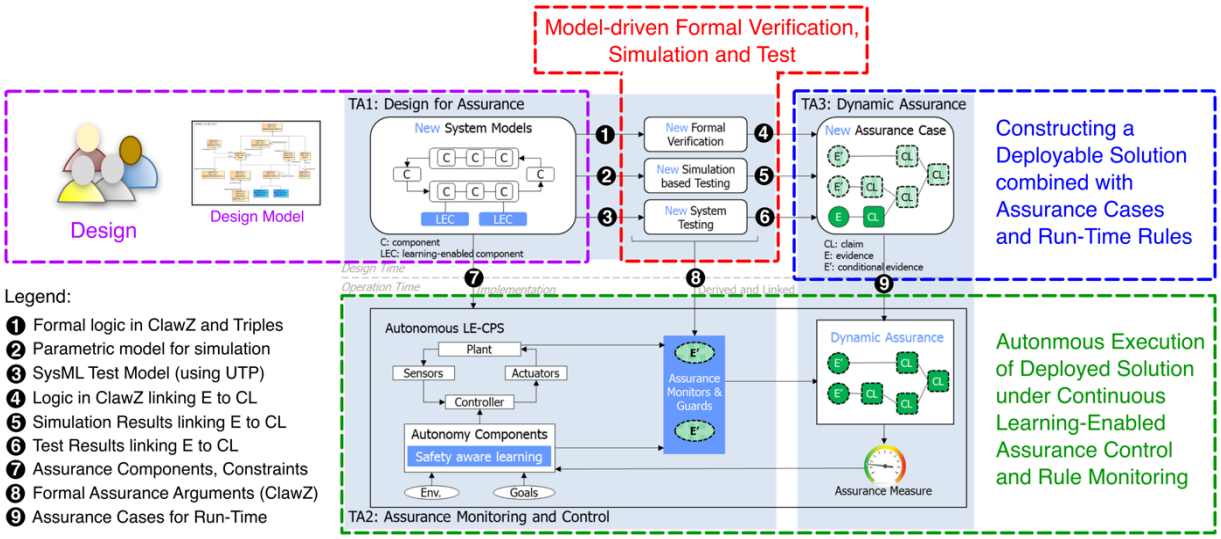



*Figure 1. Architectural Components and Interactions in AFACTS*

with a novel run-time platform that supports decision making (e.g., data collection, design optimization, and operational risk management) and (2) applies this integrated design-time/run-time framework to assure the proper functioning of production-level LE-CPS under realistic (i.e., harsh/uncertain) operational conditions.

We will share AFACTS tools, processes, and advances in dependence learning, uncertainty aggregation, and QoS management, developed in the context of the *Unmanned Tactical Control & Collaboration* (UTACC) testbed, with both the *Enterprise Engine* (E2) programs (for transition) and TA4 vendors (for Assured Autonomy collaboration). UTACC currently provides monitoring and control software for robots that replace one Marine in a 4-Marine fire team performing the same maneuvers. We plan to add cooperation to the autonomy by adding more robots to the fire team. The UTACC testbed contains a demonstrated simulation of this fire team moving across terrain and changing formations in prior work, as described in §I.F.1.

AFACTS will enhance productivity of LE-CPS development by scaling to 100 dimensions, with 10% less overhead, providing at least 1,000 conditional evidence, and .001x reduced trials to assurance (described in §I.G.2) using the following innovations for each technical area (TA):

**TA1 – MODel intEgrated fRamework for autoNomous, hIgh aSsurancE Design (MODERNISED).** The *model-integrated computing* (MIC) framework provided by the AFACTS TA1 performers extends the state-of-the-art by deriving formal descriptions from pattern-analysis of models and automating the transformation from models to formal analysis, simulation, and test code to an assurance-based proof system. The result is a method that enables highly-focused measurable conditions to monitor and control the learning process of cognitive agents during LE-CPS run-time. Conventional approaches to developing LE-CPS are limited by non-robust designs stemming from the difficulty of analyzing LE-CPS nonlinearities due to the intractability of model reachability computations. To address this limitation, MODERNISED applies an ensemble of validation approaches with advances in dependence learning, uncertainty aggregation, and QoS management demonstrated in TA4 and team-provided LE-CPS testbeds. Sierra Nevada Corp (SNC)/Northeastern (NU) will lead the TA1 effort.

**TA2 – Real-time Operation VErified Reconfigurator (ROVER).** The middleware-based QoS-aware monitoring and control platform provided by the AFACTS TA2 performers hosts the cognitive agents created in TA1, enabling continuous re-valuation and assurance of dynamic approaches to achieving a given set of goals. Conventional approaches for developing LE-CPS are limited by faulty failure identification with corrective actions stemming from the need to resiliently monitor and enforce QoS constraints of LE-CPS computations that must operate across failures and attacks. To address this limitation, ROVER combines advances in design-time analysis and run-time platforms to safely and dependably handle variations in environmental conditions. To alleviate the performance overhead stemming from excessive evidence passed from TA2 to TA3 services, ROVER's lightweight statistical monitors and alert management service support a filter and re-query approach that issues a small number of alerts to TA3, which then solicits additional metrics as needed. Vanderbilt University (VU) will lead the TA2 effort.

**TA3 – Dynamic Assurance Inferencing System (DAIS).** The metrics associated with TA1-based design-time conditional evidence and TA2-based monitored/control evidence comparison will be combined to guide AFACTS unsupervised, safety-aware learning component. Conventional approaches to developing LE-CPS are limited by the low quality assurance cases (ACs) stemming from the difficulty of synthesizing these ACs directly from engineering artifacts. To address this limitation, DAIS provides a "meet-in-the-middle" approach in which coarse-grained ACs are augmented with domain knowledge to produce high-fidelity ACs. Conventional approaches are limited by spurious confidence values due to the fact that a single AC confidence

evaluation framework is not appropriate for all AC claims. To address this limitation, DAIS employs a generic evaluation architecture that is agnostic of a single confidence framework. Finally, conventional approaches are limited by spurious confidence values stemming from emergent conditions that cannot be anticipated at design-time. To address this limitation DAIS applies abductive logic to dynamically synthesize new AC constructions that account for emergent conditions. Securboration will lead the TA3 effort.

Table 1 summarizes how the technical areas covered by the AFACTS team addresses the Assured Autonomy TA1-3 program objectives and how the AFACTS team will interact with TA4.

*Table 1. How AFACTS Technologies Address BAA Evaluation Objectives and Impact LE-CPS*

| BAA Objectives | AFACTS Technology Approach | Impact | Section |
|---|---|---|---|
| Develop & integrate executable modeling languages & model-integrated computing tools for LE-CPS design & verification | TA1. Develop, integrate & validate powerful multi-domain design-time risk analysis methods, abstractions, techniques, simulations & tools supporting decision making essential assuring proper functioning of LE-CPS under realistic operational conditions | Improved reliability from systematically assessing variations in operational trustworthiness of LE-CPS | §I.C.1.1, §I.C.2.1 |
| Produce qualified evidence regarding safety & correctness of LE-CPS design and operation | TA2. Develop, integrate & validate a QoS-aware assurance monitoring and control platform that (1) continuously & dependably checks the overall correctness of component execution within an LE-CPS to enable reactive and proactive corrections & (2) ensures that QoS, structural & behavioral variability of individual & aggregate components remain within the expected statistical bounds | Operational trustworthiness and availability by ensuring LE-CPS run within the envelope of competence & QoS in a range of operational conditions | §I.C.1.1, §I.C.2.x, §I.C.2.3.2 |
| Automate assurance case synthesis & run-time evaluation for LE-CPS | TA3. Create a framework that supports the dynamic evaluation of assurance cases (AC) using evidence gleaned from running software. Synthesize new AC constructs that account for emergent behaviors when they are (1) anticipated at design-time via domain knowledge or (2) encountered at run-time. Trigger appropriate reactive logic within the LE-CPS when goals are unlikely to be met based on run-time conditions. | LE autonomous systems, verifiable in real-time, by quantifying the degree to which LE-CPS meets design goals operating in a complex & unpredictable environment | §I.C.1.3, §I.C.2.3 |
| Apply to production - level LE-CPS | TA4. In addition to active participation and application to TA4 platform, we provide (from the kickoff of the program) a production-level LE-CPS that supports key requirements by extending existing framework. | Built-in technology transition plan that immediately impacts key DoD LE-CPS missions | §I.B.2 |

DoD LE-CPS have historically been developed via *multiple technology bases*, where each system brings its own networks, computers, displays, software tools/platforms, and people. Unfor-

tunately, these proprietary "stovepipe" architectures tightly couple many functional and QoS aspects of LE-CPS, which greatly impedes their adaptability, assurability and affordability. The affordability of certain DoD systems, such as logistics and planning, can be enhanced by commercial-off-the-shelf (COTS) technologies. However, DoD procurement efforts aimed at integrating COTS do not support affordability *and* assurability/adaptability for mission-critical LE-CPS since (1) they only address initial non-recurring acquisition costs, but fail to reduce recurring software lifecycle costs and (2) compromise adaptability and assurability due to poor QoS support in COTS software, e.g. minor perturbations in today's COTS-based solutions can cause massive failures that impact life and property.

In general, conventional COTS software is not suitable for mission-critical LE-CPS systems due to either being (1) *flexible and standard*, but incapable of ensuring stringent behavioral and QoS requirements, or (2) partially QoS-enabled, but inflexible and non-standard. Thus, the rapid progress in COTS software for enterprise information technology systems is not yet applicable for mission-critical LE-CPS. Until this problem is resolved, LE-CPS system integrators (and ultimately warfighters) will be unable to take effective advantage of future advances in COTS. **Developing the new generation of adaptive, assurable, and affordable LE-CPS technologies is essential for US national security.**

AFACTS will dramatically simplify development, optimization, validation, and integration in DoD LE-CPS. AFACTS will allow researchers and system integrators to develop and evolve complex LE-CPS assurably, adaptively, *and* affordably by (1) standardizing COTS at the multiple levels (including MIC tools and QoS-aware middleware), rather than *just* at lower-level hardware/networks/OS layers), and (2) devising optimizers, meta-programming techniques, and multi-level distributed dynamic resource management protocols/services that will enable DoD LE-CPS to customize standard COTS interfaces and tools, without the penalties incurred by today's COTS implementations.

Many DoD LE-CPS require (or will require) the capabilities and optimizations resulting from AFACTS. If the proposed design-time integration of multiple levels of abstraction, modeling, code generation, reasoning and formal methods of autonomous systems are not adapted to w.r.t. metrics derived from autonomous operation, developers of DoD LE-CPS will continue to use multiple proprietary technology bases that will continuously reinvent and maintain theoretically involved and computationally complex solutions to provide sufficient confidence that LE-CPS meet demanding mission needs. DoD LE-CPS will continue to be excessively expensive, time consuming and brittle since they will be built upon an obsolete technology base devised to meet relatively stable Cold War threats, rather than rapidly evolving next-generation threats, such as environmental, economic, terrorist, and information warfare threats. Increasingly, our adversaries are smaller, more mobile, and are already using COTS technologies against us. Although today's COTS technologies are suitable for our adversaries asymmetric warfare threats, they are not yet suitable for our *defense* since they do not satisfy mission-critical DoD LE-CPS needs, *i.e.*, dependability, scalability, security, evolvability, and timely acquisition procedures.

We will meet the schedule and *Statement of Work* per *Work Breakdown Structure*, described by *Phase*, for 48 months (see §I.G), which includes tasks and deliverables that enable execution of program goals at the cost summarized on this cover sheet and detailed in the *Cost Volume*.

## I.B Innovative Claims and Deliverables[1]

Table 2 summarizes the innovations provided by AFACTS and references where these innovations are discussed further in this proposal.

*Table 2. AFACTS Innovations*

| What is Revolutionary in AFACTS | Uniqueness & Benefits | AFACTS Deliverable | Proposal Ref |
|---|---|---|---|
| AFACTS derives formal descriptions from model pattern-analysis of learning components & automated transforms from models to formal, simulation & test code to an assurance-based proof system for focused, measurable conditions to monitor & control learning during LE-CPS operation. | AFACTS surpasses current non-robust designs stemming from LE-CPS nonlinearities, applying a unique ensemble of validation approaches with advances in dependence learning, uncertainty aggregation & QoS management, while ensuring that the goals are computable & reachable within mission time constraints. | *MODel intEgrated fRamework for autoNomous, hIgh aSsurancE Design* (MODERNISED) based on multiple abstractions, tools and formalisms in CDRLs 1, 2 & 3 | MODERNISED in §I.C.2.1 & CDRLs in §I.G |
| A middleware-based QoS-aware monitoring & control platform host cognitive agents, thereby enabling continuous re-valuation & assurance of dynamic approaches to achieving a given set of goals. | AFACTS overcomes conventional LE-CPS computations limited by poor failure identification with corrective actions and a framework for probabilistic QoS assurance combining advances in design-time analysis & run-time platforms to handle unprecedented variations in environmental conditions with a small number of alerts. | *Real-time Operation VErified Reconfigurator* (ROVER) monitoring and control software in CDRLs 1, 2& 3 | ROVER in §I.C.2.2 & CDRLs in §I.G |
| Metrics associated with design-time conditional evidence & monitored/control evidence comparison will be combined to guide unsupervised, safety-aware learning component. | AFACTS improves upon low quality assurance cases (ACs) & spurious confidence values from emergent conditions unanticipated at design-time. Its ACs are augmented with domain knowledge, a generic evaluation architecture agnostic of a single confidence framework & a unique combination of deductive, inductive & abductive reasoning to dynamically synthesize ACs. | *Dynamic Assurance Inferencing System* (DAIS) assurance case design and operation, along with metrics to detect delta between cases, in CDRLs 1, 2, & 3 | DAIS in §I.C.2.3 & CDRLs in §I.G |
| Assurance cases design & operation have not been applied successfully *at-scale* to safety-critical fielded DoD LE-CPS. AFACTS includes a low-risk, built-in technology transition plan to such safety-critical LE-CPS. | In addition to active participation & application to TA4 platform, AFACTS provides a production-level LE-CPS that supports key requirements with our extended framework. A built-in technology transition plan that will have immediate impact on key DoD LE-CPS missions. | Deployment package & associated documentation as part of CDRLs #2 & 3. | §I.B.2 |

### I.B.1 Technology Transition

The background of the *Enterprise Engine* (E2) and the QoS-aware middleware and applications we will extend appears in §II.F1. We will share AFACTS tools, processes, and advances in dependence learning, uncertainty aggregation, and QoS management, developed in the context of the UTACC testbed, back to both the E2 programs (for transition) and TA4 vendors (for Assured Autonomy collaboration). SNC is currently working on E2-next-generation, transitioning to high-available cognitive agent technology, introducing machine learning and temporal logic-based reasoning. The E2 program is expanding into Space and Radar systems as E2 can be used

[1] This is also called "Goals and Impacts" in section a) on pg. 27 of the BAA.

to integrate legacy airborne (F-16 TARS, Reaper, Predator, U2, Global Hawk, JSTARS, AWACS) and future ISR platforms into DCGS. E2 could be useful for rapidly integrating ground, air, and space assets into a cross-domain user-defined operations picture and collection management tool for situational awareness and tasking.

VU's work on QoS-aware middleware has transitioned to many DoD systems, including the Joint Tactical Radio System software defined radio program, manned/unmanned combat air vehicles, the Orbital Express low earth orbit satellite telemetry and control framework, the Ground Support System for the X33 Single Stage To Orbit Reusable Launch Vehicle, and the USS Ronald Reagan and Gerald Ford aircraft carriers, the USAF upgraded early warning radar system, the DMSO HLA/RTI and DISA TENA distributed interactive simulation middleware, among many other DoD applications. VU's work on dynamic resource management algorithms, QoS-aware component deployment and configuration middleware for system integration, and model-based tools for system execution modeling and performance analysis has transitioned to many DoD acquisition programs, including the Navy's DDG 1000 land attack destroyer and Submarine Warfare Federated Tactical System programs, as well as the Army's Common Operating Environment efforts. The QoS-aware middleware developed and transitioned by the VU team are highly relevant to the proposed effort for the DARPA Assured Autonomy program.

### I.B.2 Life Cycle and Sustainment Risks

Our AFACTS team has extensive experience transitioning software technologies to all DoD services (see Technical Transition §I.B.2 for examples). Table 3 describes typical risks encountered when transitioning DARPA software and the AFACTS approach to mitigating risks.

*Table 3. The AFACTS Approach to Mitigating Software Life-cycle & Sustainment Risks*

| Software Life-Cycle /Sustainment Risk | Approach/Mitigation |
|---|---|
| Increasing cost due to license management & royalties/fees | AFACTS software is provided with Unlimited Rights |
| High maintenance costs due to immature research software that is not sufficiently documented or tested | Our agile Research Process for software development is compatible with the tempo of a DARPA program while also providing a path for later transition to more formal software standards required by operational systems |
| Software is delivered to untrained engineers/users who cannot use it | The AFACTS deployment package (CDRL #3) assists new users (and evaluators). SNC provides dedicated staff to support deployed systems, including software developers who are intimately aware of the deployed system (for example, UTACC engineers deployed to Fort A.P. Hill to make demonstration successful). |
| Software is not transferable across hardware platforms | AFACT software uses and drives open standards and architectures & can be compiled and run on most systems (including Linux and embedded systems) |

### I.B.3 Proprietary Claims and Fundamental Research

SNC and its subcontractors make no proprietary claims or restrictions to any part of the Assured Autonomy effort. Although we will conduct advanced research, development, and assurance on topics pertaining LE-CPS, there are no intentions to assert the fundamental research exemption for technologies created under this program. While there are no formal teaming arrangements between SNC and its subcontractors, the team has committed to working together closely throughout the duration of the project across all Technical Areas.

# I.C Technical Rationale, Technical Approach, and Constructive Plan

## I.C.1 Detailed Technical Rationale

### I.C.1.1 Detailed TA1 Technical Rationale – MODel intEgrated fRamework for autoNomous, hIgh aSsurancE Design (MODERNISED)

AFACTS-based LE-CPS pursue their predefined goals at run-time (i.e., after component deployment and activation) by utilizing and depending on their onboard capabilities, knowledge, and algorithms. Since LE-CPS run completely autonomously in many environments (i.e., without relying on any outside support), AFACTS must address the following ongoing research challenges [Gaudin][Ciora]:

- On-board systems must be extremely reliable, self-healing, and prepared for unexpected situations
- The design and build infrastructure capable of producing such systems requires an unprecedented level of definition-, simulation-, verification-, production-, and testing-capabilities that are dependable and scalable.

In response to these challenges, we propose the *MODel intEgrated fRamework for autoNomous, hIgh aSsurancE Design* (MODERNISED). **MODERNISED provides a LE-CPS system architecture and formal methods toolkit based on cognitive agent technology**. Agents are autonomous systems acting without external supervision on events produced by state changes in the surrounding environment, or within the agent itself. Events are asynchronous and volatile occurrences within a particular LE-CPS that represent something happening (or is contemplated as having happened) in that domain.

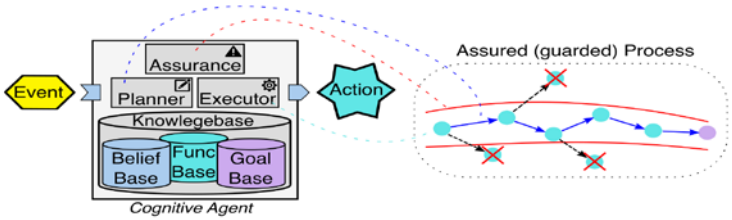



*Figure 2. Agents Running in an AFACT-based LE-CPS*

The level of sophistication in handling desires and intentions categorizes the agent. For example, *simple* agents respond to events based on a limited set of fixed rules. C*ognitive agents* maintain a knowledge-base ("mental state"), consisting of a *belief base* (expressing the knowledge and perceived information the agent has about itself and the surrounding environment), *goal base* (mission-specific goal definitions and plans how to reach these goals) and *function base* (capabilities and associated rules available to the agent to construct actions) [Horkoff][Riemsdijk1][KokarEndsley]. The left part of Figure 2 shows the cognitive agent architecture applied in AFACTS.

MODERNISED will work together with capabilities defined by TA2 and TA3 performers to provide the following capabilities to LE-CPS:

1. **Dynamic run-time adaptation and verification of agent plans**. The ability of cognitive agents to dynamically alter plans (or do substantial replanning) at run-time is well researched [Riemsdijk2]. Ensuring that the plan execution reliably reaches the desired goal and calculating any required plan alteration to support this is an open research problem [Koeman][Sabutucci][Riemsdijk3]. Our AFACTS work will extend prior reasoning methods with dynamic guards constraining and adapting the execution plan, as shown in Figure 2. LE-CPS are typically resource-constrained systems, paired with time-critical execution requirements. To ensure that the dynamic plan assurance and revision at run-time (which is the fo-

cus of TA2 and TA3) can be performed while guaranteeing a high assurance of correctness, intensive preparations are necessary prior to deployment [Neema2] (which is the focus of TA1). TA1 uses the systems engineering modeling language SysML for the AFACTS LE-CPS design [OMG1].

2. **Archetype modeling supports unlimited mission-cases through one model**. Rather than creating a monolithic model for each LE-CPS mission case, AFACTS uses an *archetype modeling* methodology which creates one mission-case-agnostic system model with broad coverage sufficient to handle all mission-cases [OMG9]. This model is then customized by narrowing down each mission-case through the application of mission-specific constraints. For example, if a subsystem is designed to assume 100 different states, but a certain mission allows only 10 out of these 100, then the mission-specific constraints will narrow down the model to allow only these 10 states. This narrowing process will be applied without imposing LE-CPS design or implementation changes.
3. **Enhanced design assurance through executable modeling techniques**. Executable modeling languages [OMG1] [OMG2] [OMG3] [OMG4] [OMG5] and *Model-Integrated Computing* (MIC) tools [Bapty][Schmidt1][Neema1][Neema3] provide a first line of assurance through detailed simulation of the complete LE-CPS behavior at model level [OMG6][OMG7][OMG8]. This approach allows LE-CPS designers to (visually) exercise any behavior of the design directly in the modeling tool. Provided interfaces between the modeling tools and implementations of mathematical simulation tools (e.g., Simulink and Modelica) [OMG6] and formal specification languages (e.g., ClawZ [Adams][Arthan][Vernob][Tehrani], Z++ [Adesina-Ojo], OWL [OWL], SHACL [SHACL]) allow extended parametrized simulations and formal verification of LE-CPS designs that guarantee decidability and can be executed within the time complexity requirements of the mission applications [Chakrabarti]. Simulation, formal verification and model-based testing is applied during TA1 to the full executable system model and its implementations. Assurance cases defined and configured in TA3 evaluate how well mission-specific narrowed archetype models and implementations perform.
4. **Mission-case goals modeled in Goal Structuring Notation (GSN)** [GSN-Std]. GSN is a graphical notation for presenting the structure of safety arguments that provides a formalism for describing how a particular set of claims has been shown to be true by means of evidence. GSN is the syntactical, standard bridge between SysML, ontologies, simulation and formal methods [Groza]. TA1 provide a tool for transforming GSN specifications into executable SysML, archetype model constraints, and ontologies. The same assurance chain utilized for the LE-CPS system components will be applied to the goal-related knowledge and algorithmic definitions. Our approach will extend the use of GSN to express claims and evidence associated with the learning processes of TA2.

In the AFACTS TA1 modeling approach, all designed behavior is fully specified through formal algorithms, which allows automated correctness proving, design space analysis [Sumit1], and model-based simulation of every mission aspect. Our novel archetype approach supports rapid provisioning for existing and new mission profiles without the need for redesign or reimplementation. Our state machine (Cameo state machine simulator connected to Simulink via parametric modeling), Z-based [Adams][Vernon][Arthan] and ontology [OWL] [SHACL] formalisms we are combining for TA1 is rich enough to automatically derive assurance and model-based test cases from models and yet computable within the mission time constraints. Expert and learning

systems are used to perform optimization and constraint derivation in an efficient computer-aided, but human controlled, process.

#### I.C.1.2 Detailed TA2 Technical Rationale – Real-time Operation VErified Reconfigurator (ROVER)

New and planned mission-critical LE-CPS are distributed real-time and embedded "systems of systems" whose challenging requirements can be characterized by the following open research problems [White3][White4]:

- Multiple quality of service (QoS) properties, e.g. predictable latency/jitter/throughput, scalability, dependability, and security, must be satisfied simultaneously and often in real-time;
- Different levels of service will occur under different configurations, environmental conditions, and costs and must be handled judiciously by LE-CPS infrastructure and applications;
- The levels of service in one dimension must often be coordinated with and/or traded off against the levels of service in other dimensions to achieve the intended application and overall mission results; and
- The need for autonomous and time-critical application behavior requires flexible LE-CPS infrastructure components that can adapt robustly to dynamic changes in mission requirements and environmental conditions.

Conventional system infrastructure (e.g., operating systems, networks, databases, and middleware services) developed by researchers and existing COTS platform technologies do not meet the requirements of mission-critical LE-CPS outlined above. For example, conventional monitoring technologies, such as Copilot [CoPilot1][CoPilot2] and ACM [Dubey], perform threshold based monitoring that samples the data on process interfaces periodically and declare errors if checks fail. Thresholds in a LE-CPS, however, must be adaptive and depend upon the context in which the system is operating. Moreover, the rate of sampling itself may need to adapt depending upon the criticality of the mode of operation of the system.

To address the gap in existing research and practice, the TA2 portion of the AFACTS project will develop the *Real-time Operation VErified Reconfigurator* (ROVER). **ROVER defines a novel adaptive QoS-aware monitoring and control platform for LE-CPS**. Applications running on LE-CPS can use ROVER's services to adapt dependably in response to dynamically changing conditions. As a consequence, the LE-CPS is able to utilize available computer and network resources dependably to the highest degree possible in support of mission needs, such as effectively managing resources utilized by robots in the UTACC testbed.

To ensure the end-to-end QoS and behavioral requirements of mission-critical LE-CPS, the system infrastructure must be able to monitor the state and if required make modifications dependably, i.e., even in the face of (partial) failures and attacks. These modifications are necessary control actions that help steer the LE-CPS to remain in safe operating regions. To address this need, ROVER will provide customizable QoS-aware middleware APIs, services, and *Model-Integrated Computing* (MIC) tools. ROVER will work together with capabilities defined by TA1 and TA3 performers to provide the following capabilities to LE-CPS:

1. **Synthesizing correct monitoring and control artifacts for LE-CPS infrastructure**. ROVER formalizes QoS-related design expertise via pattern languages [Buschmann1] [Buschmann2] and MIC tools [White1][Schmidt1] to generate QoS-enabled monitor and control platform artifacts, such as lightweight statistical monitors that compare system QoS

and behaviors over a time window against measures collected dynamically at run-time [Sun1[Sun2][Sun3][Turner1][Galindo1].

2. **Assure a flexible and QoS-enabled dynamically (re)configurable monitoring and control platform**. Conventional approaches for mapping application goals onto the underlying computational substrate tightly coupled applications to a particular run-time environment. ROVER's advanced adaptive behavior performs necessary adaptations autonomously based on conditions within the LE-CPS, in the LE-CPS environment, or in LE-CPS goals defined by users. Moreover, these adaptive reconfigurations and optimizations must maintain the stability of the LE-CPS and converge rapidly [CHARIOT1][IJPHM].
3. **Manage distributed resources dynamically and dependably in large-scale LE-CPS**. Statically deploying and configuring LE-CPS manually or via *ad hoc* means is tedious, error-prone, and non-scalable, which is problematic due to the need to schedule monitoring tasks and optimally deploy them onto the underlying computation resources. ROVER therefore provides automated capabilities that enable an LE-CPS to reflectively [Schmidt2] examine the capabilities it offers within a particular run-time context and dynamically optimize those capabilities, including the monitoring tasks in real-time [White2][Shankaran1] [Shankaran2][Edmondson][Lardieri].

LE-CPS have historically been developed via *multiple technology bases*, where each LE-CPS instance brings its own networks, computers, displays, software, and people. Unfortunately, these proprietary "stovepipe" architectures tightly couple many behavioral and QoS aspects of LE-CPS, which greatly impedes their adaptability, assurability and affordability. The affordability of certain DoD systems, e.g. logistics and planning, can be enhanced by COTS. Today's DoD procurement efforts aimed at integrating COTS do not support affordability *and* assurability/adaptability for mission-critical LE-CPS. Prior efforts (1) only address initial non-recurring acquisition costs, but fail to reduce recurring software lifecycle costs, such as "COTS refresh" and subsetting LE-CPS for foreign military sales and (2) compromise adaptability and assurability due to poor QoS support in COTS software products, e.g., minor perturbations in today's COTS systems can cause massive failures that impact life and property.

A key objective of the TA2 portion of AFACTS thus focuses on moving standardization from lower-levels (such as operating systems and network protocol stacks) to higher-levels by maturing LE-CPS software technology artifacts (e.g., middleware frameworks, micro-service components, and pattern languages) developed in ROVER so that they are available for COTS acquisition/customization. This focus will substantially lower DoD total ownership costs by leveraging common technology bases so that complex LE-CPS functionality need not be re-invented repeatedly or reworked from proprietary "stovepipe" architectures that are inflexible and expensive to evolve. ROVER will dramatically simplify LE-CPS QoS-aware monitoring and control platform development, optimization, validation, and integration by allowing researchers and system integrators to develop and evolve complex LE-CPS assurably, adaptively *and* affordably by (1) standardizing COTS at the middleware layers versus at lower hardware/networks/OS layers and (2) devising optimizers, meta-programming techniques, and adaptive resource management services for LE-CPS that will enable the creation of customized standard COTS interfaces, without the penalties incurred by today's COTS implementations.

### I.C.1.3. Detailed TA3 Technical Rationale – Dynamic Assurance Inferencing System (DAIS)

In modern software engineering workflows, assurance case creation and evaluation tasks are typically performed manually and entirely at design-time. This approach is not ideal, since the effort of manually creating robust assurance cases introduces considerable overhead into the development process. Moreover, the fidelity of design-time evaluation is fundamentally constrained by the intractability of anticipating (let alone enumerating) the various combinations of actual conditions the software might encounter in the real world.

The ability to fully automate assurance case evaluation presents an intriguing methodological solution to the problems of tractability and fidelity: *delay the final evaluation of assurance cases until run-time, using measurements extracted from running software as it executes in its environment to evaluate the arguments that support the claimed properties of the application*. In doing so, tractable assumptions about the expected run-time environment can be made at design-time and sufficiently evaluated at run-time. The most ambitious goal of such an *in situ* evaluation system involves synthesizing new assurance cases not only at design-time, but when emergent (unanticipated) behaviors are encountered at run-time.

Once assurance cases have been defined and relevant evidence gathered, they must be evaluated. The most obvious technique for doing so (and least satisfactory in terms of automation) is human inspection, in which a domain expert examines the assurance case, the observations relevant to it, and its argument to determine whether its claims are upheld. Machine evaluation of assurance cases can be performed using approaches like forward logical deduction or probabilistic induction [Rushby]. Techniques based on Bayesian epistemology [Bovens][Cohen] have been used to account for the uncertainty inherent in an inductive approach, however their effectiveness is limited by the overhead of building the probability network. More promising recent techniques e.g. defeasible reasoning [Pollock] and eliminative induction [Goodenough] attempt to quantify the strength of an inductive proof by its ability to eliminate the possibility of contradicting claims.

To explore the promise of dynamic assurance case synthesis and evaluation, we propose research and development of the *Dynamic Assurance Inferencing System* (DAIS). **DAIS provides tools and methods that perform dynamic assurance evaluation and synthesis** and will address the following open research problems in the assurance case space:

- Constructing assurance cases today is a manual and labor-intensive process [Blanchette] [Denney] [Hawkins][Rhodes][Saruwatari]. Technologies are needed that reduce this process by leveraging artifacts such as Software Design Documents (SDDs) and requirements documents that are already an established part of software engineering best practices.
- Tools for evaluating the validity of assurance cases are fundamentally limited by the quality of assumptions made about the actual operating environment [Bloomfield] [Goodenough][Sullivan] [Weinstock]. Technologies that reduce this uncertainty (quantifying and accounting for any residual uncertainty that cannot be eliminated) are needed.
- Semantics related to contextual (environmental) assumptions are often poorly defined, tool-specific, or absent [Coppit][Hawkins][Weinstock]. Implicit assumptions damage the credibility of the assurance case. Semantics and frameworks that handle the ambiguity encountered in LE-CPS are needed.

The DAIS will fulfill the role of a TA3 component within the AFACTS architecture and will involve the creation of technologies and methodologies that overcome these shortcomings in con-

junction with services and tools provided by TA1 and TA3 performers. Specifically, it will provide the following capabilities for LE-CPS:

1. **Provide formalisms that augment existing assurance case terminology** with notions of the uncertainty introduced by variation in the run-time environment, making assumptions explicit that must be validated dynamically. This capability will allow degrees of assurance to be predicted in the absence of complete evidence (at design or compile time). It also enables assurance dependent upon baked-in assumptions to be validated during live software execution.
2. **Synthesize assurance cases from specification-, design-, and test-time artifacts**. This capability will involve the augmentation of manually created content with generic problem space (domain) knowledge and inferencing rules. This will greatly reduce the manual effort required to develop robust assurance cases.
3. **Evaluate assurance cases with dependencies upon conditional evidence at run-time** to ensure that the specified goals of the software system are met in its actual execution environment. This capability will improve the quality of assurance case evaluation for complex systems that operate in environments that are difficult to prove formally. There is an implicit dependency upon scalability and performance in this task since it occurs with running software.

### I.C.2 Detailed Technical Approach

The objective of AFACTS is to **combine powerful design-time risk analysis methods, techniques and tools, with a novel run-time platform, that supports decision making (e.g., data collection, design optimization, and operational risk management), essential to assuring the proper functioning of LE-CPS under realistic (i.e., harsh/uncertain) operational conditions**. These conditions may include unforeseeable events (such as obstacles or adverse weather conditions) that require a learning component enabling the LE-CPS to handle these events without compromising assurance. The learning component must also assume imperfect hardware and software are used in the LE-CPS.

The following gives an actual example used on the *Enterprise Engine* (E2) from the *Unmanned Tactical Control & Collaboration* (UTACC) program. The example is given so that subsequent detailed approach subsections can describe how this actual example would be designed and implemented with AFACTS. Recall UTACC provides control software for robots that replace a Marine in a 4-Marine fire team to become a 3 Marines and an autonomous robot performing the same maneuvers. The goal of this UTACC algorithm is to classify a current formation of a fire team based on the positions and velocities of each human member of the fire team.

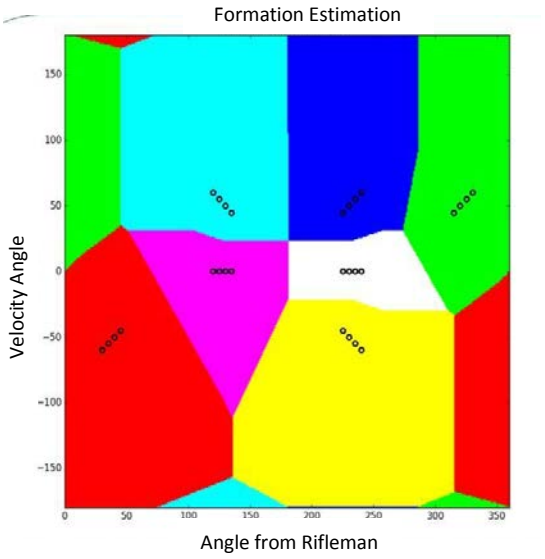



*Figure 3. Formations are Separable in 2D Angle Space*

In this example, each human fire team member carries a GPS receiver that communicates the member's position and velocity (heading and speed) to UTACC using NMEA GPS messages. UTACC maintains a position and velocity state estimate for each member by role - fire team leader, rifleman and as-



Formation Estimation

sistant automatic rifleman. Only the team members' current position and velocity state estimations are considered where we are only looking at the latest state estimates. We start with the team members' positions and headings, giving us 12 input variables (3 for position and 1 for heading multiplied by the 3 team members). Heading is formed by averaging the team members' headings.

We are only concerned with the relative positions of the team members with respect to each other, not their position in a geodetic coordinate system. We define a Cartesian coordinate system with its origin at the position of the fire team leader, since only their relative positions are important. Since the team leader and rifleman's positions are fixed in this coordination system, our variables are reduced to two: 1) the (X, Y) position of the assistant automatic rifleman, and 2) the team's heading in this coordinate system. This preprocessing reduces the classification problem to those variables (only three values), with no loss in member relative position information.

Looking at the formation in the normalized coordinates, the formations are well-discriminated based on: (1) the included angle of the positions of the assistant automatic rifleman and rifleman (X-axis), and (2) the heading angle of the team velocity vector in 2D (X, Y) plane. We can reduce the assistant automatic rifleman's position to just an angle, which leaves two input variables for estimating the formation (this assistant automatic rifleman's angle and the team heading in this coordination system). Figure 3 shows the separation of the formations in this space. Figure 4 shows the formation classification problem represented as a GSN.

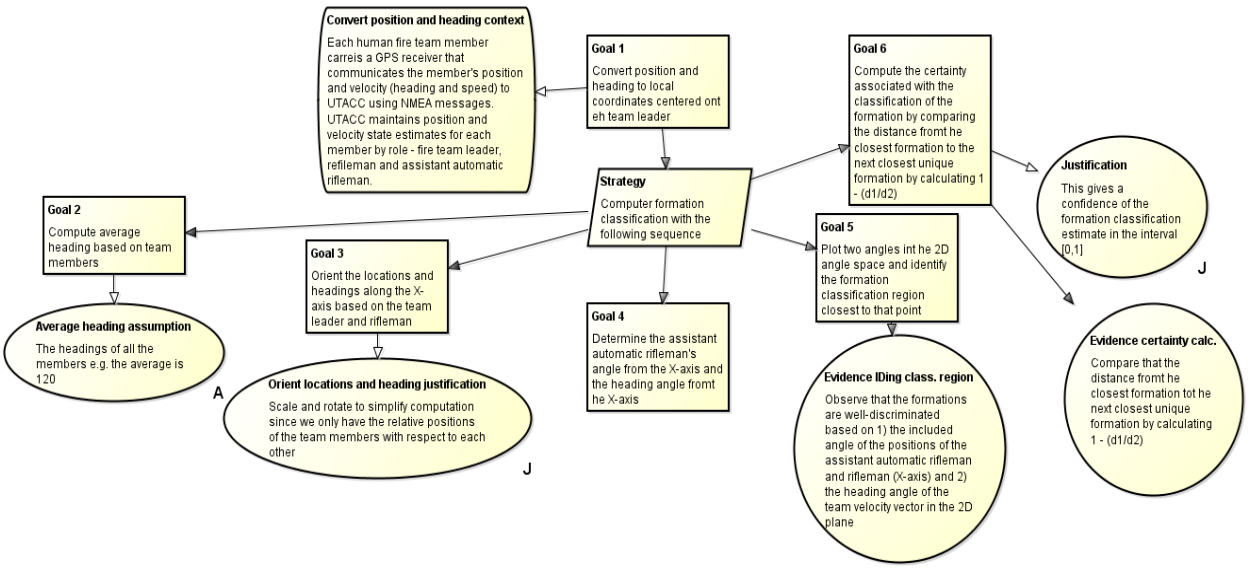



*Figure 4. GSN Representation of Formation Classification.*

### I.C.2.1 Detailed TA1 Technical Approach – MODel intEgrated fRamework for autoNomous, hIgh aSsurancE Design (MODERNISED)

#### *I.C.2.1.1 Overview*

Assured Autonomy has two primary requirements, viz. to:

1. Develop and integrate tools for design and verification of learning enabled systems by generating implementations for these systems and producing qualified evidence regarding safety and correctness of the design of LE-CPSs.

2. Develop multi-domain modeling formalisms, abstractions and DSMLs for the representation of learning-enabled components, systems and their dynamics, driving novel approaches for formal verification, simulation, and testing to generate evidence for correctness.

In prior work SNC has developed an advanced development processes and tool framework that enables the DoD to rapidly and reliably connect existing computer, communication, sensors, modeling/simulation, and weapon systems for a set of diverse E2 applications. To address requirement 1, AFACTS will extend these tools and techniques to support LE-CPS with cognitive agent technology, based on advanced formal reasoning and machine learning. To enhance the fidelity of assurance during the design phase, these ML components will be supervised during the set-up phases and will then operate unsupervised (yet monitored) at run-time, leveraging the assurance monitoring and control capabilities described in TA2 below. The supervision components will be provided via archetype-based modeling to generate reliable and secure code directly from models. To address the second requirement, we will take advantage of SysML co-simulation and parametric modeling capability to auto-generate components, ontologies and logic used for reasoning, theorem proving and model-driven simulation and test.

#### *I.C.2.1.2 Modeling and Reasoning Approach*

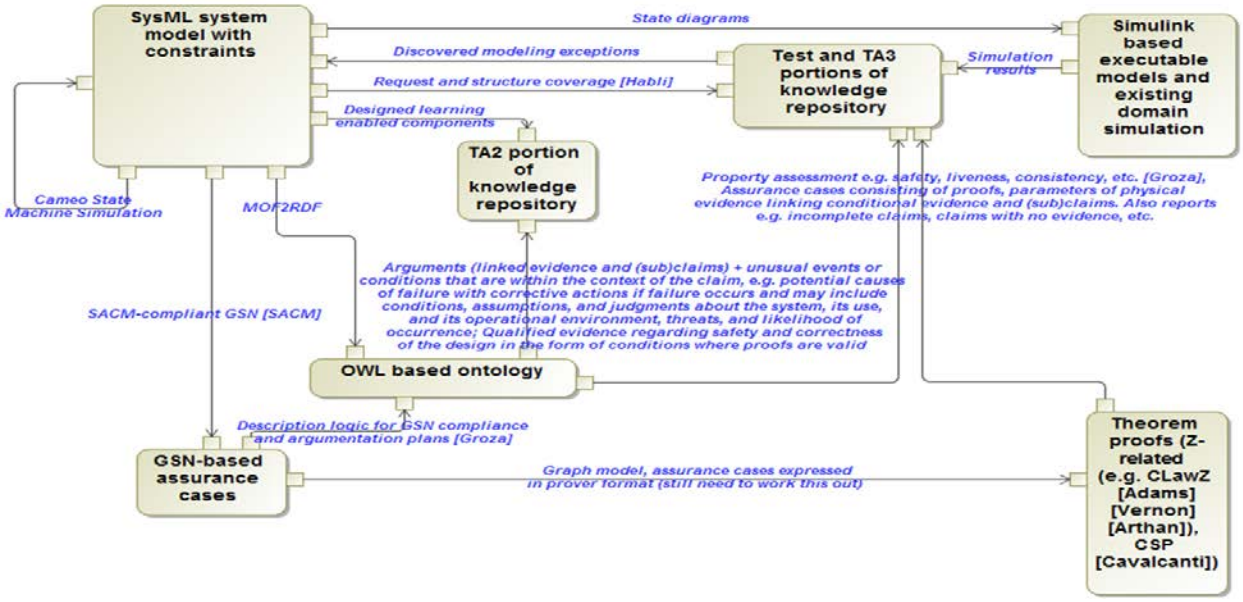
AFACTS will model LE-CPSs as collections of interrelated components represented as aggregates of subcomponents on a number of layers of abstraction. We will develop models of such systems where safety properties will be modelled as constraints. We will use formal representations of these models and formal inference engines to analyze the models. There will be two objectives to the analysis: (1) to prove that the system includes all of the required properties and (2) that none of the constraints that the system is supposed to satisfy are violated. Since these two requirements require two different reasoning strategies, we will use two kinds of formal languages to represent the requirements that are based on two different kinds of semantics. The former type of requirement is based on the open world assumption (OWA–statements that cannot be inferred are not assumed to be false) and non-unique name assumption (nUNA–an individual may be referred to by multiple names). The latter is based on the closed world assumption (CWA–what cannot be proved is automatically false) and the unique name assumption (UNA–two different names must refer to two different individuals). We will develop a validation process where both types are used collaboratively to uncover all possible constraint violations.

Reasoning over models is crucial in analyzing the satisfaction of the safety properties. Reasoning may resolve some constraint violations – “apparent” constraint violations might look like violations just because the analyzer doesn’t have sufficient knowledge about the implications of the particular constraints. The analyzer might identify an issue of the lack of a specific type of weapon on a given platform. This issue can be solved by invoking the reasoner which will uncover the existence of such information due to the fact that this platform is part of a larger platform which necessarily has such a weapon system. The same kind of reasoning process may uncover violations that are “hidden”, i.e. those that are not explicitly represented in the model.

As shown in Figure 5, LE-CPS will be modeled in SysML where we can 1) simulate state machines directly (using Cameo State machine Simulator), 2) connect to detailed and executable Simulink simulation using parametric modeling to connect to functional components, 3) represent OMG compliant GSN-based assurance cases [SACM] using stereotypes for claims, evidence, assumptions, justification, context and strategy, 4) generate request and structure cover-

age [Habli] based on SysML requirements and 5) automatically transform to OWL 2 RL (the Web Ontology Language – a language with formal semantics which also includes the capability of using rules) [Backlawski] using an E2-proven, OMG-compliant [MOF2RDF] technique. The OWL-represented models will be subject to automatic inference that will result in the derivation of implicit relationships among the model elements. While OWL can be used to express some of the constraints, OWL is not designed for this purpose. Its semantics is based on OWA and nU-NA. Consequently, it may lead to some conclusions that may not be justifiable in the real world, but will be resolved by the constraints validation engine.

*Figure 5. Reduces Risk via Designs Supporting Multi-level Modeling, Verification, Abstraction & Languages*




AFACTS will use the automatic inferencing-capable OWL RL together with the Shapes Constraint Language (SHACL), a new recommended standard by the W3C, to express safety properties that the systems should not violate and assumptions about the environment. The GSN assurance cases are transformed to graph models in SHACL, but will also, in order to increase the robustness of our approach, be expressed in theorem prover format e.g. CLawZ [Adams][Vernon] [R. Arthan] with CSP [Cavalcanti].

The models developed in TA1 will have associated safety and assurance properties that the modeled LE-CPS are expected to satisfy along with the evidence for each of the claims or subclaims. In particular, properties that need to be satisfied by the learning component will be represented explicitly. Formal descriptions will be derived from the design models in ontological form, as SHACL constraints, and as linear temporal logic. These precise descriptions will then be input for reasoning, theorem proving, and model-driven simulation and testing. The proofs for the claims will be based on assumed ground (unconditional) evidence that will be captured by the presumptions in the model. The analysis of the proof traces and reasoning engine conclusions will provide steps (conditional evidence) that will need to be verified in TA3.

This methodology will yield a rich deployable solution that contains all assurance case and rules information needed to drive autonomous AFACTS run-time implementations. The ontology-based archetype modeling approach narrows the formally verified behavioral AFACTS model

down to a mission-specific profile by applying constraints on the run-time system. These constraints are flexible and may be altered by an on-board learning component to compensate for unforeseen run-time events without compromising the run-time system's assurance state.

### *I.C.2.1.3 MODERNISED Approach Applied to Formation Classification Example*


*Figure 6. A Top-level GSN Ontology with Property Names*

Continuing with the formation classification example from the introduction of the *Detailed Approach Section* (§I.C.2), the GSN standard will be formalized as an ontology encoded in OWL (a formal language standardized by W3C). Figure 6 shows a top-level view of such an ontology, which will be used to (1) describe particular scenarios and systems, (2) specify assurance case requirements and (3) conduct formal analysis of assurance cases in specific scenarios.

As shown in Figure 6, the focus of this representation are *Goals*, aka. *claims*. Goals are connected to other goals (subgoals), which *support* or are *supportedBy* other goals. Additionally, any inherences made by the formal system will be documented as instances of the *hasInference* relation. Inference will be directed by *Strategies*. The formal analysis tool will establish relationships between particular goals by finding *Solutions* that support them, while those solutions will be dependent on some *Evidence*.

*Table 4. Properties from the Domain Ontology*

| Property | Domain | Range |
|---|---|---|
| subClassOf | Formation | Context |
| subClassOf | Wedge, Column, Skirmishers, Echelon | Formation |
| subClassOf | Heading, Angle | State |
| subClassOf | FireTeamLead, Rifleman, … | Role |
| subClassOf | Marine, Robot | TeamMember |
| hasFormation | Team | Formation |
| memberOf | TeamMember | Team |
| hasRole | TeamMember | Role |
| hasState | Team, TeamMember | State |

The whole argumentation will be performed in a specific context (represented here as the class Context). While GSN is generic, the Context class is a connection to a specific application domain. To describe contexts, we will need a domain-specific ontology. Figure 7 shows a small portion of such an ontology for our running example – classification of formations of a 3 marines plus one robot team. This figure shows just the classes from this small ontology, whose properties are listed in Table 4. The structure of the requirements for an assurance case for this scenario will make use of both the GSN ontology and the domain-specific ontology shown in Figure 7. The structure of the goals, along with the rest of the necessary specifications, will follow the argumentation structure shown in Figure 4.

Various scenarios will be developed for the purpose of formal analysis, simulation and experimentation. The inference engine will be used to derive conclusions regarding (in this case)

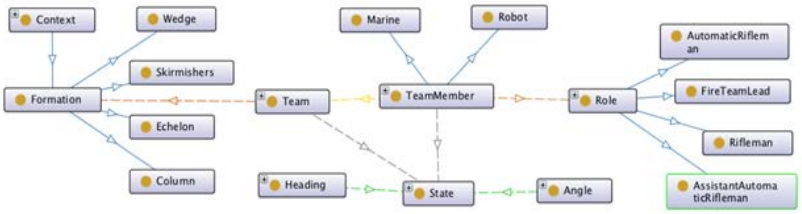


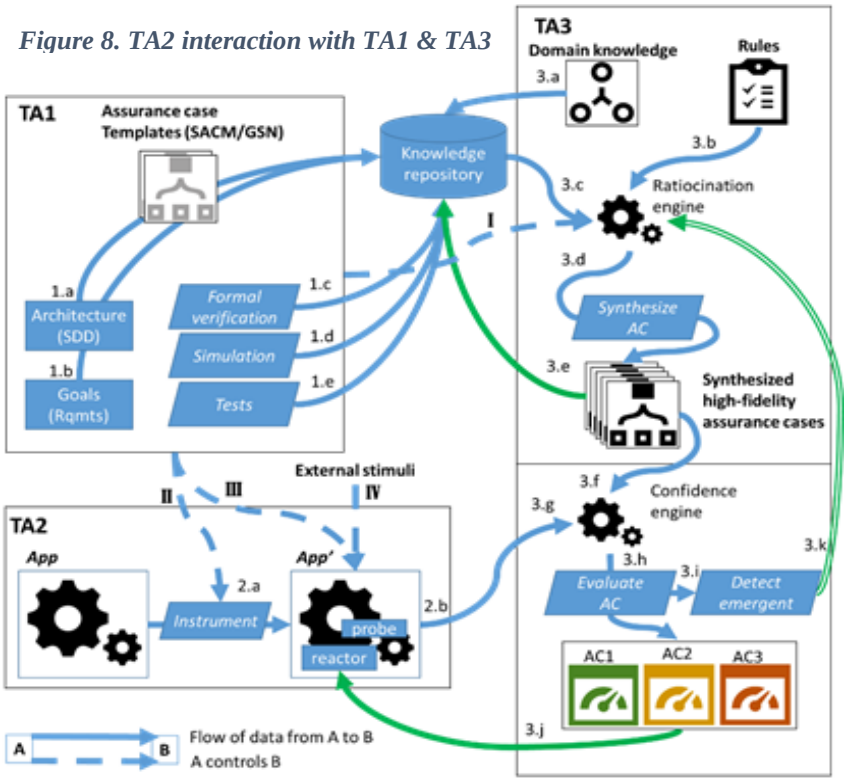

*Figure 7. A Simple Domain Ontology with Property Names*

the classification of the formation of the team. These conclusions will be documented using OWL language and stored in the Knowledge Repository in the form of RDF triples – one of the ways to serialize OWL facts. The OWL inference engine will be used to derive classifications, while the traces of the inference will be used to form solutions and evidence.

Since OWL is based on the OWA, its inferences will be "conservative", i.e., OWL will not pronounce that a specific formation is not a column formation because there is a lack of full supporting information, i.e., OWA does not infer that something is false based on the lack of evidence that it is true. To address this issue, the Shapes Constraints Language (SHACL) is able to explicitly represent constraints that need to be satisfied for something to be pronounced true. A SHACL engine will thus supplement the OWL/ontology base reasoning. It will be able to infer negative conclusions. For instance, if we specify that in order to infer that a specific configuration of team members (e.g., alignment in two parallel lines) is necessary for the column formation and this is not the case, then SHACL will identify the violation of such constraints and thus will support precise reasoning.

We have described multi-domain modeling formalisms, abstractions and DSMLs for the representation of components used to derive assurance cases to TA3. TA3 will use property assessment e.g. safety, liveness, consistency, etc. [Groza], assurance cases consisting of proofs parameters of physical evidence linking conditional evidence and (sub)claims to perform finer-grained, design time assurance case checks. Discovered modeling exceptions will be fed back to our original SysML models for refinement.

*Figure 8. TA2 interaction with TA1 & TA3*




AFACTS will also provide a two-level *dynamic Bayesian network* (DBN) to enable higher-fidelity risk analysis during design phases. The higher level DBN considers the coupling across LE-CPS subsystems, whereas the lower-level DBN considers the interactions within each subsystem. The structure of the lower-level DBN depends on the underlying subsystem architecture, e.g., synchronous or asynchronous. This DBN framework can be used as a surrogate model to estimate the performance of all components when they are integrated into an LE-CPS. Surrogate modeling is useful to compare alternative design configurations quickly [Dubey2]. TA1 will provide all above tools and framework to TA4 winners.

### I.C.2.2. Detailed TA2 Technical Rationale – Real-time Operation VErified Reconfigurator (ROVER)

#### I.C.2.2.1 Overview

The *Real-time Operation VErified Reconfigurator* (ROVER) is a quality-of-service-aware assurance monitoring and control platform for *Learning-Enabled Cyber-Physical Systems* (LE-CPS). It will play a key role in the TA2 portion of the AFACTS project as shown by the TA2 box in Figure 8. This section describes the integration of ROVER with TA1 and TA3 performers in AFACTS and explains how ROVER will meet the technical goals identified earlier.

Figure 8 depicts the TA2 interaction with TA1 and TA3 at a high level. In the figure, a hypothetical flow of data from A to B is depicted by a solid directed arrow from A to B. Flows of control (A drives B) are denoted by dashed directed arrows from A to B. Informational nodes are rectangular whereas process nodes are parallelograms. Figure 8 will be referenced extensively within this section. ROVER's capabilities will be exposed to the other TAs via a set of QoS-aware middleware APIs, services and MIC tools amenable to scalable deployment in a cloud computing environment. These services will be accessed via RESTful APIs.

#### I.C.2.3.2 ROVER Internal Operation

ROVER will contain a goal-based QoS-aware monitoring and control platform that continuously and dependably checks the overall correctness of component execution within an LE-CPS to enable:

- *Reactive corrections*, e.g., in response to violations of a confidence case that have already occurred, such as UTACC being unable to classify the fire team's formation, and
- *Proactive correc*tions, e.g., for expected violations of a confidence case based on measured degradation of QoS, such as increases in the latency that position data is received regarding the squad members, as well as
  measures of the health of confidence cases dependent upon evidence observed and emitted by the TA3 Confidence Engine (CE), such as very low confidence in UTACC's classification of the fire team's formation.

ROVER will enable the monitoring and control of guards that ensure the QoS, structural and behavioral variability of individual and aggregate components specified by models provided by TA1 performers remain within the expected statistical bounds. These guards will be used to ensure safety, e.g. preventing UTACC from making decisions that could injure squad members when it is unsure of their positions. The functionality performed by ROVER in TA2 will provide the capabilities described below.

##### I.C.2.3.2.1 Ingesting TA1 Models into ROVER Monitoring and Control Services

TA1 will provide the specifications of the goals and the expected distributions of sensor values against the control actions that should be observed by ROVER services at run-time, e.g. the latency and accuracy of positional data and formation classifications. The goal of the TA2 MIC-based tools is to convert these high-level component specifications described with respect to system sensor values into statistical tests that run as the system is operational, e.g. sampling a subset of position updates to ensure timeliness without adversely impacting timeliness via the sampling.

To achieve this goal, ROVER MIC-based tools will map system models defined by the TA1 team via high-level formal documents specifying requirements, goals, and component implementation artifacts via the *Goal Structuring Notation* (GSN). Inputs from these models (shown as flows II and III in Figure 8) will define key component properties, such as QoS bounds and requirements for distribution and replication, via formats that can be ingested and processed by the

QoS-aware monitoring and control platform provided by ROVER (shown within the TA2 portion of Figure 8). For example, these goals will specify that UTACC needs to maintain high accuracy understanding of relative squad positions and how that translates into raw inputs from GPS, inertial, and other sensor measurements.

#### *I.C.2.3.2.2 Lightweight Statistical Monitors (LSM)*

ROVER's instrumentation capabilities (shown within the TA2 portion of Figure 8) will use the inputs from TA1 outlined above to automatically synthesize *Lightweight Statistical Monitors* (LSM). These monitors will compare system QoS and behaviors over a time window against several sources collected dynamically at run-time, including:

- Metrics collected internally over a time period that compare the observed sensor values and the control actions, viz. ROVER can determine that the autonomous system is moving at a rate which is greater than the 90% confidence interval prescribed by TA1 when the squad is grouped closely together. Such probabilistic correlations can be learned at design-time using Gaussian Mixture Models and then computed at run-time analytically. Any violations identified via specifications processed by ROVER's *Alert Management Service* (AMS) (described in §I.C.2.3.2.4) will be then sent to TA3 (see blue arrow connecting 2.b and 3.g in Figure 8).
- Measures of the validity of evidence—known as "confidence cases"—emitted by TA3 (shown by the green arrow labeled 3.j from TA3 to TA2 in Figure 8) to convey information that will enable TA2's reactive correction engine to address the underlying source of problems. The TA3 Confidence Engine can inject probes into components within running application managed by TA2 (again shown by the blue arrow connecting 2.b and 3.g in Figure 8).

ROVER partitions LE-CPS into the following two classes of systems: (1) t*raining-based systems*, which are trained at design time using a training dataset and then validated with a test dataset, and (2) l*earning-based systems*, which are trained during the run-time operation of an LE-CPS. In either case, the whole sample space of sensor data describes the context that the ACPS has observed before (either in training or during operation). These two classes don't form all possible operational contexts. The goal of ROVER's run-time assurance system is twofold: (1) find the probability that the currently observed context has been seen before and (2) find if the current actual actions fit well within the distributions seen in the observed context. An operational context typically comprises of several inputs. At design-time, ROVER will compute the sensitivity of each input in the context to the classification result. These sensitivity results will then guide the choice of additional training and testing contexts.

The first step when ROVER in used for safety-critical operations is to check if a new context has been experienced before, which requires storing all training/testing contexts appropriately (e.g., storing images as matrices). The set of current sensor values will be compared periodically against the stored distributions. If the set of current sensor values fall below a given threshold this indicates the context has not been observed before. If the new context is observed longer than a set time then the LSM can raise an alert and transfer the LE-CPS to a simplex controller that provides a higher level of assurance. The set time have an upper bound determined based on the dynamic behavior of the system.

If a context has been observed before, the LSM use the Bayesian network created at design time for each component to predict the distribution of the output values and then compare them with the observed data. Due to uncertainty, this comparison is based on a probability distribution,

while a test output is a point-value. Thereafter, the LSM periodically computes the confidence in the component operation by checking the probability distribution of the observed error distribution (comparison of observed distribution with expected distribution) and if the confidence is below a threshold the LSM can raise an alert via ROVER's *Alert Management Service* (AMS) (described in §I.C.2.3.2.4).

To make our LSM approach computationally efficient, we will investigate the use of probabilistic models, such as *Gaussian Mixture Models* (GMM) and *Gaussian Copulas* (GC), which can be analytically analyzed (instead of a Bayesian network). The dependence between the components can be modeled as a set of correlations (linear functions) as opposed to complicated relationships that are used in a Bayesian networks. Our LSM can also compare error distribution for the system level goal values. The use of a GMM and a GC represent a lower fidelity abstraction compared to a Bayesian network. Such low fidelity abstractions can be used during run-time risk analysis where time is critical. For performance monitoring, we propose to investigate dynamics variations of GMM and GC. GMM and GC typically consider continuous variables. We can therefore build several GMM models corresponding to each discrete state of the system.

#### *I.C.2.3.2.3 Adaptive Reconfiguration Management Service (ARMS)*

At the heart of ROVER is an *Adaptive Reconfiguration Management Service* (ARMS), which is QoS-aware middleware that performs actions which attempt to remedy problematic confidence cases by adjusting and/or adaptively reconfiguring relevant LE-CPS resources and properties e.g.

- *QoS levels*, e.g., by adjusting the priorities of end-to-end strings of operational components that are logically and/or physically interdependent,
- *Structural relationships*, e.g., creating replicas to manage faults and balance load, and
- *Dynamic behaviors*, e.g., replacing defective or non-optimal implementations with alternatives that are better suited for the context in which they execute.

ROVER's ARMS middleware will schedule the *Lightweight Statistical Monitors* (LSM) (see §.C.2.3.2.2) in parallel with executing component tasks by running monitors during available slack time. If the end-to-end task schedule is well-defined (e.g., in a time-triggered LE-CPS) these slack times are known a priori, so ARMS can generate a monitor schedule that does not affect real-time task execution. Each LSM will aggregate component inputs and outputs, estimating the QoS using either a physics-based model or a test-based model created in TA1 from previously observed data relationships between component inputs, system sensors and outputs. As discussed in §I.C.2.3.2.2, our LSM approach is based on Gaussian mixture models that can enable analytical inference at run-time and be trained based on the studies and simulation in TA1.

ROVER's ARMS middleware will also compose component QoS variations to estimate the distribution of QoS of the overall LE-CPS goals. A surrogate model built using the design assurance tools created in TA1 will help in this estimation. Run-time probes deployed by TA3's *Confidence Engine* will then perform statistical tests against the observed system-level distributions and emit confidence cases (shown by the green arrow labeled 3.j from TA3 to TA2 in Figure 8) to TA2. ROVER's ARMS middleware will combine these confidence cases with its fault propagation model. Deviations from the models will be conveyed back to TA3 by ROVER's *Alert Management Service* (AMS), described next in §I.C.2.3.2.4.

#### *I.C.2.3.2.4 Alert Management Service (AMS)*

When the significant statistical deviations are detected by the *Lightweight Statistical Monitors* (LSM) described in §I.C.2.3.2.2. ROVER's AMS is used to propagate these alerts to the appropriate handlers, such as the TA3 *Confidence Engine* (shown by the blue arrow connecting 2.b and 3.g in Figure 8). These alerts are propagated to TA3 in a reliable and timely manner so the dynamic assurance models managed by TA3 can help to identify remediation for emergent behaviors that were not anticipated by TA1 at design-time.

For example, after being notified by ROVER's AMS, TA3's *Confidence Engine* will work in conjunction with ROVER's ARMS middleware to identify, isolate, and attempt to remedy problems at run-time by adjusting relevant platform resources and properties, such as the QoS levels, structural relationships, and dynamic behaviors discussed in §I.C.2.3.2.3.

### I.C.2.3. Detailed TA3 Technical Approach – Dynamic Assurance Inferencing System (DAIS)

DAIS will fulfill the role of a TA3 performer in the AFACTS project. In this section, we outline the integration of DAIS with other team performers and describe how the technical goals identified earlier will be met.

#### *I.C.2.3.1 Overview*

Architecturally, DAIS functionality will be exposed as a set of web services amenable to scalable deployment in a cloud computing environment. These services will be accessed via RESTful APIs. Artifacts consumed and emitted during an analysis workflow that do not significantly affect run-time performance will be stored in a central knowledge repository (triple store) shared between performers. Performance critical run-time data (e.g., evidence) will be passed directly through an appropriate TA3 REST API. Figure 8 (shown in §I.C.2.2.1) depicts the TA3 interaction with TA1 and TA2 at a high level and will be referenced heavily within this section.

Internally, DAIS operations are divided into design and run-time behaviors. In Figure 8, those dataflows and processes in the upper portion of the TA3 box represent design-time operations where those in the lower portion correspond to run-time operations. The SDD (1.a) and requirements document (1.b) are the primary inputs to TA1 in the form of high-level formal documents specifying requirements and goals. One of our initial assumptions is that these documents are formalized as low-fidelity claims (or goals) using an OMG-compliant form of GSN [SACM].

TA1 performs formal verification (1.c, which emits proofs that take the form of unconditional evidence), simulation (1.d, which emits observations about the software's sensitivity to its environment and form the basis for the inference of conditional evidence), and testing (1.e, which emits observations about the software's nominal behavior during testing) using 1.a and 1.b as inputs. The results of these operations are converted into a normalized vocabulary and inserted into the knowledge repository (triple store). Upon completion of these tasks, TA1 initiates control flow ***I***, which is the trigger for the TA3 design-time operations that ultimately result in the synthesis of new assurance cases.

The output dataflows of TA3 are shown with green arrows in Figure 8. During design-time, DAIS emits synthesized assurance cases, which are stored in the knowledge repository. At run-time, these assurance cases are evaluated and a structure representing ratiocination that supports or rejects the claims contained within each AC is provided to TA2. These evaluated ACs serve as stimuli for reactive components injected into the original application by TA2 ensuring software

goals software are met at run-time. TA2 understands the reactive mechanics whereas TA3 provides enough detail to achieve meaningful reactive behaviors.

There is a TA3 recurrence between design- and run-time shown in Figure 8, highlighted by the double-lined green dataflow connecting the *detect emergent* process within TA3's run-time system to the ratiocination engine design time component. This recurrence indicates the discovery of some unforeseen emergent behavior relevant to the assurance of the system, which triggers the emission of new or revised assurance cases during live software execution. This ability to account for emergent behaviors is a key discriminator of AFACTS and the means by which it is achieved is described below. Also shown in Figure 8 for completeness are control flows ***II*** (TA1 triggers the TA2 workflow at design time), ***III*** (TA1 performs testing of the running software at design time), and ***IV*** (the software encounters some external stimuli as it executes in its deployed environment at run time).

#### *I.C.2.3.2 DAIS Internal Operation*

DAIS functionality is abstracted into two core internal components that correspond to design/run time operations in the following: a *Ratiocination Engine* performs design time synthesis of assurance cases, and a *confidence engine* performs run-time evaluation of those assurance cases.

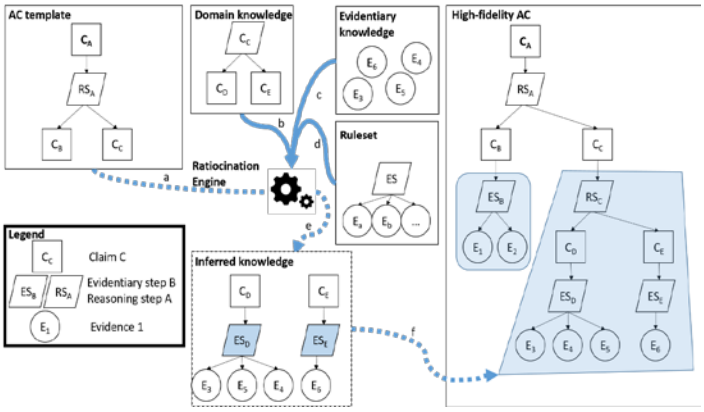



*Figure 9. Ratiocination Example*

#### *I.C.2.3.2.1 Ratiocination Engine*

The *Ratiocination Engine* (RE) synthesizes high fidelity assurance cases from low-fidelity, claim-oriented GSN assurance case templates provided by TA1. The RE operates by applying domain knowledge (3.a in the Figure 8) to rules (3.b) describing various proving strategies (e.g., deduction, forward induction, eliminative induction). These inputs are applied to evidentiary knowledge gleaned from TA1 analytics (1.c, 1.d, 1.e) retrieved from the knowledge repository (3.c) by the RE, resulting in synthesis and publication of fine-grained assurance cases (3.d, 3.e).

A design goal and risk mitigation strategy of the RE system will be to enable plug-and-play proving strategy rulesets (e.g., deductive, forward inductive, eliminative inductive, defeasible) on a per-AC basis. The RE doesn't rely upon any single framework for AC creation and evaluation, which accounts for the varied applicability of different techniques under different circumstances (deductive reasoning works well for certain mathematically-oriented claim types where more abstract claims with incomplete evidence require inductive proof techniques).

To show how the RE will operate, we will reference our UTACC fire team motivating example. Recall, UTACC's formation detection algorithm is heavily dependent upon the retrieval of accurate location information (typically provided by GPS or inertial sensors). The following three hypothetical claims describe this dependency (the inputs are provided in GSN and OWL):

$C_A$. `The system operates in a safe manner`
$C_B$. `Distance not too close to DMZ (distance > 1km)`
$C_C$. `Location error of GPS sensor not too large (error < 10m SEP95)`

These claims represent the granularity of assertions typically found in requirements and design

documents. They could be combined into a hypothetical high-level deductive argument supporting proposition A as follows: $RS_A$: ($C_B \wedge C_C$) → A. This template is depicted as "AC template" at the top left in Figure 9, with the deductive implication depicted as parallelogram "$RS_A$" (reasoning step A). The figure illustrates operation of the RE on the template (edge a in the figure) to produce a high-fidelity assurance case (f) using domain knowledge (b), evidentiary knowledge (c), and a proof ruleset (d) via inferred knowledge (e) and is outlined below.

The AC synthesis process begins with evidence binding to answer the questions (1) what unconditional evidence provided by TA1 is indicative of claim X? (2) what conditional evidence provided by TA2 is indicative of claim X? and (3) what is the uncertainty of these mappings given domain knowledge? For example, claim $C_B$ above (`distance from DMZ > 1km`) is determined by the predicate `distanceToDMZ() < 1km`, which manifests as evidentiary step $ES_B$ dependent upon evidentiary parameters $E_1$ (current location of the sensor) and $E_2$ (current DMZ definition). The uncertainty of this evidentiary step is equal to the prior uncertainty of measurements $E_1$ and $E_2$, constrained at design time by TA1 formalisms in the case of unconditional evidence or at run-time by TA2 in the case of measured conditional evidence (note that it is uncertainty of the measurement that must be constrained during evidence binding, not the value itself).

The result of evidence binding is the creation of one or more evidentiary steps (depicted as $ES_{B,D,E}$ in the figure above) for naked claims (e.g., $C_B$, $C_C$). Evidentiary steps [Rushby] apply rules from the reasoning ruleset to one or more given instances of evidence (conditional or not) to dynamically support or reject some claim given evidence with some degree of certainty. For example, a rule from the deductive framework for ESB (supporting claim $C_B$ above) would depend upon the existence of $E_1$ and $E_2$ and the certainty of those measurements.

The second step in AC synthesis is claim expansion. Claim expansion attempts to recursively subdivide claims into subordinate claims or evidentiary assertions until all leaf nodes of the claim graph are evidentiary. This expansion is achieved using a combination of domain knowledge and a ruleset. For example, claim $C_C$ (location error not too large) can be expanded into subordinate claims using the following knowledge and a deductive ruleset (1) GPS signals are degraded when line of sight is occluded and (2) GPS signals can be jammed or spoofed. From this ruleset we can derive subordinate claims to $C_C$:

$C_D$. `GPS signal not occluded`
$C_E$. `GPS signal not jammed/spoofed`

Using the deductive reasoning step: $RS_C$ : ($C_D \wedge C_E$) → $C_C$. The evidence binding step is then performed again to determine evidences for $C_D$ and $C_E$. Claim $C_D$ is supported by measurements about occlusion ($E_3$: `isStructureOccluding`, $E_4$: `isTerrainOccluding`, $E_5$: `isFoliageOccluding`). Claim $C_E$ is supported by a single measurement about jamming/spoofing ($E_6$: `signalIntegrity`). The process is then completed because all leaf nodes are evidentiary. The high-fidelity AC that results from this hypothetical workflow is shown at right in Figure 9, with the inferred knowledge relevant to claim $C_C$ explicitly shown in the dashed "Inferred knowledge" box and the resultant expanded claims shown highlighted in blue.

#### *I.C.2.3.2.2 Confidence Engine*

The Confidence Engine (CE) produces *confidence cases* from assurance cases. A confidence case [Goodenough] consists of (1) an assurance case, (2) confidence values for each claim in the

assurance case, (3) a rationale for the assigned values based on observed evidence. As in Figure 9, confidence cases are used to stimulate reactive mechanics injected into the application by TA2. The CE operates by ingesting assurance cases synthesized by the RE (3.f in the figure) and metrics published by probes injected into the running application by TA2 (2.b/3.g). The CE then evaluates the ACs and emits confidence cases (3.j) to reactive logic injected by TA2.

The confidence cases constructed by the CE and provided to TA2 must carry an actionable rationale that enables TA2 to address the underlying source of problems. This rationale must align with remediation strategies enumerated at design time of which TA2 is aware. A high-level claim about safety defined in an AC may be violated because GPS accuracy guarantees are voided by observed evidence indicating terrain occlusion. In this case TA2 should revert to an inertial guidance strategy. Contrast this with a safety violation caused by proximity to a DMZ where the corrective action would be to move immediately away from the DMZ.

The interplay between TA2 and TA3 is more complex than hinted above in Figure 8. It would be wasteful (and significantly impact performance) for TA2 to continuously report all its gathered metrics to TA3 along the critical execution path. TA2 implements a more scalable strategy via an off-thread filter-and-alert mechanism that pushes alerts to a TA3 API infrequently unless conditions indicative of problems are observed (in which case an alert is pushed immediately to TA3). TA3 can selectively requery TA2 retrieving metrics relevant to potential problems without relying on TA2 to publish a full dump of all metrics (many irrelevant at a given point in time).

The DAIS goal is to allow plug-and-play rule engines. The confidence engine is aware of the logical rule framework under which each claim in an assurance case is asserted. If a claim is asserted using deductive rules, it will be evaluated deductively with absolute confidence whereas inductive rules will be applied with confidence values based upon defeasible metrics [Pollock][Weinstock] or partial induction [Bloomfield] as appropriate.

One of the most ambitious DAIS goals is the ability to identify emergent behaviors unforeseen during design-time and then synthesize new assurance cases to deal with these conditions. There are no evidentiary or reasoning steps, in the previous UTACC scenario, that relate GPS signal integrity to weather conditions. Suppose that weather conditions (specifically cloud cover) are known to attenuate GPS signals. If, during execution, TA3 received metrics indicate a loss of GPS signal without also observing evidence supporting occlusion or jamming, this would trigger abductive logic [Hobbs] that would attempt to account for the observed behavior (recall that abduction is a process of moving from observation to hypothesis).

When a suitable explanatory hypothesis is discovered (cloud cover), it is woven back into claims to update existing assurance cases (to check future assurance cases). The normalized uncertainty value carried by an abduced relationship will reflect (1) the existence of any corroborative evidence (e.g., sensor data indicating cloud cover) and (2) the number of abduced hypotheses that fit some observed phenomenon (less possible hypotheses indicate less uncertainty). If no suitable hypothesis can be abduced (after applying an uncertainty threshold), the event can still be reported to human operators, examining the event to determine for assurance case update.

The *Risk Management Section* (§I.D.3) combines technical risk identification with risk mitigation and impact.

# I.D. Management Plan

## I.D.1 Team Organization

The AFACTS team is composed of leading industry and academic partners specializing in autonomous control systems research, development, test, and evaluation. SNC is the Prime Contractor and has partnered with Securboration, Vanderbilt University, and Northeastern University as subcontractors. This arrangement is shown below in Figure 10, along with project responsibilities for each teammate.

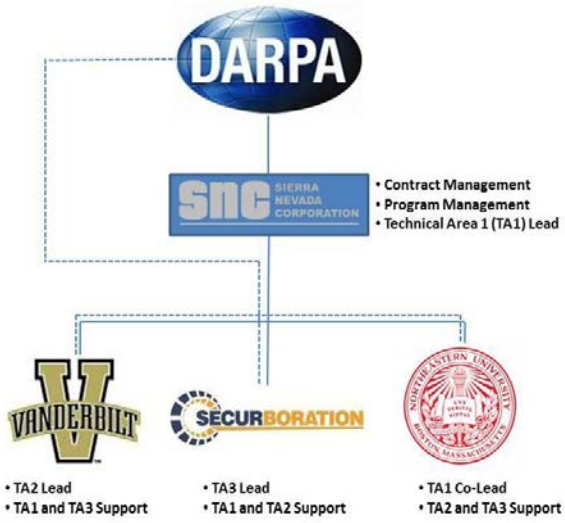



*Figure 10. AFACTS Team Organization*

Under the AFACTS program, SNC has a contractual relationship with DARPA and acts as the primary point of contact for communication between the Government and contractor team. Each AFACTS team member has informal relationships with DARPA, however, to ensure complete, clear, and continuous communication throughout program execution.

Dr. Jeff Smith is the PI for the project. Dr. Smith has worked with all AFACTS partners previously, including collaboration on several DARPA-sponsored efforts. Additional information on key personnel, unique capabilities, planned effort towards this project, and responsibilities is included in the *Personnel, Qualifications, and Commitments* section below.

### I.D.2 Program Management

The AFACTS team expertly manages the technical delivery of an innovative and comprehensive assured autonomy solution while ensuring objectives for cost, schedule, and quality are met or exceeded. As Prime, SNC tailors program management processes based on industry standards, such as Defense Acquisition University (DAU) and Project Management Body of Knowledge (PMBOK) to fit the unique needs of the program and stakeholders. The program management team performs multiple iterations of the design, development, and test process to implement a system solution with greater fidelity to DARPA's needs based on continuous communication with the DARPA Program Manager. The AFACTS team establishes specific processes for reporting, risk management, and configuration management to provide DARPA insight and confidence into the day-to-day program management.

### I.D.3 Risk Management

Table 5 summarizes the key technical risks faced by the TAs in this program and describes how AFACTS will address and/or mitigate these risks.

*Table 5. AFACTS Addresses Technical Risks Through Mitigation Options & Decision Points in the Project Plan*

| Risk (by TA) | Impact | Addressed/Mitigated by |
|---|---|---|
| (**TA1**) Difficulty formally analyzing LE component nonlinearities as model reachability computations are often intractable | *Non-robust designs* | Multiple validation approaches with advances in dependence learning, uncertainty aggregation & QoS management developed with TA4 and team-provided autonomous platforms |
| (**TA1**) Resiliently monitoring and enforcing QoS (QoS) constraints of LE-CPS computations that operate dependably across failures & attacks | *Faulty identification of failure causes* | Develop framework for probabilistic QoS assurance that combines advances in design-time analysis and run-time platforms to safely & dependably handle environmental condition variations |
| (**TA2/TA**3) Potentially overwhelming amount of metrics related to evidence passed from TA2 to TA3 | *Low performance & QoS* | Filter & re-query approach in which TA2 issues a small number of alerts & TA3 solicits additional metrics as-needed |
| (**TA3**) Hard to synthesize ACs directly from engineering artifacts | *Synthesis of low quality ACs* | Meet-in-the-middle approach where coarse grained ACs are augmented with domain knowledge to produce high-fidelity ACs |
| (**TA3**) It is unclear whether one AC confidence evaluation framework is appropriate for all AC claims | *Spurious confidence values* | Generic evaluation architecture agnostic of any single confidence framework |
| (**TA3**) Emergent conditions that cannot be anticipated at design time | *Spurious confidence values* | Abductive logic to dynamically synthesize new AC constructs that account for emergent conditions |

## I.E Personnel, Qualifications, and Commitments

The AFACTS team is led by acknowledged industry and academic experts in model based software, large scale software development, distributed applications, and formal methods. Table 6 expands upon the qualifications of key personnel from Sierra Nevada Corporation (SNC), Northeastern University (NU), Vanderbilt University (VU), and Securboration.

*Table 6. AFACTS Combines Top Leaders in key Model-Integrated Computing, Generative Programming, QoS-aware Middleware, and Formal Methods technologies and standards*

| | Role | PhD. Institution | Clearance |
|---|---|---|---|
| Dr. Jeff Smith | Principal Investigator (PI) technical lead on AFACTS | Northeastern U | TS/SCI |
| Dr. Smith is a SNC Chief Systems Engineer leading the E2 Programs described in this document. He was a Chief Scientist at BAE and Director of Mercury Computer Labs. He has 40 years-experience in technologies, e.g. SDR, secure network protocols, OSs, simulation, multi-sensor fusion, random optimization, formal methods, and agent and object oriented software engineering domains. He has written extensively on UML for modeling ontologies and formal methods. He is co-chair of OMG Analysis and Design Task Force and chair of OMG Software based Communication Working Group, Senior ACM Reviewer/IEEE member. PhD research on UML formalization and transformation [Smith1] [Smith2]. PI on DARPA SHARE, ANTS, PERFECT and leading role on SoSITE Programs. | | | |
| Dr. Doug Schmidt | Vanderbilt University TA2 lead | UC Irvine | TS/SCI |
| Dr. Douglas C. Schmidt is the Cornelius Vanderbilt Professor of CS at Vanderbilt University. He has published 12 books and more than 600 technical papers on a range of software-related topics, including patterns, optimization techniques, and empirical analyses of frameworks and model-driven engineering tools that facilitate the development of mission-critical middleware and mobile apps running over wireless/wired networks and embedded system interconnects. From 2010-2014 he served a member of the Air Force Scientific Advisory Board, where he served as Vice Chair of a study on Cyber Situational Awareness for Air Force mission operations. Dr. Schmidt also served on the Advisory Board for the *Future Airborne Capability Environment* (FACE) and was recently co-lead of a task area on "Published Open Interfaces and Standards" for the US Navy's *Open Systems Architecture* (OSA) initiative. From 2000 to 2003 Dr. Schmidt served as a Deputy Office Director and a Program Manager at DARPA, where he led the national research and development effort on middleware for distributed real-time and embedded (DRE) systems. He has also made significant contributions to international standardization efforts, such as the OMG Real-time CORBA and *Data Distribution Service* (DDS) specifications that are widely adopted as the basis of DoD combat systems. Dr. Schmidt has led the development of ACE, TAO, and CIAO for the past three decades. These open-source middleware frameworks and model-driven tools constitute some of the most successful examples of software R&D ever transitioned from research to industry, being widely used by thousands of companies and agencies worldwide in many domains, including national defense and security, datacom/telecom, financial services, healthcare, and online gaming. | | | |
| Manfred Koethe | SNC TA1 co-lead | U. of Karlsruhe | Secret Pending |

| | | | |
|---|---|---|---|
| Manfred Koethe is Lead Principal Engineer at SNC, leading the Model-oriented Development Environment R&D Project, which focuses on automated software assessment and on the production of secure and reliable code directly out of models. He has a deep background in object-oriented, semantic and functional modeling, federated real-time systems, Agent technology, and pattern-based formal model analysis. At OMG he is leader of the Agent Submission, co-author of the MOF and SMOF, MOF2RDF and executable UML (fUML) standards, co-chair of the UML and MOF Task forces and elected member of the OMG Architecture Board. In the past he was co-lead of the Federal Enterprise Architecture OSERA project and chief architect of the two European R&D projects PISA and VEGA | | | |
| Dr. Mitch Kokar | Northeastern University TA1 co-lead | Northeastern U | Secret |
| Mieczyslaw "Mitch" Kokar is a Professor in ECE at Northeastern University. He is an active researcher in Situational Awareness, Cognitive Radios, Information Fusion, Ontologies and Semantic Web and Self-Controlling Software. His research focuses on the formal semantics of communication among heterogeneous computer and human agents. He has authored over 200 journal/conference papers and two books. Currently he is one of the Co-PIs on the DARPA CONCERTO program. He was a Co-PI in the Autonomous Negotiating Teams (ANTS) for which the PI was Jeff Smith of BAE. He was also a PI for the DARPA Agent Markup Language (DAML), which later became Web Ontology Language – OWL. Currently, he is working on a DARPA supported STTR project that is aligned with the Radio Map program. Additionally, he is part of the BAE team on the DARPA SSPARC program. For over three years, he served in the role of advisor to Dr. Preston Marshall in relation to the DARPA programs XG, DTN and WNaN. Dr. Kokar was a PI on the AFRL project Trusted Autonomy and V&V. The main focus of this project was detection and learning of emergent behaviors. | | | |
| Lee Krause | Securboration TA3 lead | Rochester IT (RIT) | TS/SCI |
| Mr. Krause has authored over 14 approved/pending patent applications for innovations ranging from automatic code parallelization to the optimization of systems based on models. Mr. Krause currently leads the AFRL Stampede effort focused on developing tools to provide commanders with a mission centric view that improve mission assurance and mission awareness. Mr. Krause supports the BRASS effort for the BBN immortals team focused on Semantic Web mapping tools that link Java programming abstractions to semantic concepts defined in an ontology. Along with bytecode analysis tools for building semantic models of application architecture from compiled artifacts that utilize these abstractions. Mr. Krause is an accomplished PI supporting both SBIR and BAA programs for the DoD and NIH and has over 30 years of experience developing major DoD systems such as Joint STARS, the Satellite Data Handling System, and the Air Force Global Weather Systems. Mr. Krause serves on the Advisory Board for Florida Tech Computer Engineering Department and serves as an active member of Florida Tech Research Park Tech Council focused on technology transition. | | | |

As shown in Table 6, not only does our team have a long track record of success transitioning various middleware and application technologies commercial industry and defense industry base (see §II.B.1), the AFACTS team also has a long history of success transitioning technologies into broad adoption via open standards. These successful engagements with standards organizations provides yet another means for the AFACTS team to transition their work on the Assured Autonomy program from research into practice.

For example, Douglas C. Schmidt has made significant contributions to international standardization efforts of the *Object Management Group*'s (OMG) Real-time CORBA and *Data Distribution Service* (DDS) specifications that are widely adopted as the basis of DoD combat systems and the telecom and CPS domains. He also served on the Advisory Board for the *Future Airborne Capability Environment* (FACE) consortium and was recently co-lead of a task area on "Published Open Interfaces and Standards" for the US Navy's *Open Systems Architecture* (OSA) initiative.

Jeff Smith and Manfred Koethe have made significant contributions to the standardization of the OMG's Unified Modeling Language (UML) efforts. In particular, Jeff Smith is the co-chair of OMG Analysis and Design Task Force and chair of OMG Software-based Communication Working Group. Likewise, Manfred Koethe is the leader of the OMG's Agent Submission, co-author of the OMG's MOF and SMOF, MOF2RDF, and executable UML (fUML) standards. In addition, he is co-chair of the UML and MOF Task forces and elected member of the OMG Architecture Board.

Time commitments for key individuals on the AFACTS team are shown in Table 7.

*Table 7. Time Commitments for Key Individuals on the AFACTS Team*

| Key Individual | Project | Status | Hours on Project | | |
|---|---|---|---|---|---|
| | | | Phase 1 (18 Months) | Phase 2 (15 Months) | Phase 3 (15 Months) |
| Jeff Smith | Assured Autonomy | Proposed | 2241 | 1868 | 1868 |
| | Enterprise Engine (E2) | Current | 729 | 607 | 607 |
| Doug Schmidt | Assured Autonomy | Proposed | 1664 | 1144 | 1144 |
| Manfred Koethe | Assured Autonomy | Proposed | 2241 | 1868 | 1868 |
| | Enterprise Engine (E2) | Current | 729 | 607 | 607 |
| Mitch Kokar | Assured Autonomy | Proposed | 390 | 325 | 325 |
| | DARPA CONCERTO | Current | 240 | 200 | 200 |
| Lee Krause | Assured Autonomy | Proposed | 315 | 273 | 263 |
| | NIH-ISPS | Current | 500 | 200 | 0 |

# I.F Capabilities

## I.F.1 Previous Accomplishments and Related Work

Our team is unique in that it has a built-in technology transition plan for the AFACTS framework, QoS-aware monitoring and control middleware, MIC tools, and our autonomous application to a production-level set of safety-critical LE-CPS programs. This plan is summarized in Table 8 and described below.

*Table 8. The AFACTS Team Builds on a Solid Foundation of Research and Technology Transition Success*

| RSMT (Robust Software Monitoring Tool)/BRASS | Sponsor: ONR/DARPA | PoP: 2014-2018 |
|---|---|---|
| **Relevance**: Validated the ability to leverage dynamic analysis to analyze, monitor, and enforce constraints on a CPS. For AFACTS we plan to extend this body of research to constrain AcCPS to ensure safe operations.<br>**Summary**: Securboration and Vanderbilt developed RSMT. RSMT uses code-level instrumentation to gather exemplar behavior models of software during unit, integration and validation testing. When the software is deployed, its actual behavior is compared to these models to determine whether untested (and potentially dangerous) behaviors are occurring. Securboration and Vanderbilt also actively participate in the BRASS program as part of the IMMoRTALS project led by Raytheon BBN. IMMoRTALS is creating advanced program analysis, resource specification, program synthesis, and run-time techniques to manage resource-related changes in application ecosystems. | | |
| **Distributed Real-time Autonomously Guided Operations eNgine** | **Sponsor: ONR/NGA** | **PoP: 2016 -2017** |
| **Relevance**: The assurance framework in this proposal is based on one developed under the SNC Enterprise Engine (E2) program. We will extend this framework for AFACTS, as well as extend simulation and actual AcCPS to provide a basis for metrics.<br>**Summary:** DRAGON reduces UxS user/operator burden by communicating what tasks to perform and not how to perform them. DRAGON provides more accurate and rapid targeting services e.g. navigation and detection, recognition, tracking, and location of objects, increasing the standoff distances for manned platforms, and reducing the amount of manpower required to deploy ISR assets. DRAGON autonomously detects, classifies, and identifies bridges, buildings, vessels, vehicles, and people (with weapons). DRAGON also provides object of interest location, and course and speed in real-time for targeting operations. | | |
| **Unmanned Tactical Autonomous Control and Collaboration (UTACC)** | **Sponsor: Marine Corps** | **PoP: 2015-2017** |
| **Relevance:** The assurance framework in this proposal is based on one developed under the SNC Enterprise Engine (E2) program. We will also extend robot-marine maneuvering with 3 humans to perform cooperative maneuvers with multiple robots. We will extend simulation and actual systems to provide a basis for metrics.<br>**Summary:** The decentralized multi-unmanned manager enables manned-unmanned teaming under mission orders. Developed a robot and simulation that replaced one U.S. Marine in a 4-Marine fire team moving across terrain & changing formations. UTACC's AI engine assists Marines to manage multiple UxS platforms with one operator and accomplish the mission faster. UTACC provides automated mission planning, dynamic replanning as conditions dictate, unburdens operator yet keeps him in loop, one-to-many C2 of UxS platforms & Integrated Fires for improved ground battlespace awareness, increased maneuver time/space, economy of force. | | |

The *Enterprise Engine* (E2) family of programs, developed by SNC, consists of a semantic integration framework layered on top of a federated integration, communication, command, and control framework. The E2 technology has been an enabler for a multitude of projects, spanning a wide

range from safety-critical enterprise applications, as largely automating the Notice to Mariners process for the NGA, to unmanned vehicles as in the *Distributed Real-time Autonomously Guided Operations eNgine* (DRAGON), and the *Unmanned Tactical Control & Collaboration* (UTACC) programs.

Vanderbilt University (VU) has a long history of developing and transitioning software technologies for QoS-aware middleware and applications. For example, the VU team has led the development of ACE, TAO, and CHARIOT, which are widely used, open-source QoS-aware middleware frameworks that implement patterns and product-line architectures for high-performance distributed real-time and embedded (DRE) systems. These QoS-aware middleware platforms constitute some of the most successful examples of software R&D ever transitioned from research to industry, being widely used by tens of thousands of developers in thousands of companies and agencies worldwide in many domains, including national defense and security, datacom/telecom, financial services, healthcare, and online gaming. These QoS-aware middleware technologies are also commercially supported in open-source form by multiple companies, including Riverace, Object Computing Inc., Remedy, and Micro Focus.

### I.F.2 Facilities

We propose the SNC facility, called the *Multi-Agency Collaboration Environment* (MACE), at 3076 Centreville Road, Herndon, VA 20171, as the primary work location for AFACTS. The MACE was established in 2008 as a consortium of industry and government partners designed to provide innovative technical, analytical, and process solutions to support a rapidly changing threat environment. MACE activities currently focus on data and system interoperability, open source analytics, artificial intelligence and machine learning applications, and perception management. The MACE operates out of more than 40,000 square feet within a state-of-the art development environment optimized for the sharing and testing of ideas, and implementing technical solutions.

**Figure 11: Commercial Imagery Lab Facilities at the MACE**

This facility is currently leveraged by multiple federal agencies for collaborative, agile development briefings and meetings. The design of the facility was specially architected to enhance collaboration between engineers and teams working in the space and can accommodate different size teams and a variety of working environments, as shown in Figure 11. Project spaces for sensitive efforts can be accommodated based on security classification or customer requirements. The MACE is accredited up to the Top Secret SCI level by the Defense Security Service and DIA, and has space to accommodate all levels of classification to include unclassified, law enforcement sensitive, secret and top secret.

# I.G Statement of Work, Schedule and Milestones

## I.G.1 SoW

Table 9 provides the Statement of Work (SOW) per Work Breakdown Structure (WBS), by Phase, for 48 months, which includes tasks and deliverables that enable execution of program goals where key milestones are tied to program deliverables. The deliverable numbers follow the scheme outlined in the deliverables and schedule that follow the statement of work (Figure 10).

*Table 9. The AFACTS Statement of Work*

| WBS # Work Element Title | |
|---|---|
| 1 Phase 1 Assured Autonomy Integration Technology Development | |
| **1.1** System Engineering | Objective: Participate in program TIMs and review meetings, develop requirements via team interaction |
| **1.1.1** Requirements Analysis | Objective: Develop AFACTS requirements |
| Approach: Conduct a requirements analysis based on developed use cases to prioritize AFACTS models to develop. Define capabilities of toolset to be developed, interface languages, and implementation design. Support ongoing coordination with team to identify technology insertion points. | |
| Deliverables (CDRL #): 2 | Completion Criteria/Milestone: Delivery of Design Package by PI 2 |
| Supporting Orgs: SNC, VU, S, NU | Duration (months): 9 |
| **1.1.2** Technical Interactions and Reporting | Objective: Participate in technical interchange and status review meetings |
| Approach: Prepare for and participate in program kickoff, programmatic and technical reviews. Support weekly telephonic conferences to discuss the status of the work effort, key issues, and key emerging results. Participate in technical exchange meetings with internal and government team. | |
| Deliverables (CDRL #) 1,2,3,4,5,6,7 | Completion Criteria/Milestone: Reporting completed Phase 1 demo |
| Supporting Orgs: SNC, VU, S, NU | Duration (months): 18 |
| **1.2** TA1 - Research and Prototype Tools: Tool Development, model translation, and process development | Objective: Develop tools for design and verification of learning enabled systems as well as multi-domain modeling formalisms, abstractions and languages for the representation of learning-enabled components/systems |
| **1.2.1** Build tool framework | Objective: Build tool framework to support cognitive agents, archetype modeling and SACM compliant models. |
| Approach: Extend existing E2 Framework tools and techniques to support AcCPS with cognitive agent technology, compliant with SACM models, based on advanced formal reasoning and machine learning. | |
| Deliverables (CDRL #) 2,3 | Completion Criteria/Milestone: Toolset ready for delivery and evaluation by Phase 1 demo |
| Supporting Orgs: SNC,NU | Duration (months): 18 |
| **1.2.2** Translate SysML models | Objective: Translate SysML models to code, ontologies, simulation and formal components used for pre-test before passing to TA2. |
| Approach: Build translations needed to support multi-domain modeling formalisms, abstractions and languages for the representation of learning-enabled components/systems. | |
| Deliverables (CDRL #) 1,2,3 | Completion Criteria/Milestone: Toolset ready for delivery and evaluation by Phase 1 demo |
| Supporting Orgs: SNC,NU | Duration (months): 18 |
| **1.2.3** Develop validation, reasoning and simulation processes | Objective: Develop validation, reasoning and simulation processes to report model constraint violations to testing and TA3 processes. |
| Approach: Develop a validation process to uncover all possible constraint violations, co-simulation process to pre-test state machines at the model and simulation levels and formal processes to perform property assessment. | |
| Deliverables (CDRL #) 1,2,3 | Completion Criteria/Milestone: Toolset ready for delivery and evaluation by Phase 1 demo |
| Supporting Orgs: SNC,NU | Duration (months): 18 |
| **1.2.4** Perform tasks for tool delivery | Objective: Package and test the AFACTS toolset to prepare for evaluation |
| Approach: Test the AFACTS toolset to find and resolve software, design and deployment defects. Provide technical direction for tool chain assembly and delivery to team performers. Develop deployment package to include instructions on deployment, deployment testing, use, and evaluation of toolset. | |
| Deliverables (CDRL #) 2,3 | Completion Criteria/Milestone: Toolset Deployment Package tested per test plan and delivered by Phase 1 demo |
| Supporting Orgs: SNC,NU | Duration (months): 1.5 |
| **1.2.5** Experiment Support | Objective: Support TA4 experimentation |
| Approach: Provide tools to TA4, help TA4 address TA4 platform-specific challenge problems, contribute to TA4 from internal test applications, consult on the application of TA4 techniques to the target platform and participate with technology development teams working on the TA4 challenge problems. | |

| | |
|---|---|
| Deliverables (CDRL #) 2,3 | Completion Criteria/Milestone: Contributed to TA4 e.g. we've met their requirements during last 6 months of Phase 1 |
| Supporting Orgs: SNC,NU | Duration (months): 6 |
| **1.3** TA2 - QoS-Aware Monitoring and Control Platform Development & Integration: Complete for TA2 Phase 1 | Objective: Produce qualified evidence regarding safety and correctness of the design and operation of LE-CPS |
| **1.3.1** Develop prototype model ingestion tools | Objective: Ingest TA1 Models into ROVER Monitoring and Control Services |
| Approach: Create prototype MIC tools that map system models defined in TA1 (which define QoS bounds and component distribution requirements for the components) into input formats that can be processed by the QoS-aware monitoring and control platform developed by TA2. | |
| Deliverables (CDRL #) 2,3 | Completion Criteria/Milestone: Prototype MIC tools ready for delivery and evaluation by Phase 1 demo |
| Supporting Orgs: VU, SNC, NU | Duration (months): 18 |
| **1.3.2** Develop prototype lightweight statistical monitors | Objective: Synthesize lightweight statistical monitors from models |
| Approach: Create prototype MIC tools that automatically synthesize lightweight statistical monitors compare system QoS and behaviors over a time window against expected QoS measures and measures of the validity of evidence derived dynamically from services provided by the TA3 team. | |
| Deliverables (CDRL #) 2,3 | Completion Criteria/Milestone: Prototype lightweight statistical monitors ready for delivery and evaluation by Phase 1 demo |
| Supporting Orgs: VU, S | Duration (months): 18 |
| **1.3.3** Develop prototype adaptive resource management service | Objective: Attempt to remedy problematic assurance cases by adjusting and/or adaptively reconfiguring relevant LE-CPS resources |
| Approach: Develop prototype adaptive resource management service middleware that performs actions which attempt to remedy problematic assurance cases by adjusting relevant platform resources and properties, such as QoS levels, structural relationships, and dynamic behaviors. | |
| Deliverables (CDRL #) 1,2,3 | Completion Criteria/Milestone: Prototype adaptive resource management service middleware ready for delivery and evaluation by Phase 1 demo |
| Supporting Orgs: VU | Duration (months): 18 |
| **1.3.4** Develop prototype alert management service | Objective: Map alerts from lightweight statistical monitors to TA3 services |
| Approach: Develop a prototype alert management service middleware that automatically triggers alerts when significant statistical deviation occurs and propagates these alerts in a reliable and timely manner to the dynamic assurance models managed by services provided by the TA3 team. | |
| Deliverables (CDRL #) 2,3 | Completion Criteria/Milestone: Prototype alert management service middleware ready for delivery and evaluation by Phase 1 demo |
| Supporting Orgs: VU, S | Duration (months): 18 |
| **1.3.5** Experiment Support | Objective: Support TA4 experimentation |
| Approach: Provide tools to TA4, help TA4 address TA4 platform-specific challenge problems, contribute to TA4 from internal test applications, consult on the application of TA4 techniques to the target platform and participate with technology development teams working on the TA4 challenge problems. | |
| Deliverables (CDRL #) 2,3 | Completion Criteria/Milestone: Contributed to TA4 e.g. we've met their requirements during last 6 months of Phase 1 |
| Supporting Orgs: VU | Duration (months): 6 |
| **1.4** TA3 - Research and Prototype Tools: Complete for TA3 Phase 1 | Objective: Automate assurance case synthesis and run-time evaluation for LE-CPS |
| **1.4.1** Develop TA3 Service APIs | Objective: Develop service API to support interfaces between TA1, TA2, TA4 |
| Approach: Define and develop the data types, protocols, and interfaces that will enable communication between the TAs | |
| Deliverables (CDRL #) 2,3 | Completion Criteria/Milestone: Published API |
| Supporting Orgs: S | Duration (months): 18 |
| **1.4.2** Gather/define domain knowledge relevant to CPs to support AC synthesis | Objective: Extend the domain model to support the alignment of evidence to supportive Assurance claims |
| Approach: Leverage Securboration's past experience in semantic modeling to develop a domain model relevant to the team challenge problems that supports claim synthesis | |
| Deliverables (CDRL #) 2,3 | Completion Criteria/Milestone: Domain model delivery and evaluation by phase 1 demo |
| Supporting Orgs: S | Duration (months): 18 |
| **1.4.3** Implement Ratiocination Engine | Objective: Develop tools to synthesize high-fidelity assurance cases from low-fidelity, claim-oriented GSN assurance case templates provided by TA1 |
| Approach: Design a RE that permits rulesets (e.g., deductive, forward inductive, eliminative inductive, defeasible) to be plug-and-play on a per-AC basis. Explore several relevant rulesets relevant to the team challenge problems. | |
| Deliverables (CDRL #) 1,2,3 | Completion Criteria/Milestone: Toolset ready for delivery and evaluation by Phase 1 demo |
| Supporting Orgs: S | Duration (months): 18 |
| **1.4.4** Implement Confidence Engine | Objective: Develop tools to support confidence cases, consisting of (1) an assurance case, (2) confidence values for each of the claims in the assurance case, (3) a rationale for the assigned values based on observed evidence. |

| | |
|---|---|
| Approach: Develop CE tool to ingest assurance cases synthesized by the RE and metrics published by probes injected into the running application by TA2. The CE subsequently evaluates the ACs and emits confidence cases | |
| Deliverables (CDRL #) 1,2,3 | Completion Criteria/Milestone: Toolset ready for delivery and evaluation by Phase 1 demo |
| Supporting Orgs: S | Duration (months): 18 |
| **1.4.5** Process Evidence | Objective: Develop techniques to process evidence from TA2 that are aligned to claims in the Confidence Engine |
| Approach: Dynamically identify evidence-to-claim mappings and requery TA2 for supporting evidence when confidence case integrity is low. Work with TA1 to define this requery API. | |
| Deliverables (CDRL #) 2,3 | Completion Criteria/Milestone: Toolset ready for delivery and evaluation by Phase 1 demo |
| Supporting Orgs: S | Duration (months): 18 |
| **1.4.6** Evaluate Dynamic Assurance | Objective: Develop techniques to map violations to assurance case to corrective actions performed by the application |
| Approach: Identify claim violations and map them to possible corrective actions. Work with TA1 and TA2 to understand these corrective actions and the conditions under which they should be enacted. Transmit the resultant information to reactive code regions injected into the application by TA2. | |
| Deliverables (CDRL #) 2,3 | Completion Criteria/Milestone: Toolset ready for delivery and evaluation by Phase 1 demo |
| Supporting Orgs: S | Duration (months): 9 |
| **1.4.5** Experiment Support | Objective: Support TA4 experimentation |
| Approach: Provide tools to TA4, help TA4 address TA4 platform-specific challenge problems, contribute to TA4 from internal test applications, consult on the application of TA4 techniques to the target platform and participate with technology development teams working on the TA4 challenge problems. | |
| Deliverables (CDRL #) 2,3 | Completion Criteria/Milestone: Contributed to TA4 e.g. we've met their requirements during last 6 months of Phase 1 |
| Supporting Orgs: S | Duration (months): 6 |

| Reporting Deliverables | | |
|---|---|---|
| CDRL# | Reporting Deliverables Description | Date-Frequency (MAC=Months after Contract) |
| 1 | Technical Papers | One month before all Figure X events |
| 2 | Design Package: Algorithm and Interface Description Document, user guides, other necessary data, and documentation, assumptions and limitations | One month before all Figure X events |
| 3 | Toolset Deployment package: Source code and supporting test and tools | One month before all Figure X events |
| 4 | Slide Presentations | One month before all Figure X events |
| 5 | Quarterly Progress Reports | Quarterly |
| 6 | Monthly Progress Reports | Monthly |
| 7 | Financial Reports | Monthly |
| 8 | Final Phase Report | 10/19 and 11/20 - One month before Phase 1 and 2 demonstrations |
| 9 | Final Technical Report | 2/22 - One month before Capstone demonstration |

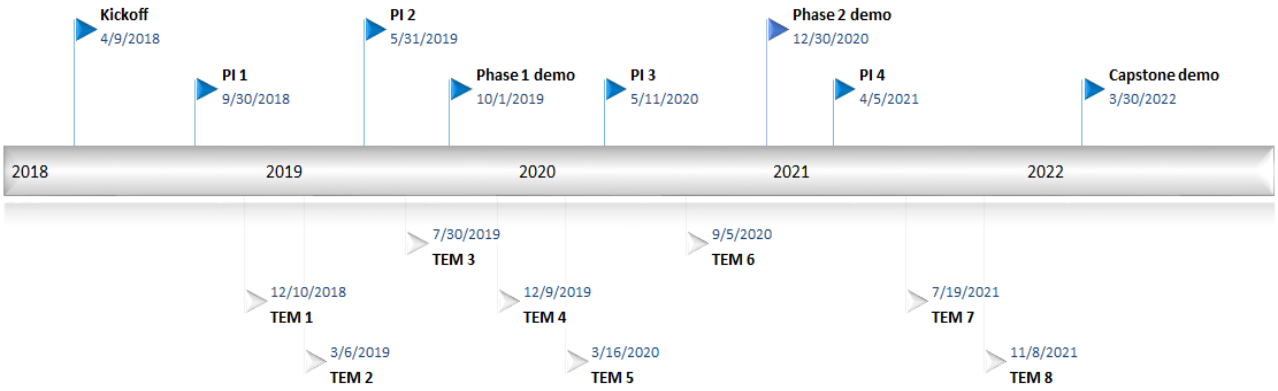



*Figure 5. Schedule, Milestones, and Deliverables*

### I.G.2 Metrics

We will evaluate the performance of our proposed MODERNISED, ROVER, and DAIS technologies using metrics designed to measure AFACTS' ability to scale to 100 dimensions, with 10% less overhead, providing at least 1,000 conditional evidence and .001x reduced trials to assurance. Monitoring tasks have traditionally been scheduled based on a periodic time schedule [CoPilot1][CoPilot2][ACM]. In LE-CPS with real-time QoS constraints, however, this approach creates additional overhead that reduces the amount of resources available for system tasks that run concurrently. Though this overhead cannot be removed completely, the AFACTS' TA2 ROVER middleware reduces this impact by adaptively scheduling its *Lightweight Statistical Monitors* (see §I.C.2.3.2.2) depending upon the criticality of the mission phase (e.g., in the UTACC example ROVER samples the GPS data less often when the robot(s) are further away from the enemy territory) and the likelihood of errors (which can be derived from the past performance of the LE-CPS in that context and the mission phase).

Another problem for LE-CPS involves the impact of reconfiguring the run-time system to recover from likely failures. Often, redundant deployments of software components enable dynamic masking of failures. These redundant deployments are also resource intensive, however, so there is a tradeoff between the level of redundancy and required reliability is desirable, i.e., the LE-CPS can be reconfigure proactively when the likelihood of failures become apparent. Thus, the AFACTS' TA2 ROVER middleware supports the dynamic reconfiguration via its *Adaptive Resource Management Service* (see **I.C.2.3.2.3**) to enable applications to use a different sets of components that are less likely to fail when monitoring data indicates high likelihood of failures.

Table 10 contains a summary of our proposed metrics and the specific technical approach to realize each metric objective.

***Table 10. Assured Autonomy Program Metrics for AFACTS Evaluation Measure Progress Against All Thrusts***

| Metric | Description | Technical Approach Ref |
|---|---|---|
| Number of sensors streams being sampled simultaneously | Adaptively schedule *Lightweight Statistical Monitors* (LSM) depending upon the criticality of the mission phase and the likelihood of errors (which can be derived from the past performance of the LE-CPS in that context and the mission phase). | See discussion of ROVER's *Lightweight Statistical Monitors* in §I.C.2.3.2.2 |
| Availability of the critical functions of the system during mission | Support dynamic reconfiguration via its *Adaptive Resource Management Service* (ARMS) to enable applications to use a different sets of components that are less likely to fail when monitoring data indicates high likelihood of failures | See discussion of ROVER's *Adaptive Resource Management Service* in §I.C.2.3.2.3 |
| Quantitative assessment of a safety cases, e.g., (1) the number of non-verified goals during different stages of validating safety cases and (2) volume and quality of conditional evidence derived from proof traces | TA2 monitors the rate where the number of individuals with unverified goals decrease. Analysis of proof traces & reasoning conclusions provide steps (conditional evidence) that is verified in TA3. Claim proofs are based on assumed ground (unconditional) evidence captured by model presumptions, viz. to infer that a specific configuration of team member is necessary for the column formation, and this is not the case, then SHACL's precise reasoning will identify violations. | See discussion of *Detailed TA1 Technical Approach – MODel intEgrated fRamework for autoNomous, hIgh aSsurancE Design* (MODERNISED) in §I.C.2.1 |

# Appendix A

(1) Team Member Identification: Provide a list of all individual team members from the prime, subcontractor(s), and consultant(s), as applicable. Identify specifically whether any are a non-US organization or individual, FFRDC and/or Government entity. Use the following format for this list:

| Individual Name | Role (Prime, Sub-contractor, or Con-sultant | Organization | Non-US? | | FFRDC Or Govt? |
|---|---|---|---|---|---|
| | | | Org | Ind. | |
| Sierra Nevada Corporation | Prime | Large Business | No | No | No |
| Northeastern University | Subcontractor | University | No | No | No |
| Vanderbilt Uni-versity | Subcontractor | University | No | No | No |
| Securboration | Subcontractor | Small Business | No | No | No |

(2) Government or FFRDC Team Member Proof of Eligibility to Propose:
If none of the team member organizations (prime or subcontractor) are a Government entity or FFRDC, state "NONE".
**NONE**

(3) Government or FFRDC Team Member Statement of Unique Capability: If none of the team member organizations (prime or subcontractor) are a Government entity or FFRDC, state "NONE".
**NONE**

(4). Organizational Conflict of Interest Affirmations and Disclosure: If none of the proposed team members is currently providing SETA or similar support as described in §III.B, state "NONE".
**NONE**

(5). Intellectual Property (IP): If no IP restrictions are intended, state "NONE". The Government will assume unlimited rights to all IP not explicitly identified as having less than unlimited rights in the proposal.
**NONE**

(6). Human Subjects Research (HSR): If HSR is not a factor in the proposal, state "NONE".
**NONE**

(7). Animal Use: If animal use is not a factor in the proposal, state "NONE".
**NONE**

(8). Representations Regarding Unpaid Delinquent Tax Liability or a Felony Conviction under Any Federal Law: For further information regarding this subject, please see www.darpa.mil/work-with-us/additionalbaa.

Please also complete the following statements.
(1) The proposer is [ ] **is not [x]** a corporation that has any unpaid Federal tax liability that has been assessed, for which all judicial and administrative remedies have been exhausted or have lapsed, and that is not being paid in a timely manner pursuant to an agreement with the authority responsible for collecting the tax liability,
(2) The proposer is [ ] **is not [x]** a corporation that was convicted of a felony criminal violation under a Federal law within the preceding 24 months.

(9). Cost Accounting Standards (CAS) Notices and Certification: For any proposer who submits a proposal which, if accepted, will result in a CAS compliant contract, must include a Disclosure Statement as required by 48 CFR 9903.202. Further information regarding the CAS notices and certification can be found in the FAR 52.230-1, as well as at www.darpa.mil/work-withus/additional-baa.

If this section is not applicable, state "NONE".

**Disclosure statement provided as part of Cost Volume.**

# Appendix B – Bibliography

[Adams] M.M. Adams and P.B. Clayton, "ClawZ: Cost-Effective Formal Verification for Control Systems", INSPEC Accession Number: 8802762, 24th Digital Avionics Systems Conference, 2005. DASC 2005.

[Adesina-Ojo] Ayodele A. Adesina-Ojo, John A. van der Poll, and Lucas M.Venter, "Towards the Formalisation of Object-oriented Methodologies", Proceedings of the South African Institute of Computer Scientists and Information Technologists Conference on Knowledge, Innovation and Leadership in a Diverse, Multidisciplinary Environment, 2011, pp.259-262.

[Arthan] R. Arthan, P. Caseley, C. O'Halloran, A. Smith, "ClawZ: Control laws in Z", Third IEEE International Conference on Formal Engineering Methods, 2000. ICFEM 2000.

[Baclawski] K. Baclawski, M. Kokar, P. Kogut, L. Hart, J. Smith, W. Holmes, J. Letkowski, M. Aronson, P. Emery, "Extending the UML for Ontology Development", SOSYM 2002, Software System Model (2002) 1: 1-15, Springer-Verlag 2002.

[Bapty] Bapty, T., S. Neema, J. Scott, J. Sztipanovits, and S. Asaad, "Model-Integrated Tools for the Design of Dynamically Reconfigurable Systems", VLSI Design, vol. 10, pp. 281--306, 2000.

[Blanchette] S. Blanchette, "Assurance cases for design analysis of complex system of systems software," in Proc. of the AIAA Infotech@Aerospace Conference, Seattle, WA, 6-9 April 2009.

[Bloomfield] Bloomfield, R., Littlewood, B., & Wright, D. "Confidence: Its Role in Dependability Cases for Risk Assessment", pp. 338-346. Proceedings of the International Conference on Dependable Systems and Networks (DSN 2007). Edinburg, U.K., April 2007. IEEE, 2007.

[Bovens] Bovens L., Hartmann S. (2003) *Bayesian epistemology*. Oxford University Press, Oxford, 2003.

[Buschmann1] Frank Buschmann, Kevlin Henney, and Douglas C. Schmidt, *Pattern-Oriented Software Architecture: On Patterns and Pattern Languages*, Wiley and Sons, 2007.

[Buschmann2] Frank Buschmann, Kevlin Henney, and Douglas C. Schmidt, *Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing*, Wiley and Sons, 2007.

[Cavalcanti] Ana Cavalcanti, Phil Clayton and Colin O'Halloran, "From Control Law Diagrams to Ada via Circus", Formal Aspects of Computing, July 2011, Volume 23, Issue 4, pp 465–512.

[Chakrabarti] Arindam Chakrabarti, Pallab Dasgupta, P. P. Chakrabarti, and Ansuman Banerjee, "Formal Verification of Module Interfaces Against Real Time Specifications", Proceedings of the 39th Annual Design Automation Conference, 2002, pp. 141-145.

[CHARIOT1] Subhav Pradhan, Abhishek Dubey, Tihamer Levendovszky, Pranav Srinivas Kumar, William A. Emfinger, Daniel Balasubramanian, William Otte, and Gabor Karsai, "Achieving Resilience in Distributed Software Systems via Self-reconfiguration," Journal of Systems and Software, 122():344 - 363, 2016. DOI http://dx.doi.org/10.1016/j.jss.2016.05.038.

[Cioara] Tudor Cioara, Ionut Anghel, Ioan Salomie, Mihaela Dinsoreanu, Georgiana Copil, and Daniel Moldovan, "A Reinforcement Learning Based Self-healing Algorithm for Managing Context Adaptation", Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services, 2010, pp. 859-862.

[CISQ1] Automated Source Code Maintainability Measure, Version 1.0, 2016, http://www.omg.org/spec/ASCMM/1.0.

[CISQ2] Automated Source Code Performance Efficiency Measure, Version 1.0, 2016, http://www.omg.org/spec/ASCPEM/1.0.

[CISQ3] Automated Source Code Reliability Meassure, version 1.0, 2016, http://www.omg.org/spec/ASCRM/1.0.

[CISQ4] Automated Source Code Security Meassure, version 1.0, 2016, http://www.omg.org/spec/ASCSM/1.0.

[Cohen] Cohen, J. "An Introduction to the Philosophy of Induction and Probability", Clarendon, 1989.

[CoPilot1] Lee Pik , Alwyn Goodloe, Robin Morisset, and Sebastian Niller, "Copilot: A Hard Real-Time Runtime Monitor," proceedings of the 1st International Conference on Runtime Verification (RV 2010), St. Julians, Malta, November 1-4, 2010.

[CoPilot2] Lee Pike and Nis Wegmann and Sebastian Niller and Alwyn Goodloe, "Copilot: Monitoring Embedded Systems," Innovations in Systems and Software Engineering: Special Issue on Software Health Management, Springer, volume 9, number 4, 2013, pp. 235-255.

[Coppit] Coppit, D., Sullivan, K. J., and Dugan, J. B. Formal semantics of models for computational engineering: A case study on dynamic fault trees. In Proceedings of the International Symposium on Software Reliability Engineering, pages 270–282, San Jose, California, 8–11 Oct. 2000. IEEE.

[Denney] Denney, E., Pai, G., Pohl, J.: AdvoCATE: An Assurance Case Automation Toolset. In: Ortmeier, F., Daniel, P. (eds.) SAFECOMP Workshops 2012. LNCS, vol. 7613, pp. 8–21. Springer, Heidelberg (2012)

[Dubey] A. Dubey, G. Karsai and N. Mahadevan, "Model-based Software Health Management for Real-time Systems," 2011 Aerospace Conference, Big Sky, MT, 2011, pp. 1-18.

[Dubey2] Saideep Nannapaneni, Sankaran Mahadevan, Abhishek Dubey, David Lechevalier, Anantha Narayanan, and Sudarsan Rachuri. Automated uncertainty quantification through information fusion in manufacturing processes. Journal of Sustainable and Smart Manufacturing Systems, 2017.

[Edmondson] James Edmondson and Douglas C. Schmidt, "Multi-Agent Distributed Adaptive Resource Allocation (MADARA)," International Journal of Communication Networks and Distributed Systems (IJCNDS), Special Issue on: Grid Computing, Edited by Michal Wozniak and Krzysztof Walkowiak, Volume 5, Number 3, 2010, pp. 229-245.

[Galindo1] Jose Galindo, David Benavides, Hamilton Turner, Jules White, “Testing Variability Intensive Systems Using Automated Analysis: An Application to Android,” Springer Journal of Systems and Software, Volume 24, Issue 2, pp. 365–405, June, 2016.

[Gaudin] Benoit Gaudin, Emil Iordanov Vassev, Patrick Nixon, and Michael Hinchey, “A Control Theory Based Approach for Self-healing of Unhandled Runtime Exceptions”, Proceedings of the 8th ACM International Conference on Autonomic Computing, 2011, pp. 217-220.

[Goodenough] J. Goodenough, C. Weinstock and A. Klein, “Toward a Theory of Assurance Case Confidence”, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2012.

[Gray1] Gray, Jeff, Ted Bapty, Sandeep Neema, and James Tuck, “Handling Crosscutting Constraints in Domain-Specific Modeling”, Communications of the ACM 44, no. 10 (2001): 87-93.

[Groza] Adrian Groza and Nicoleta Marc, ”Consistency Checking of Safety Arguments in the Goal Structuring Notation Standard”, INSPEC Accession Number: 14702935, 2014 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP).

[GSN-Std] GS Community Standard Version 1.
http://www.goalstructuringnotation.info/documents/GSN_Standard.pdf .

[Habli] Ibrahim Habli, Tim Kelly, “A Generic Goal-Based Certification Argument for the Justification of Formal Analysis”, Elsevier Electronic Notes in Theoretical Computer Science 238 (2009) 27–39.

[Hawkins] Hawkins, R., Habli, I., Kelly, T.P.: The Need for a Weaving Model in Assurance Case Automation. Ada User Journal 36(3), 187–191 (Sep 201

[Hobbs] Jerry R. Hobbs , Mark Stickel , Paul Martin , Douglas Edwards, “Interpretation as abduction”, Proceedings of the 26th annual meeting on Association for Computational Linguistics, p.95-103, June 07-10, 1988, Buffalo, New York.

[Horkoff] Jennifer Horkoff, and Eric Yu, “Analyzing Goal Models: Different Approaches and How to Choose Among Them”, Proceedings of the 2011 ACM Symposium on Applied Computing, 2011, pp. 675-682.

[IJPHM] Saideep Nannapaneni, Abhishek Dubey, Sherif Abdelwahed, Sankaran Mahadevan, Sandeep Neema, and Ted Bapty, “Mission-based Reliability Prediction in Component-based Systems”, International Journal of Prognostics and Health Management, February 2016.

[Kelly] “The Goal Structuring Notation – A safety argument notation”, Kelly, Weaver - 2004.

[Koeman] Vincent J. Koeman, Koen V. Hindriks, and Catholijn M. Jonker, “Automating Failure Detection in Cognitive Agent Programs”, Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems, 2016, pp. 1237-1246.

[KokarEndsley] M. M. Kokar and M. R. Endsley. Situation awareness and cognitive modeling. IEEE Intelligent Systems, 27, no. 3:91-96, 2012.

[Kuroe] Y. Kuroe, H. Inayoshi and T. Mori, “Feedback-error-learning control with considering smoothness of unknown nonlinearities”, International Conference on Neural Networks, 12-12 June 1997, INSPEC Accession Number: 5691638.

[Lardieri] Patrick Lardieri, Jaiganesh Balasubramanian, Douglas C. Schmidt, Gautam Thaker, Aniruddha Gokhale, and Tom Damiano, “A Multi-layered Resource Management Framework for Dynamic Resource Management in Enterprise DRE Systems,” the *Journal of Systems and Software*: special issue on Dynamic Resource Management in Distributed Real-Time Systems, editors C. Cavanaugh and F. Drews and L. Welch, Vol 80, Issue 7, July 2007, pgs. 984-996.

[MOF2RDF] MOF to RDF Mapping, OMG Revised Submission document ad/2015-11-11, 2015

[Neema1] Neema, S., T. Bapty, and J. Scott, “Development Environment for Dynamically Reconfigurable Embedded Systems”, Proceedings of the International Conference on Signal processing Applications and Technology, Orlando, FL, November, 1999.

[Neema2] Scott, J., T. Bapty, S. Neema, and J. Sztipanovits, “Model-Integrated Environment for Adaptive Computing”, Proceedings of the Military and Aerospace Applications of Programmable Devices and Technologies Conference, Greenbelt, MA, September, 1998.

[Neema3] Balasubramanian, Krishnakumar, Aniruddha Gokhale, Gabor Karsai, Janos Sztipanovits, and Sandeep Neema. “Developing Applications Using Model-Driven Design Environments”, Computer 39, no. 2 (2006): 33-40.

[OMG1] OMG Systems Modeling Language, version 1.5, 2016, http://www.omg.org/spec/SysML/1.5.

[OMG2] Semantics of a Foundational Subset for Executable UML Models (fUML), version 1.3, 2017, http://www.omg.org/spec/FUML/1.3.

[OMG3] Precise Semantics of UML Composite Structures (PSCS), version 1.1, 2017, http://www.omg.org/spec/PSCS/1.1.

[OMG4] Precise Semantics of UML State Machines (PSSM), version 1.0, 2017, http://www.omg.org/spec/PSSM/1.0.

[OMG5] Action Language for Foundational UML (Alf), version 1.1, 2017, http://www.omg.org/spec/ALF/1.1.

[OMG6] SysML-Modelica Transformation, version 1.0, 2012, http://www.omg.org/spec/SyM/1.0.

[OMG7] SysML Extension for Physical Interaction and Signal Flow Simulation (SysPISF), version 1.0, 2016, http://www.omg.org/spec/SysPISF/1.0.

[OMG8] UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems, version 1.1, 2011, http://www.omg.org/spec/MARTE/1.1.

[OMG9] Archetype Modeling Language (AML), version 1.0, 2017, http://www.omg.org/spec/AML/1.0.

[OWL] W3C. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition). https://www.w3.org/TR/owl2-syntax/.

[Pollock] J. Pollock, "Defeasible Reasoning," in Reasoning: Studies of Human Inference and Its Foundations, J. E. Adler and L. J. Rips, Eds., Cambridge University Press, 2008, pp. 451-469.

[Rhodes] Rhodes T., Boland F., Fong E., Kass M.: "Software Assurance Using Structured Assurance Case Models", NIST Interagency Report 7608, US Department of Commerce (2009)

[Riemsdijk1] M. Birna van Riemsdijk, Mehdi Dastani, and Michael Winikoff, "Goals in agent systems: a unifying framework", Proceedings of the 7th International joint Conference on Autonomous Agents and Multiagent Systems, 2008, pp. 713-720.

[Riemsdijk2] M. Birna van Riemsdijk, "Cognitive Agent Programing", PhD Thesis, University of Utrecht, 2006.

[Riemsdijk3] M. Birna van Riemsdijk, Louise A. Dennis, Michael Fisher, and Koen V. Hindriks, "Agent Reasoning for Norm Compliance: A Semantic Approach", Proceedings of the 2013 International Conference on Autonomous Agents and Multiagent Systems, 2013, pp. 499--506.

[Rushby] Rushby, J. "The interpretation and evaluation of assurance cases" Technical Report SRI-CSL-15-01, Computer Science Laboratory, SRI International, Menlo Park, CA (2015).

[Sabatucci] Luca Sabatucci, and Massimo Cossentino, "From Means-end Analysis to Proactive Means-end Reasoning", Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, 2015, pp. 2-12.

[SACM] Structured Assurance Case Metamodel, version 2.0, 2017, http://www.omg.org/spec/SACM/2.0.

[Schmidt1] Douglas C. Schmidt, "Model-Driven Engineering," *IEEE Computer*, Vol. 39, No. 2, February 2006, pp. 41-47.

[Schmidt2] Douglas C. Schmidt, Rick Schantz, Mike Masters, Joseph Cross, David Sharp, and Lou DiPalma, "Towards Adaptive and Reflective Middleware for Network-Centric Combat Systems," Crosstalk, November, 2001.

[SHACL] W3C. Shapes Constraint Language (SHACL): W3C Recommendation 20 July 2017. https://www.w3.org/TR/shacl/

[Shankaran1] Nishanth Shankaran, Xenofon Koutsoukos, Chenyang Lu, Douglas C. Schmidt, and Yuan Xue, "Hierarchical Control of Multiple Resources in Distributed Real-time and Embedded Systems," the Springer *Real-time Systems Journal*, Volume 39, Numbers 1-3, August, 2008, pgs. 237-282.

[Shankaran2] Nishanth Shankaran, John Kinnebrew, Xenofon Koutsoukos, Chenyang Lu, Douglas C. Schmidt, and Gautam Biswas, "An Integrated Planning and Adaptive Resource Management Architecture for Distributed Real-time Embedded Systems," *IEEE Transactions on Computers*, Special Issue on Autonomic Network Computing, Volume 58, Number 11, 1485-1498, November 2009.

[Saruwatari] T. Saruwatari and S. Yamamoto. Definition and application of an assurance case development method (d*). Journal of SpringerPlus, 2(1):1–8, May 2013.

[Smith1] J. Smith, M. Kokar and K. Baclawski, Formal Verification of UML Diagrams: A First Step Towards Code Generation, UML 2001, Practical UML-Based Rigorous Development Methods Workshop, Toronto, CA, Oct 2001. Also published with Andy Evans, Robert France, Ana Moreira and Bernhard Rumpe (Eds.), in Practical UML-Based Rigorous Development Methods - Countering or Integrating the eXtremists, Lecture Notes in Informatics, GI-Edition, Proceedings, Springer, pages 224-240, 2001.

[Smith2] J. Smith, UML Formalization and Transformation, Ph.D. Thesis, Northeastern University, College of Engineering, Dec.1999.

[Sullivan] K. Sullivan, J. Yang, D. Coppit, S. Khurshid, and D. Jackson. Software assurance by bounded exhaustive testing. In ISSTA, 2004.

[Sumit] Mohanty, Sumit, Viktor K. Prasanna, Sandeep Neema, and J. Davis. “Rapid Design Space Exploration of Heterogeneous Embedded Systems Using Symbolic Search and Multi-granular Simulation”, ACM SIGPLAN Notices 37, no. 7 (2002): 18-27.

[Sun1] Yu Sun, Jules White, Bo Li, Hamilton Turner, Michael Walker, “Automated QoS-Oriented Cloud Resource Optimization using Containers,” Springer Automated Software Engineering Journal, Volume 24, Number 1, pp. 101-137, March, 2017

[Sun2] Yu Sun, Jules White, Sean Eade, Douglas C. Schmidt, “ROAR: A QoS-Oriented Modeling Framework for Automated Cloud Resource Allocation and Optimization,” Journal of Software and Systems, Volume 116, pp. 146-161, June, 2016.

[Sun3] Yu Sun, Jeff Gray, Romain Delamare, Benoit Baudry, Jules White, “Automating the Management of Non-functional System Properties Using Demonstration-based Model Transformation”, Journal of Software Maintenance and Evolution Research and Practice, incorporating Software Process and Practice, Volume 25, Issue 12, pp. 1335-1356, 2013.

[Tehrani] Sobhan Yassipour Tehrani and Kevin Lano, “Temporal Logic Specification and Analysis for Model Transformations”, Proceedings of the Fourth International Workshop on Verification of Model Transformations co-located with Software Technologies: Applications and Foundations, pp. 2-11, 2015.

[Turner1] Hamilton Turner, Brian Dougherty, Jules White, Russell Kegley, Jonathan Preston, Douglas C. Schmidt, and Aniruddha Gokhale, “DRE System Performance Optimization with the SMACK Cache Efficiency Metric”, Springer Journal of Systems and Software, Volume 98, pp. 25-43, 2014.

[Vernon] Michael Vernon, Frank Zeyda and Ana Cavalcanti, “Communication Systems in ClawZ”, International Conference on Abstract State Machines, Alloy, B and Z, ABZ 2010: Abstract State Machines, Alloy, B and Z pp 334-34.

[Weinstock] C. B. Weinstock, J. B. Goodenough, and A. Z. Klein. “Measuring assurance case confidence using baconian probabilities” In Proceedings of the 1st International Workshop on Assurance Cases for Software-Intensive Systems (ASSURE), 2013.

[White1] Jules White, David Benavides, Tripti Saxena, Brian Dougherty, Douglas C. Schmidt, Jose A. Galindo, "Evolving Feature Model Configurations in Software Product Lines," Journal of Software and Systems, Volume 87, pp. 119-136, 2014.

[White2] Jules White, Brian Dougherty, Chris Thompson, Douglas C. Schmidt, "ScatterD: Spatial Deployment Optimization with Hybrid Heuristic/Evolutionary Algorithms," ACM *Transactions on Autonomous and Adaptive Systems* Special Issue on Spatial Computing, Volume 6 Issue 3, September 2011, 18:1--18:25.

[White3] Jules White, Brian Dougherty, Richard Schantz, Douglas C. Schmidt, Adam Porter, and Angelo Corsaro, "R&D Challenges and Solutions for Highly Complex Distributed Systems: a Middleware Perspective," the Springer Journal of Internet Services and Applications special issue on the Future of Middleware, Volume 2, Number 3, December 2011, pp. 1-8.

[White4] Jules White, Christin Groba, Siobhan Clarke, Brian Dougherty, Chris Thompson, and Douglas C. Schmidt, "R&D Challenges and Solutions for Mobile Cyber-Physical Applications and Supporting Internet Services," the Springer Journal of Internet Services and Applications, Volume 1, Number 1, 2010, pp. 45-56.

[Zhu] Q. Zhu, D. Wei, K.Ji] Cyber Security for Industrial Control Systems, Chapter 6, pp. 151–182, CRC Press 2016, Print ISBN: 978-1-4987-3473-8.