

ASsurance as Code (ASaC)

Shuji Kinoshita (AIIIT, Japan)

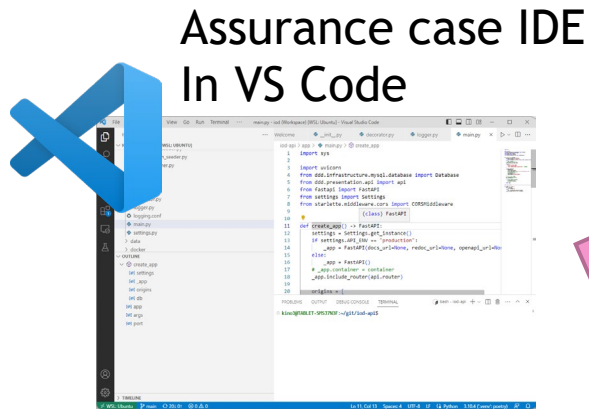
2023-09-21

Motivations

- ▶ The importance of system assurance is not only for mission-critical systems, but also for ordinary web services.
- ▶ However, many engineers (at least in Japan) do not even know the term “systems assurance”.
- ▶ I sometimes introduce assurance cases as a lecturer at IT companies, and many companies say that they know it is important, but it is difficult to introduce it to our company because of the increased cost.
- ▶ A mechanism is needed that would reduce the cost of assurance case development in two ways
 1. Facilitate the construction of the assurance case itself
 2. When the model or system that is the subject of the assurance case is updated, the assurance case is also semi-automatically modified



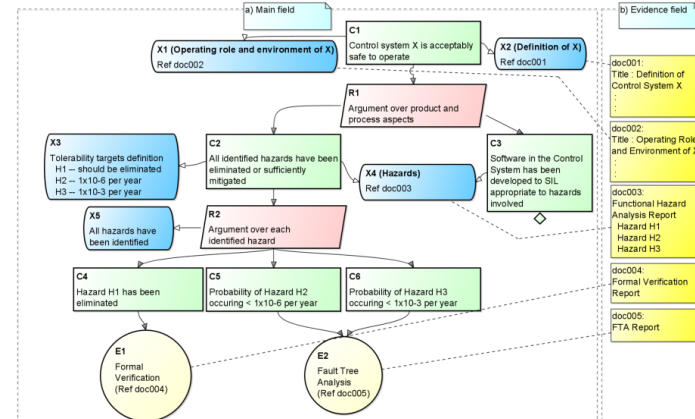
ASsurance as Code (ASaC) - an idea sketch



GitHub Copilot and
Other NLP Techs

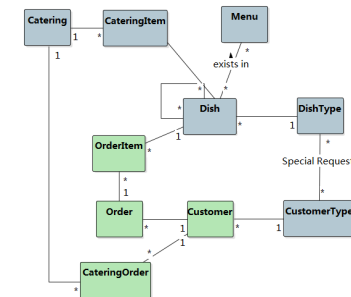
Generate

Linking



```
1SV/dec = { type = 'ISObject',  
  id: 1,  
  isElement = true,  
  isObject = true,  
  // Check if boxer is already active, return true if not  
  // If 'boxer' is active, return false  
  // Kill event  
  killEvent: function() {  
    // Cache internal data  
    data = { custom: 1, 1,  
      subwindow: 1,  
      body: 1,  
      target: 1,  
      subject: 1,  
      visible: 1,  
      resizeMode: 1,  
      touchTimer: 1,  
      gallery: 1,  
      active: false  
    }  
  }  
}
```

Codes



Models

Details

1. Programmatic assurance case description (by new programming language or DSL in host language)
2. Automatic generation of GSN and other representations from assurance case description language (SACM-based XML code generation)
3. Detection of changes by linking to evidence models and codes

It is also possible to support automatic code generation by AI. In other words, it will be possible to support assurance case writing through code generation support such as GitHub Copilot, and natural language processing can be applied to large-scale system development documents as input and a part of assurance case as output.

XaC Paradigm

Not so explicitly stated, the history of software development can be organized as "writing something in code" (X as Code).

- ▶ “Test as Code” -- the automation of testing, which used to be done manually by humans (XUnit)
- ▶ “Modeling as Code” -- the coding of modeling, represented by PlantUML or mermaid.js
- ▶ “Infrastructure as Code (IaC)” -- The coding of infrastructure definitions using Docker, terraform which has become popular since the late 2010s
- ▶ “Assurance as Code (ASaC)” would be next one?