IDAS: Intent-Defined Adaptive Software

Mr. Jacob I. Torrey

Proposers Day

July 9, 2019

IDAS Objective

Develop technologies that capture software engineering intentions, to enable rapid adaptation of DoD systems.

Automate the software sustainment loop

Background

- The cost of software engineering as currently practiced is constraining the ability of the government to deploy new software-based capabilities, as over 70% of the federal IT budget is dedicated to operations and maintenance[1]

- Software today is brittle with respect to changes in requirements and/or computing resources, requiring frequent and ultimately unaffordable modernization efforts to maintain adequate functionality

- Management of complexity is a central problem for software engineering. A common approach is *concretization*, in which the software engineer chooses from a set of apparently or almost equivalent options a particular value that enables the resulting code to compile
  - Assuming that network latency will never exceed a particular value
  - Selecting a 16-bit integer to represent the horizontal velocity of a rocket
  - Implementing data management tools using the APIs of a specific cloud service offering

1. White House Office of Management and Budget, "Information Technology," US Federal Government, Washington, DC, 2017

Concretization

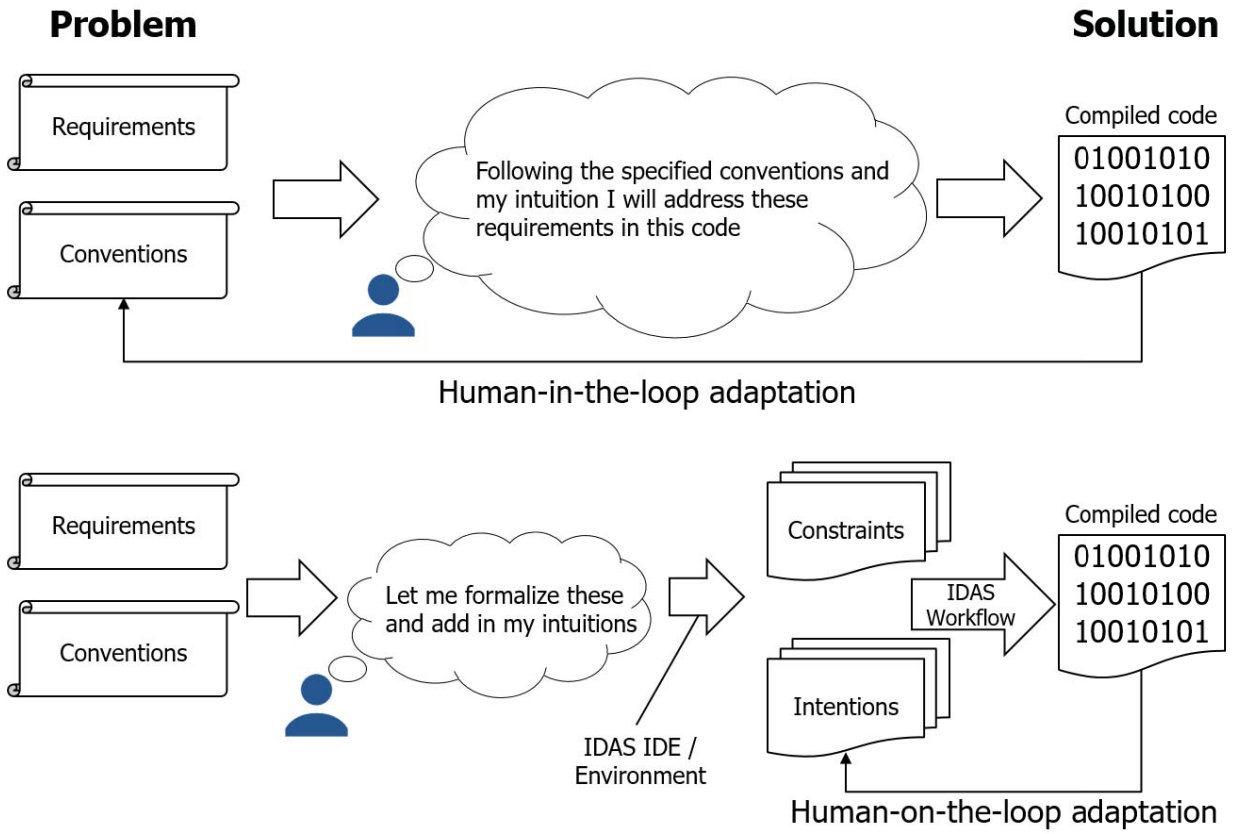
- Concretization makes software brittle, as it typically occurs at design time, when information about all possible future environments of the operational system is not available to guide the choice of concrete values or types
  - Leads to bugs when software is reused in a new context

- Concretization also creates inefficiencies in software, as engineers tend to choose conservative values that account for worst-case scenarios or future uncertainty but waste resources when handling the far more common average use cases

- A substantial number of concretization choices will be wrong at some point in a system's lifecycle, in the sense that they will hamper or preclude rapid adaptation to unanticipated requirements or changes in computing resources

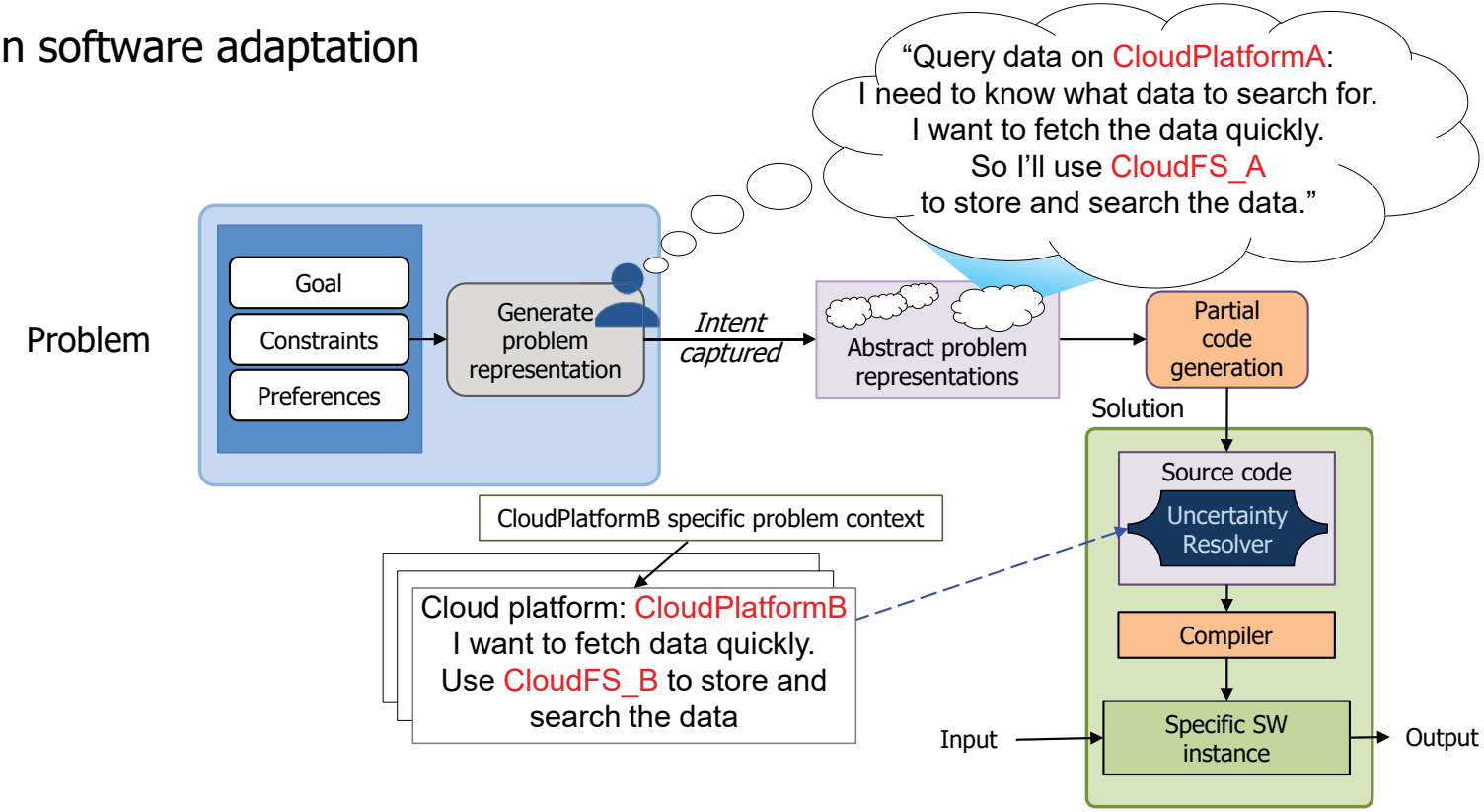
- IDAS goal: capture developer intent in the design/development process to allow concretization decisions to be deferred—easing adaptation to future changes in resources, platforms, and requirements

Intent-Defined Adaptive Software (IDAS)



IDAS vision

Intent-driven software adaptation



Develop technologies that capture programmer engineering intentions, to enable rapid adaptation to changing requirements and computing resources for all DoD software-enabled systems

Technical Areas

- TA1 – Automated software generation
- TA2 – Problem set generation
- TA3 – Integrated test and evaluation
- TA4 – Experimental control and transition

TA1: Automated software generation

Objective

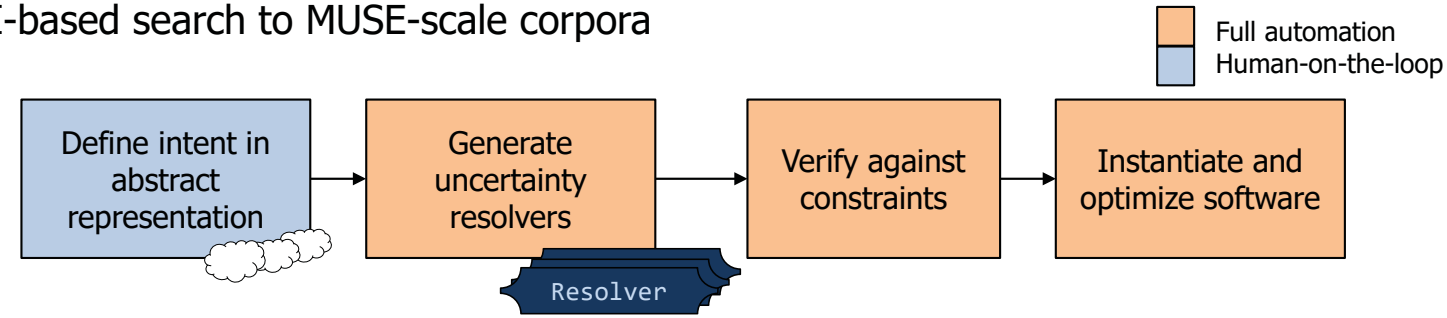
- Create engineering framework to rapidly generate adaptable software from programmer intentions

Challenges

- Single-step synthesis relies on exponential solvers that do not scale
- Problem and solution cannot be expressed in the same representation – human nature is to combine the two
- Must handle accumulation of errors and edge cases

Possible Approaches

- Model-based software engineering
- Expressively-typed languages to separate intentions and implementation
- Applying AI-based search to MUSE-scale corpora



Strong TA1 proposals should address the following

- Methods to capture programmer intent at design, development, or build time while keeping that intent separate from the concrete decisions needed to execute in a specific instance. These methods must be sufficiently intuitive to enable programmers to readily learn and apply them, and should not impose burdens that discourage adoption and use

- A compelling argument that the proposed methods can be learned and adopted by traditional developers without advanced, formal logic/mathematics training

- Automation technologies to generate or adapt software to new requirements, resources, and platforms. These technologies should reduce human-in-the-loop effort in the sustainment tail of software

- Evidence generation to provide assurance that the adapted software satisfies the new environment/requirements

TA2: Evaluation problem generation

- Develop sets of requirements and environments drawn from representative DoD needs without the security sensitivities specific to actual DoD systems

- Develop changes to requirements that will be released during temporally-compressed evaluation periods
  - 1 month to develop system with requirements (1.a)
  - Oh, sorry, requirements have changed, 1 week to retune to (1.b)
  - New policy requires new database architecture, 1 more week (1.c)

- Multiple problem domains to demonstrate true flexibility in approach
  - Logistics
  - Cloud agility
  - Held-back domain

IDAS problem domains

Multiple problem domains to demonstrate true flexibility in approach

- Logistics
  - Tracking heterogeneous inventory across multiple nations via multiple different means
  - Inventory non-fungible, unlikely for a single schema to support all edge cases
  - Legal, political, or technical changes occur post-development, changing requirements, E.g.,
    - Lithium ion batteries have special handling requirements after fire incidents
    - Import/export laws different between locations and routing impact costs

- Cloud agility
  - Platform-as-a-service has become commoditized across providers, allowing for ease of migration
  - Software-as-a-service more tied with services (e.g. APIs, data formats, etc.) inducing lock-in
  - Legal changes now impact software in the cloud, breaking the "location-less" abstraction
    - A serverless function processing personal information must change its behavior based on its physical location

Strong TA2 proposals should address the following

- Generation of evaluation exercises that subject all portions of the software development lifecycle to repeated changes in requirements, platforms, and available resources
- Methods for protecting any security sensitivities of the problem domains to ensure that the generated exercises can be released
- Identification of ways to reduce overall engineering effort during exercises that does provide experimental value, while still exploring the scalability of approaches
- Proposal of relevant third problem domain for exercises that is of comparable complexity
  - Explain why it is significant
  - Describe the types of events that would trigger adaption or requirements churn
- Representation of the end customer's needs in a variety of means to suite each TA1 and TA4 performer, from a written requirements document to meetings with developers in a more agile approach

TA3 & TA4: Experimentation

- TA3 (IT&E) validates solutions and assists TA2 with understanding how to best challenge TAs 1&4
  - Evaluate TA1 against benchmarks established by TA4 on TA2 problem sets
  - Manages separation between TAs 1&4 and TA2 to ensure fairness

- TA4 is transition & experimental control team following industry best practices
  - Competitive software vendor with Agile DevOps based approach
  - Phase 3: provide TA1 technology to TA4 and measure learning curve

TA1 Metrics (vs TA4 control)	Phase 1	Phase 2	Phase 3
Human effort (FTE per exercise)	150%	50%	10%
Human skillset	IDAS Expert	IDAS Journeyman	Traditional developer
Synthesis time (From problem to binary)	Days	~Day	~Hour

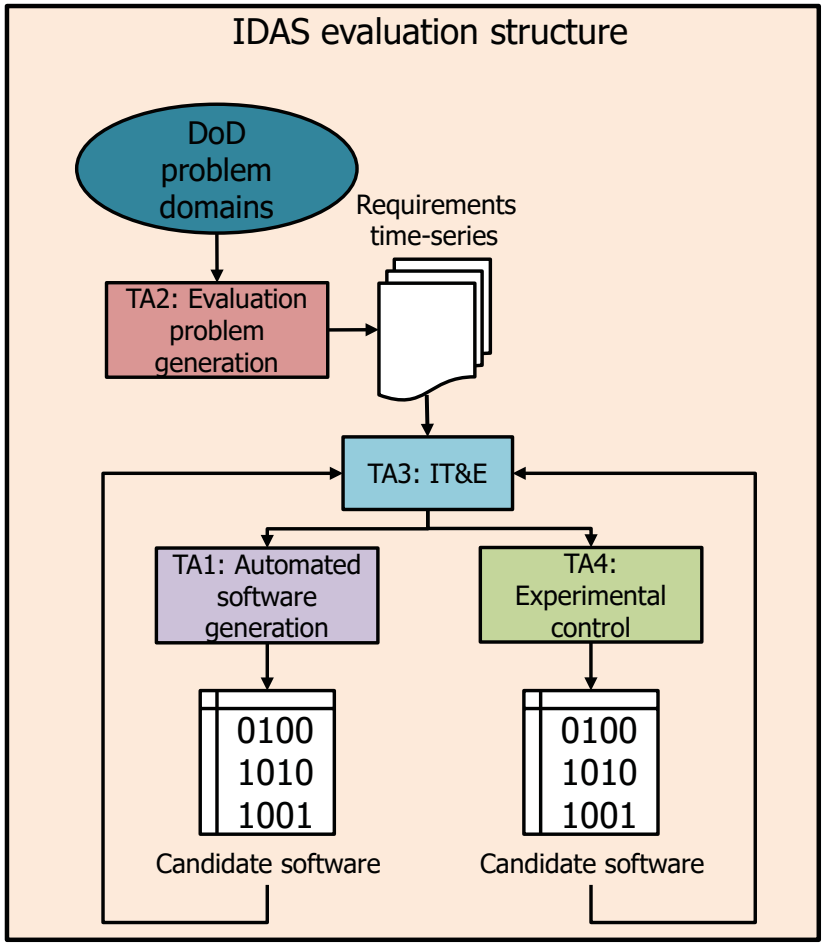
Strong TA3 proposals should address

- Guide TA2 problem set adaptation to differentiate and maximize exercise value to the IDAS program
- Evaluating TA1 compliance with the TA2 problem sets when requirements may be ambiguous or difficult to measure
- Measuring performance of TA1 approaches, both compared to TA4, as well as ranking strengths and weaknesses between each TA1 approach

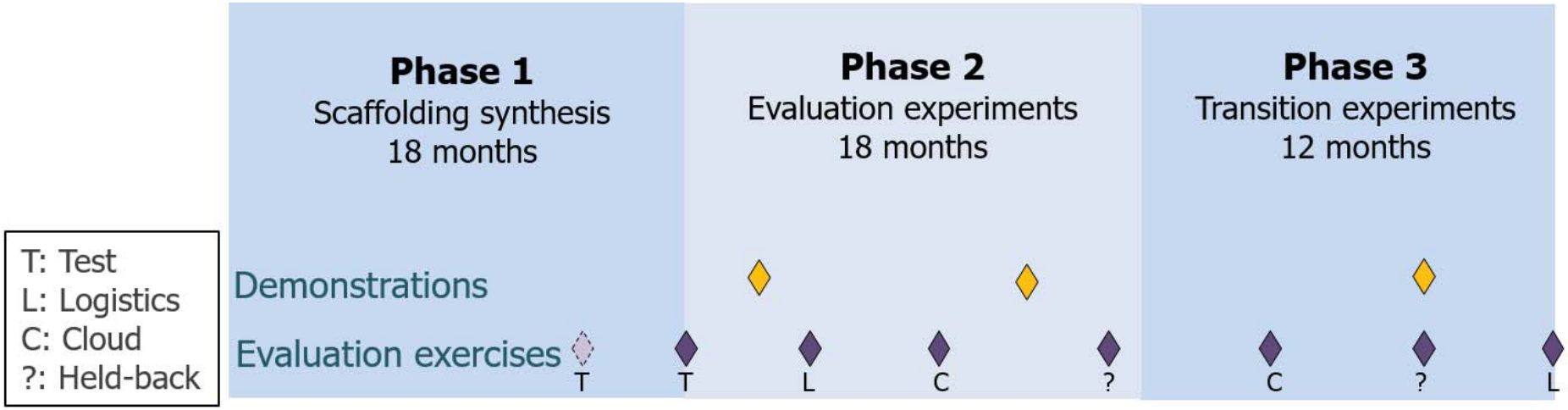
Strong TA4 proposals should provide

- Track record of applying agile development paradigms to produce large software systems
  - Preferably across a broad spectrum of platforms and architectures
- Detailed past use cases where agile development was used to rapidly address broad requirements or resources changes

IDAS evaluation exercise structure



IDAS programmatic details



Total program funding available for award is ~$65m over 4 years

- Multiple awards planned for TA1, single awards for TAs 2-4

Proposals may address only one Technical Area

Organizations can submit proposals to all Technical Areas

- Which to consider for award is at the discretion of the Government
- Only TA2 and TA3 may be awarded to a common organization

www.darpa.mil