Socio-technical risks and simulation

Background and some observations

Yoav Hollander – Foretellix ltd

Agenda

- System Risk Management (SRM)
  - "Finding and qualifying risks in socio-technical systems"
  - What is it, how it is currently done, why improve on it
- Metric Driven Verification (MDV)
  - "The current simulation-based way to verify HW blocks / chips"
  - What is it, why it works well
- Simulation for SRM
  - Why MDV is not enough, what are the challenges
- Open questions

System Risk Management (SRM)

Some suggested definitions

- What is SRM?
  - "Finding and qualifying risks in socio-technical systems"
- What are socio-technical systems?
  - complex systems-of-systems comprising of electronics, software, mechanics, people and so on (also called cyber-physical systems)
  - Examples: airports, airplanes, robotic systems, military systems, hospitals, smart cities, energy systems, medical systems, …
- What kinds of risks?
  - Accident-causing risks:
    - Plugger friendly-fire, Bhopal, Ariane 5, aviation, …
  - Project risks:
    - Project arrives late or never, or has low availability / maintainability, e.g. because of inconsistent assumptions
- Nir's presentation will give some examples

How people do SRM today

- This is mainly a manual process
  - Smart people thinking of all possible scenarios
  - Using whiteboards and word documents
  - Using fault tree software etc. only to sum things up
  - The process is mostly regulation-based
- Some localized simulation
  - E.g. of drone navigation software + weather
  - E.g. of ice buildup on drone wing
  - But no full-system simulation
- Other techniques
  - Model checking
  - Nancy Leveson's STAMP
  - …
- Boris' presentation will review common practices

Why improve current SRM techniques

- Claim: Current techniques work, but:
  - Costly, somewhat ineffective: Too much manual labor, little automation
  - Don’t allow enough iterations and optimizations
  - Don’t find enough of the risks in new systems:

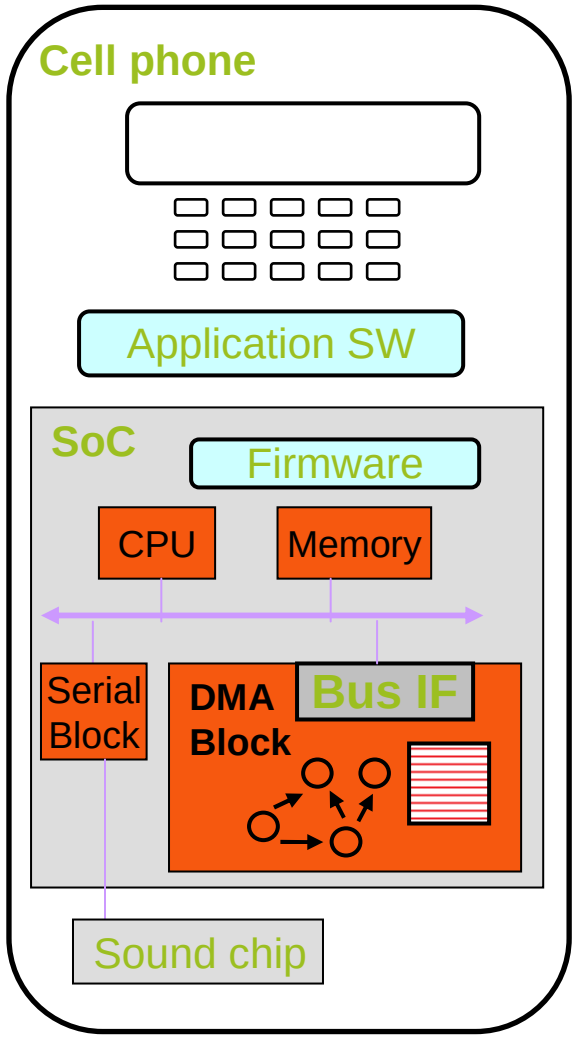
- Newer socio-technical systems are different
  - More complex, with much more software
  - People are more dependent on them
  - More interaction between systems
  - Less time to create them (so regulation cannot keep up):

- Big systems need to be designed faster, because of
  - Everything else is moving so fast
  - Raised expectations (e.g. drones in civil airspaces)
  - Competitive pressure (e.g. very deep sea drilling)
  - Problems created by technology (e.g. geo-engineering)

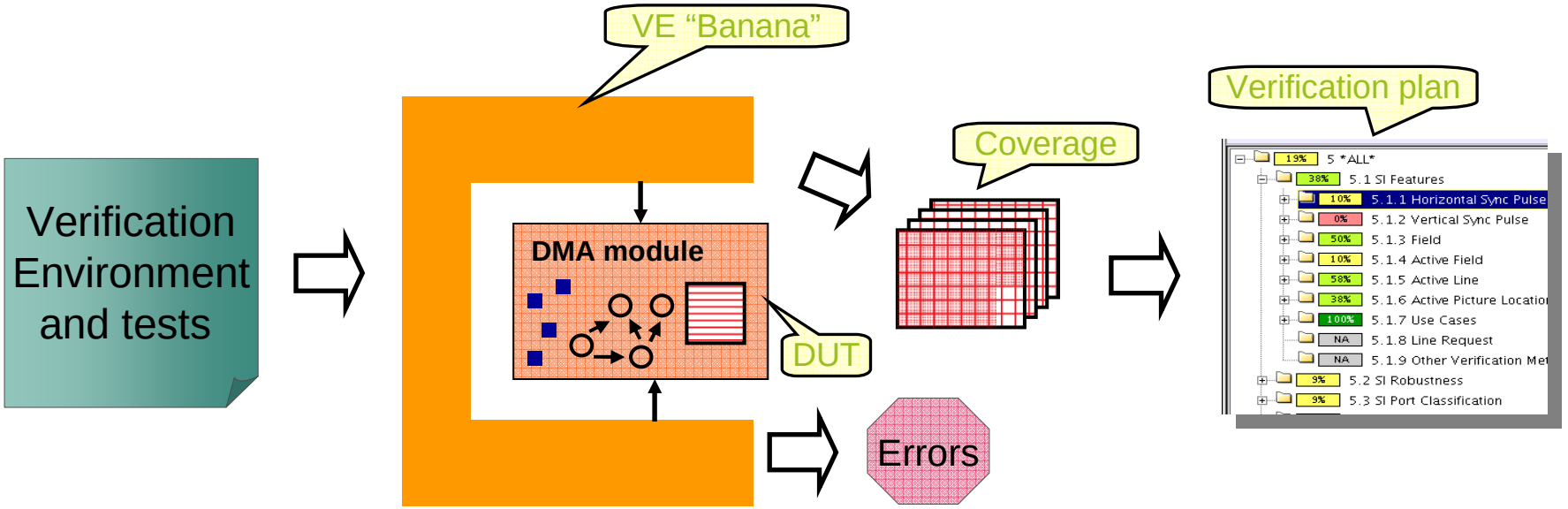
Metric Driven Verification (MDV)

MDV is the verification workhorse for HW blocks / chips

- Some terminology
  - DUT: Device Under Test
  - VE: Verification Environment
  - DVE: DUT + VE
  - VIP: Verification IP
  - HDL: Hardware Description Language
- Chips are written in an HDL
  - The most common: Verilog and VHDL
  - HDLs can be simulated and synthesized
  - Some of the DUT is SW
- Chip verification techniques
  - MDV (most bugs are found that way)
  - Deterministic testing (the old way)
  - Formal verification
  - Post-silicon testing
  - …

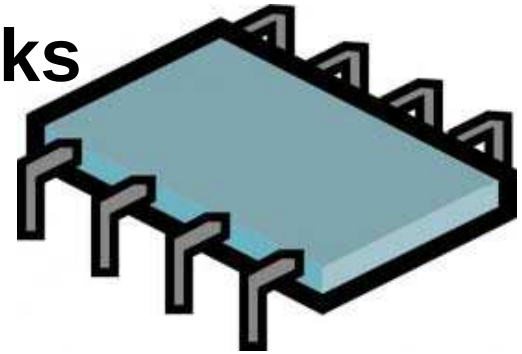


MDV at a glance



- Continuous day-and-night running of many simulations
- Directed-random test generation
- Self-checking
- Coverage tracking
- Clustering of errors
- Reasonable way to do debugging

Why MDV works so well for HW blocks

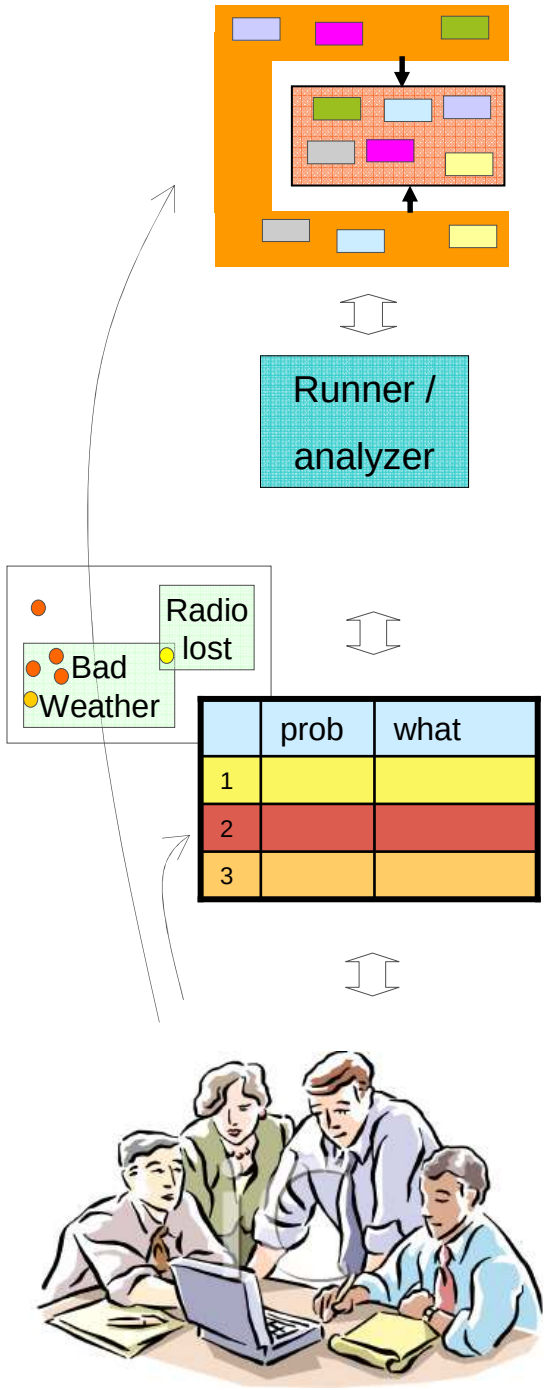


- Cost of failure is very high
  - So projects willing to invest 50%+ in verification
- Easy to simulate the DUT
  - There is a full, simulation-ready model, created for other reasons
  - Model is “just” hardware (and some software)
  - Simulation is getting cheaper quickly
- There are lots of standard protocols
  - So there is a market for pre-created VIPs
- Tools and methodologies exist for 20 years
  - Because of above-mentioned high cost of failure
  - So had time for improvements and adoption
- Moshe’s presentation will discuss possible lessons from chip verification and how they may apply to system engineering

Simulation-based SRM (SSRM)

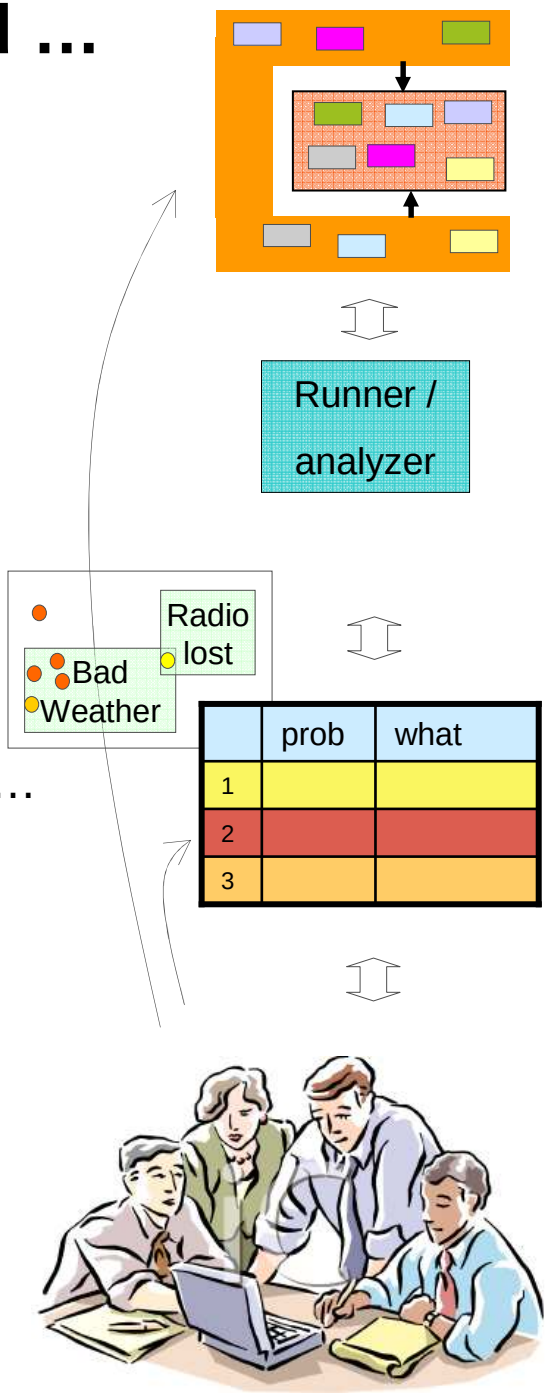
A possible SSRM flow

- Model of DUT and VE is written
  - By multiple people, using multiple notations
- Many simulation runs are executed
  - Over night / weekend
  - Analyzed for “concern clusters”
  - Projected on a “functionality map”
- Presentation, e.g. for drones:
  - Cluster 1: Crash in bad weather with loss of radio
  - Cluster 2: Unable to return to base in bad weather
  - Cluster 3: …
- People investigate / discuss results, then:
  - Fix the design
  - Tweak the model
  - Tweak the clustering directives
  - Declare some clusters as impossible
- Repeat



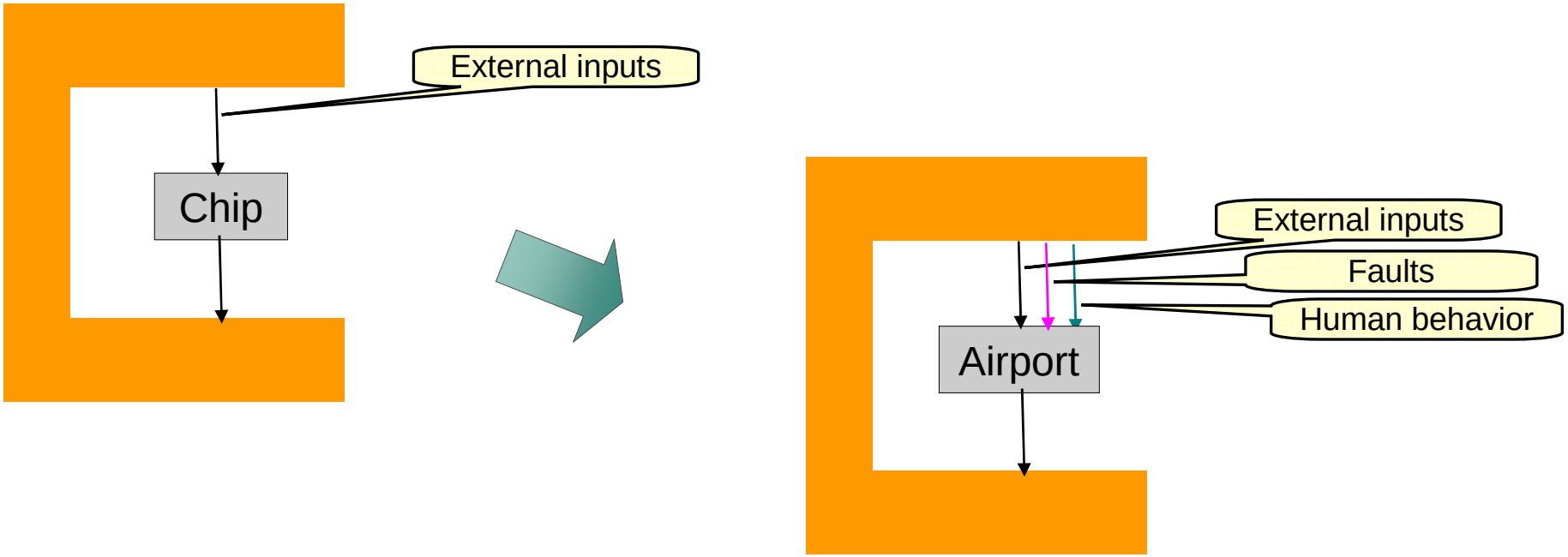
If this “works”, it could change the world …

- Full system simulations could revolutionize “big system” design
  - In planning, tradeoff analysis, risk management
- Could succeed because
  - Simulations are now 1,000,000 times cheaper than in the early 80’s
    - Note: running actual system is also “simulation”
  - Simulation may be the best way to combine all risk sources: Mechanical faults, SW bugs, spec bugs, …
  - We see the beginnings of many of the needed ideas and technologies
- But there are lots of challenges
  - Modeling is the biggest, but there are others
  - Need to tie together experience and ideas from many fields



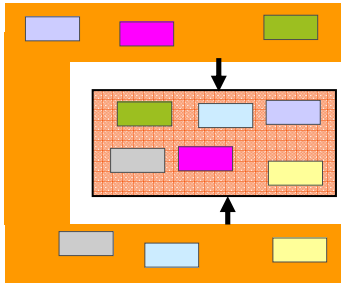
Why MDV is not good enough for SSRM

- In principle, we need to “just” extend MDV slightly:



- But reality is much more complex

SSRM modeling problems



- No full model of DVE exists
  - Chips almost always have a simulation-ready HDL model
  - Big socio-technical systems never model all domains: HW, SW, mechanics, human behavior, infinite variety of the external world, …
  - Need ways to abstract domains and embed them in one simulation
- Need to model man / machine interaction
  - Gennady's presentation will show an example
  - Need also to model human mistakes
- Need new languages to simplify modeling
  - Especially for abstraction and programming by use case
    Assaf's presentation will show one way to do it
- Need to find modeling shortcuts
  - Using the actual hardware / software (once they exist, or previous version)
  - Using training simulators
  - Using real-life info to calibrate simulation
  - Automatic extraction of risks from the web
- Need to model faults and reduced functionality
  - Many bad bugs happen at intersection of multiple failures

Other SSRM issues

- Need to find very rare events
  - Random simulation with “real-life” probabilities will never find them
- Need a mechanized way to estimate risk probability
  - At least very roughly
- Need to connect multiple simulation paradigms / systems
  - Continuous, event-based, agent-based, …
  - Written by multiple people
  - Need to let each person debug using his terms/notations
- Concern of the “echo effect”
  - I.e. “simulation can only tell you about risks you told it”
- No guarantee
  - You can never promise “all risks were found”
  - You need to use coverage, but this also scares people
- “Concerns” are more complex than “errors”
  - It is all a big optimization game

Open Questions

What should SSRM do? (my suggestion is underlined)

- What kinds of "big systems" should we tackle?

  engineered systems ..... living systems

- What should we find out about risks?

  new risk mechanisms ..... probability of known risks ..... risk mitigations

- What should we simulate?

  HW SW mechanics people environment ..... design process ...

- What kinds of risks should we help avoid?

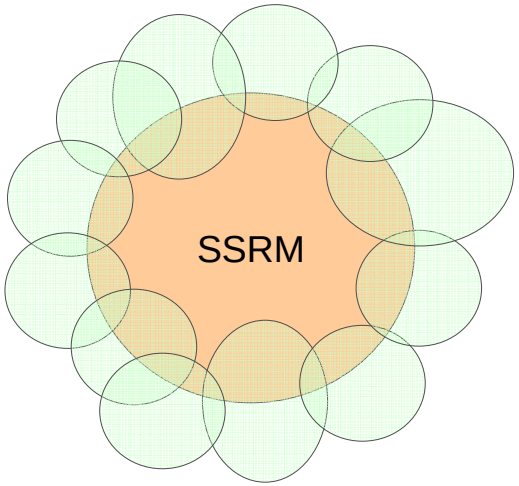
  accident risks ..... project risks ..... bad tradeoff risks ..... security risks

- What project stages should we help in?

  specification design implementation integration usage maintenance

Other questions

- How does SSRM relate to:
  - Model-based development and simulation (e.g. with UML)
  - Agent-based simulation
  - Formal verification
  - Game theory
  - System engineering
  - Monte-Carlo simulation
  - System verification and validation
  - Software testing techniques
  - Operational Risk Management
  - Financial risk



- How can SSRM help system integration
  - Many project spend most time there, never get to full verification
- Who will pay for SSRM
  - Who cares about those risks
  - What's a good industry to start in

Summary

- SRM needs to improve
- MDV-style simulation may be a good way
- Lots of challenges and open questions
- Thank you

Questions?

yoav.hollander@gmail.com