

どんな研究？

ソフトウェアが社会の至るところに埋め込まれ、人の生活に深く関わるようになってきました。ますます複雑に、そして重要になるソフトウェアを、「賢く頼れる」ようにする研究です。

モデル検査：ボタン一発全自動（複雑すぎなければ・・・）

テスト：とにかくいろいろ動かしてみるの比較的手軽（無数にやってもバグを見つけられないかも・・・）

定理証明：複雑なシステムでも理論的に解きほぐして検証（解きほぐすためのヒントがあるかも・・・）

うまくいいとこどりして、人の理解・納得が伴う品質保証のためのモデル検証技術を！

どんな技術？

複雑なソフトウェアにおけるディペンダビリティの保証・向上のために、仕様や設計のモデルに対し、理論と経験・知能の両方を活かす検証アプローチを追求しています。

人工知能：経験による知識を得て、手探りを繰り返して検証の仕方を学習・進化（「だいたい」が得意で正確な○×は付けられない・・・）

物理世界とソフトウェアの世界が相互作用する複雑な動作を検証する

ポスター C02

車やロボットなど、物理世界とソフトウェア世界、全く異なる両者が連動するシステムにおいて、両側面をしっかりとらえ検証する高度技術に挑戦

設計・想定環境のモデルをシステム自身が常に検査・改善し続ける

ポスター B04

常に変化する実世界において、システム自身が動作を変えていく自己適応技術をIoTに応用

複雑なシステムの把握・保証の際に正確さを失わずにうまくぼかす

ポスター B06

安全性などの保証を厳密に行いつつ、その理解容易性や再利用性を追求

仕様内の記述同士のつながりを抽出して記述意図の理解を支援する

猿渡

根本となる要求とその実現方針など、様々な抽象度の仕様記述が作られるが、後から別の人が読んでそれらのつながりを把握するのは難しい！

このつながりを自動抽出して理解を支援

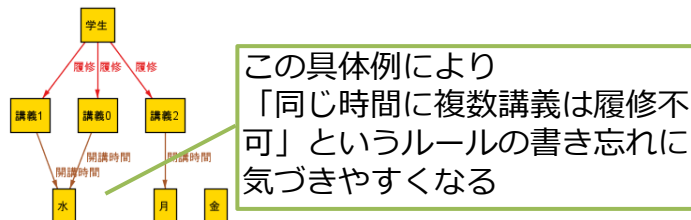
Diagram showing the extraction of requirements from abstract to specific descriptions. Includes text: 「信号が青のときのみ橋を渡ってもよい」, 「信号が青になるのは、反対から誰も渡ってこないときだけ」, 「反対から誰も渡ってこないときのみ橋を渡らせるようにする」

怪しいところをつつく具体例を作り記述の意味を人に実感・確認させる

佐伯

一般論の「ルール」が羅列された仕様書や規約などについて、意図に合っており、抜け・誤りがないことを確認するのは難しい！

ルールに合う具体例や合わない具体例の一式を、経験上抜けや誤りがありそうなものにしぼり生成



この具体例により「同じ時間に複数講義は履修不可」というルールの書き忘れに気づきやすくなる

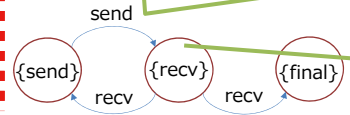
動作の個別ルールから、システム全体の動作の全可能性を描き出す

森田

多数の構成要素が自発的・反動的に動くシステムでは、個々の動作の組み合わせや順序により全体がどう振る舞うか把握することが難しい！

個々の動作の発火条件をもとにした抽象化により全体の動作における重要部分を端的に可視化

元の記述は「・・・が起きると・・・する」という無数の起動ルールの集まり



どういう順序で何が起きるか一目瞭然に