CHALMERS | UNIVERSITY OF GOTHENBURG

DIT045 H17 Requirements and User Experience

Lecture 13: Verification and Validation

Jennifer Horkoff
Email: jenho@chalmers.se

Schedule Proposal

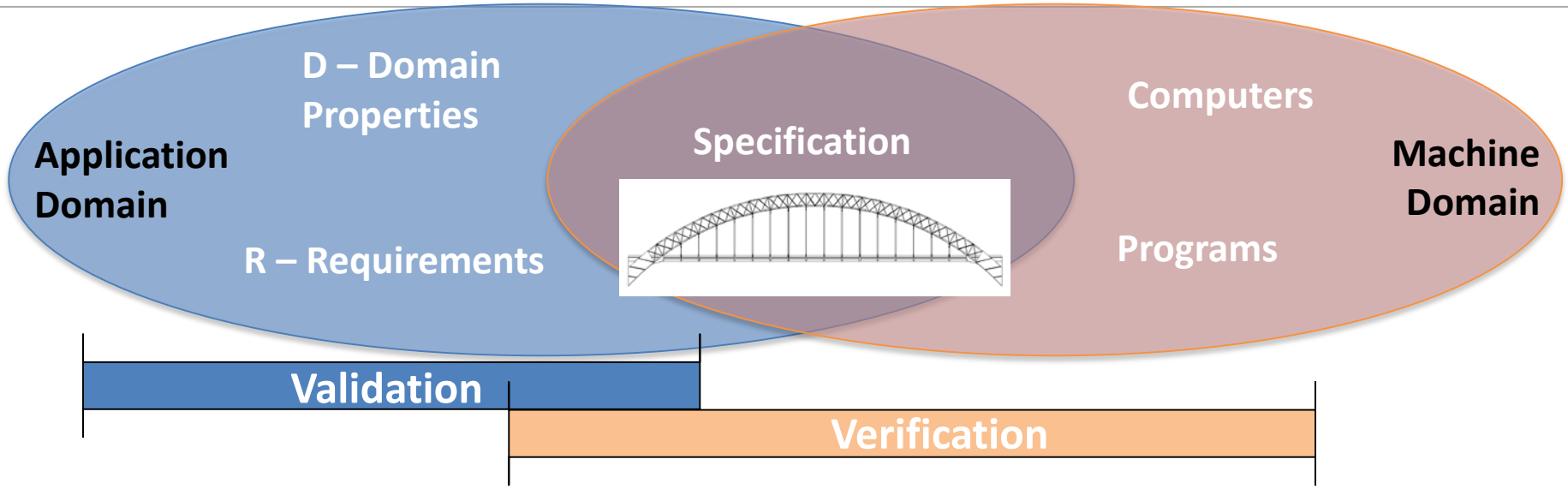
Current:

50	Dec. 11th, 2017	Lecture	RE & UX: Verification & Validation
	Dec. 13th, 2017	Lecture	Course Summary
	Dec. 13th, 2017	Supervision	Group Supervision for A3
51	Dec. 18th, 2017	Review	(Optional) Exam Preparation
	Dec. 22nd, 2017	Assignment	A3: UI Design & UX Evaluation
2	Jan 10th, 2018 (morning)	Exam	Final Exam http://cse.gu.se/english/student/examination

Proposed:

50	Dec. 11th, 2017	Lecture	RE & UX: Verification & Validation
	Dec. 13th, 2017	Lecture	A3 User Studies & Supervision
	Dec. 13th, 2017	Supervision	A3 User Studies & Supervision
51	Dec. 18th, 2017	Review	Course Summary & Exam Preparation
	Dec. 20th, 2017	Lecture	A3 User Studies
	Dec. 20th, 2017	Supervision	A3 User Studies
	Dec. 22nd, 2017	Assignment	A3: UI Design & UX Evaluation
2	Jan 10th, 2018 (morning)	Exam	Final Exam http://cse.gu.se/english/student/examination

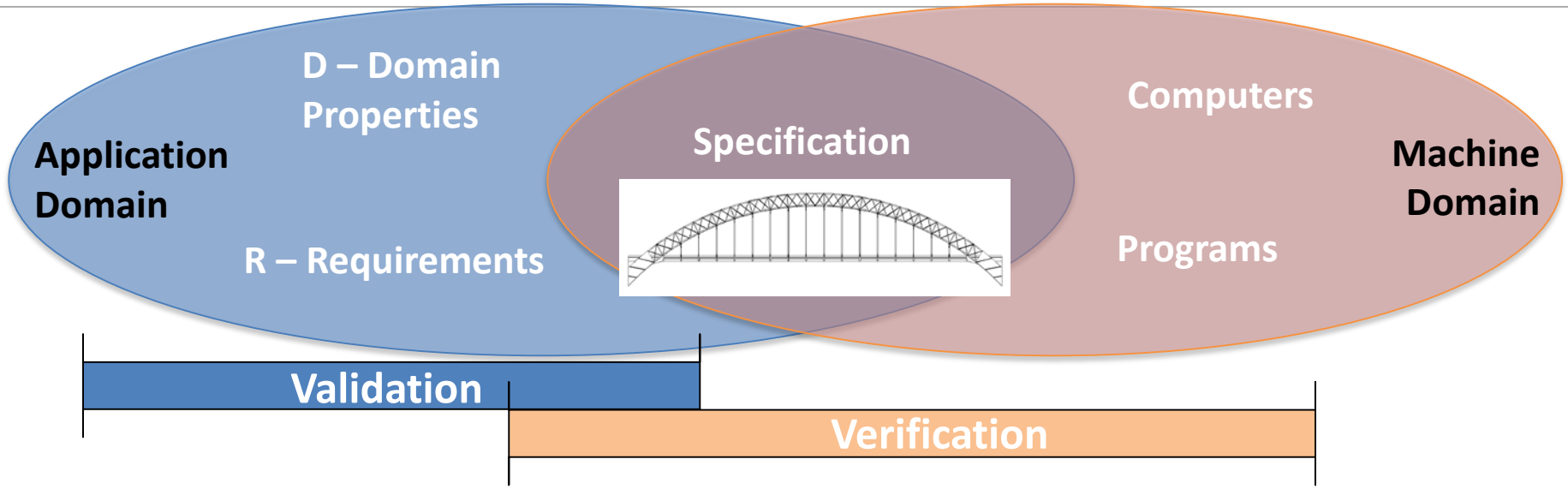
Validation & Verification



- Validation: Are we building the right system?
  - Does our solution solve the real problem?
  - Did we account for the most important stakeholder needs?
- Verification: Are we building the system right?
  - Does our design meet the requirements?
  - Does the system do what we say it would do?
  - Are our requirements representations consistent?

(Zave & Jackson, Easterbrook)

Validation & Verification Criteria

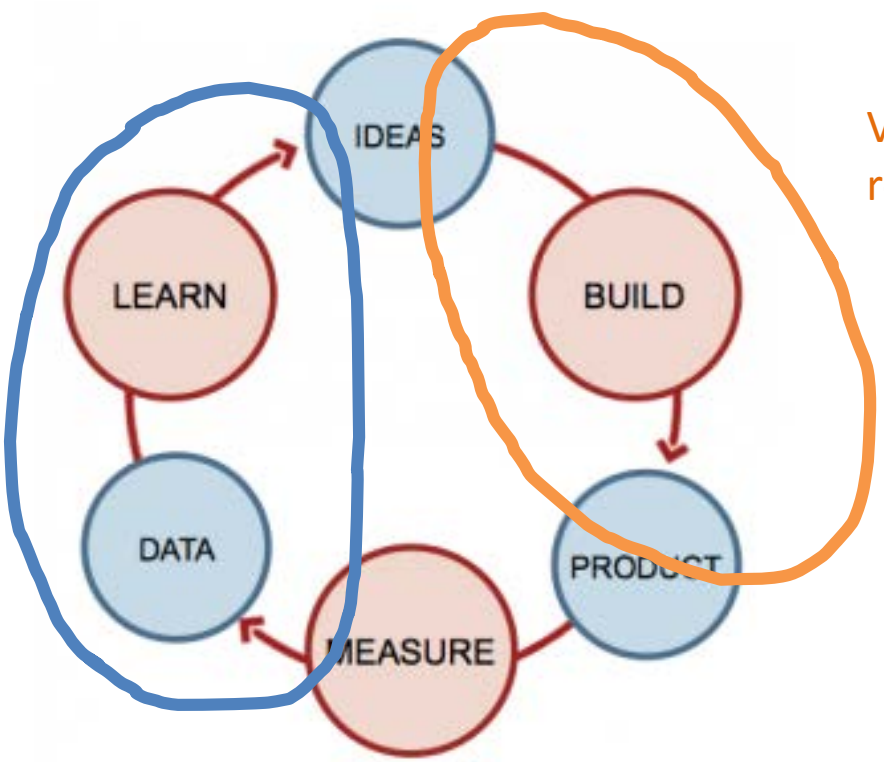


- Example Verification Criteria
  - The software running on a computer satisfies the specification
  - The specification, given the domain properties, satisfies the requirements
- Example Validation Criteria
  - Did we discover all the important/essential requirements (for the MVP, for example)?
  - Did we discover all the important domain properties/assumptions?

(Zave & Jackson, Easterbrook)

Feedback Loop

- Recall Build-measure-learn feedback loop from Lean Startup
- How do validation and verification fit in?

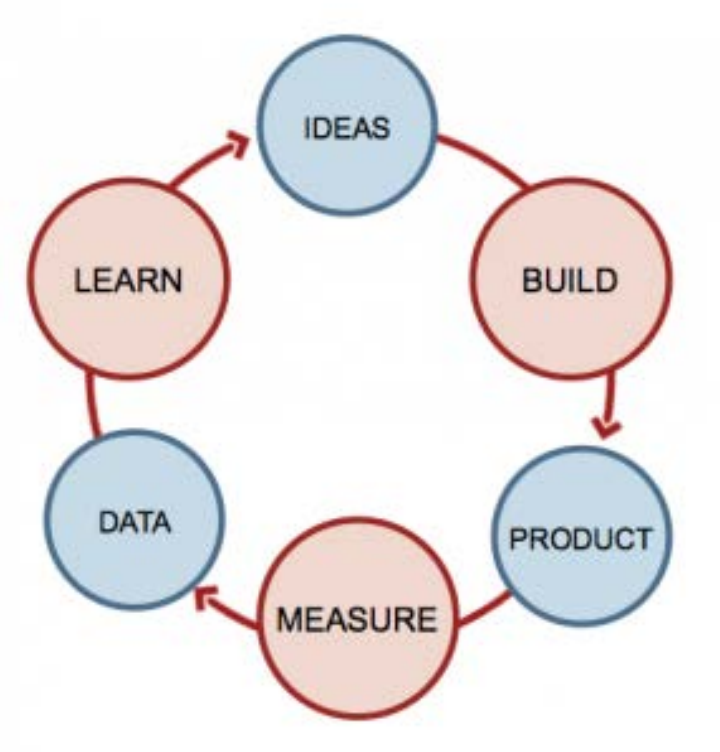


Validation is roughly here

Verification is roughly here

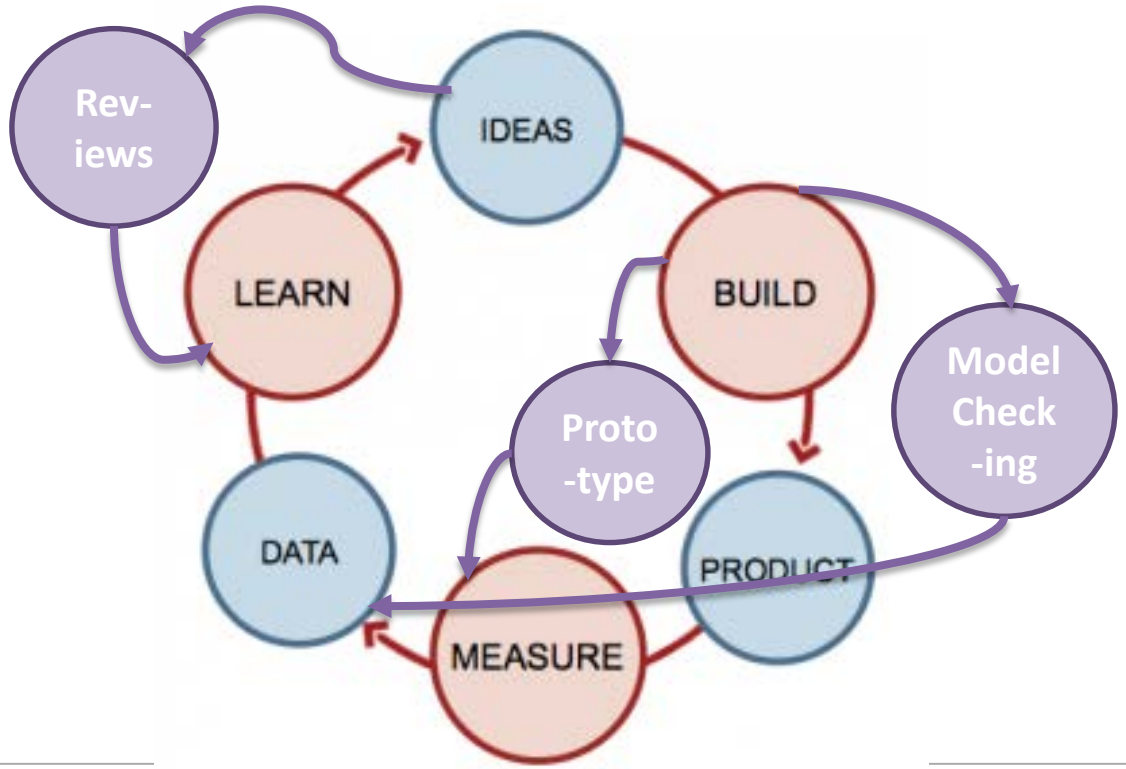
Feedback Loop

- Challenge: the full loop requires a finished product
- Solution 1: MVP, work in a agile way, have product as soon as possible
- Solution 2: Emulate the solution, test the emulation



Shortcuts in the inquiry cycle

- Emulate the solution:
  - Prototypes (see Lecture 8)
  - Model checking
  - Reviews, walkthroughs, inspections



(Easterbrook & Campbell)

Model Checking

- Evaluating, Validating, and Analyzing the requirements models before listing the requirements or designing the system
- Checking for:
  - Errors like logical inconsistencies
  - Incompleteness
  - Confusions and misunderstandings
  - Satisfied goals
  - Process bottlenecks
  - Etc.
- What you can check for depends on what kind of models you create
  - The more formal and detailed your model, the more detailed your analysis can be → More upfront effort

Model Checking V&V

- Verification:
  - Is the model well formed?
  - Are parts of each model consistent with each other?

- Validation:
  - Formal model checking
    - Does this property hold?
    - Will the system ever reach this state?
  - What if analysis
    - Reasoning about the consequences of particular requirements choices
  - Simulation
    - Where are the bottlenecks
    - Best for process models (we didn't really cover them, closest was customer journey map)

(Easterbrook & Campbell)

Model Well-formedness

- Does the model follow the rules for this model
  - If not, the model is not capturing the information intended by the designers of the modeling language
  - Possible that the modelers do not have a common understanding of what they are modeling
  - Someone else would have trouble reading and interpreting the model later

- Examples:
  - Context diagram: system actor in center, all flows are data
  - Use case: system boundary, actors are actually actors, use cases are actions/functions, part of boundary between stakeholders and systems
  - Goal model: correct type of elements used, correct types of links used between elements, correct types of links used inside/outside actors

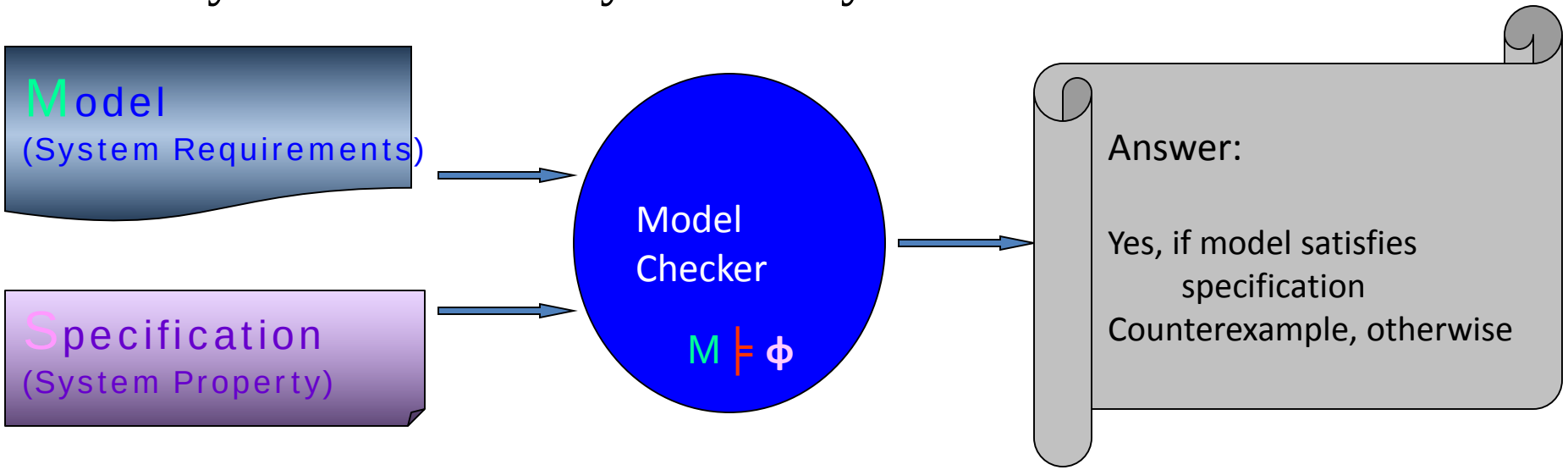
Recall: Approximate Model Mappings

Context Diagram	Use Cases	Goal Models
System actor	System boundary	System actor
Other actors	Actors	Actors
Inputs/Outputs	Roughly map to use cases	Dependencies
	Use Case	(Usually) Task
		Qualities
		And/Or Refinement
		Contribution

- This is a form of verification for the requirements models
  - Inconsistencies = misunderstandings, incompleteness, etc.

Formal Model Checking

- We don't do this in this course
- Very useful for safety critical systems

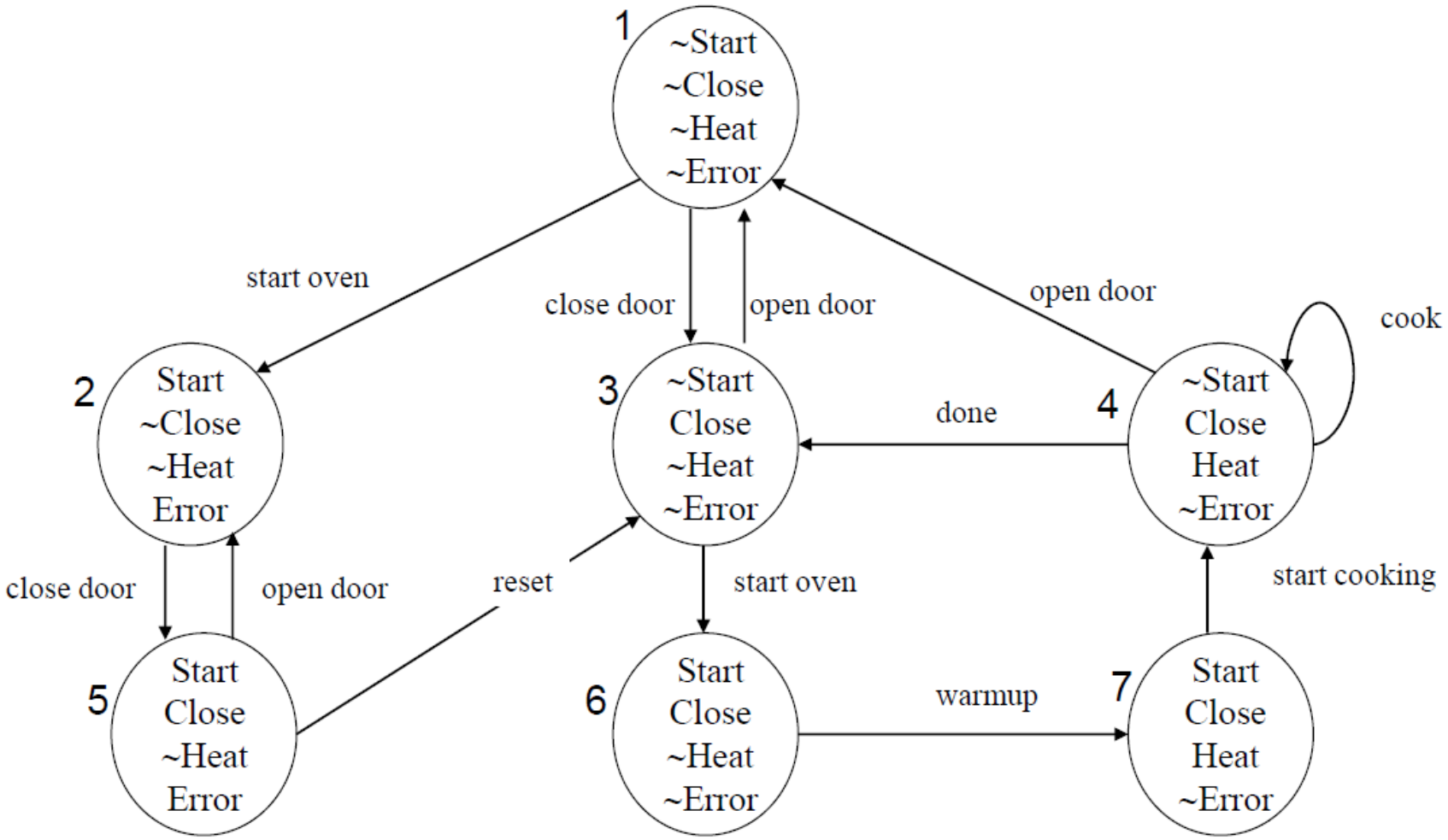


For increasing our confidence in the correctness of the model:

- Verification: The model satisfies important system properties
- Debugging: Study counter-examples, pinpoint the source of the error, correct the model, and try again

(Lawrence Chung)

Model Checking Example



http://www.dsi.unive.it/~avp/14_AVP_2013.pdf (Edmund M. Clarke, Jr.)

Model Checking Properties

- We would like the microwave to have the following properties (among others):
- – No heat while door is open
  - **AG**(*Heat* → *Close*):
- –If oven starts, it will eventually start cooking
  - **AG** (*Start* → **AF** *Heat*)
- –It must be possible to correct errors
  - **AG**(*Error* → **AF** ¬ *Error*):

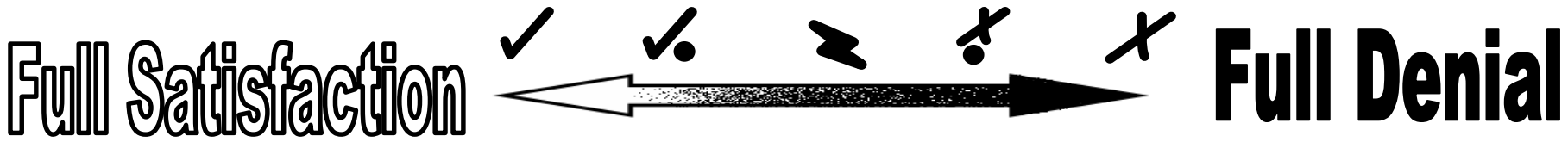
- Does it? How do we prove it?
  - Answer: model checking

http://www.dsi.unive.it/~avp/14_AVP_2013.pdf

Goal Model Analysis

- Goal models can be assigned formal semantics (meaning)
- Analysis procedures can answer questions, e.g.,
  - What if a particular alternative is selected? How does this effect goals and actors in a model?
  - If I want to satisfy a (set of) goal(s), what alternatives should I select? What tasks should I implement?

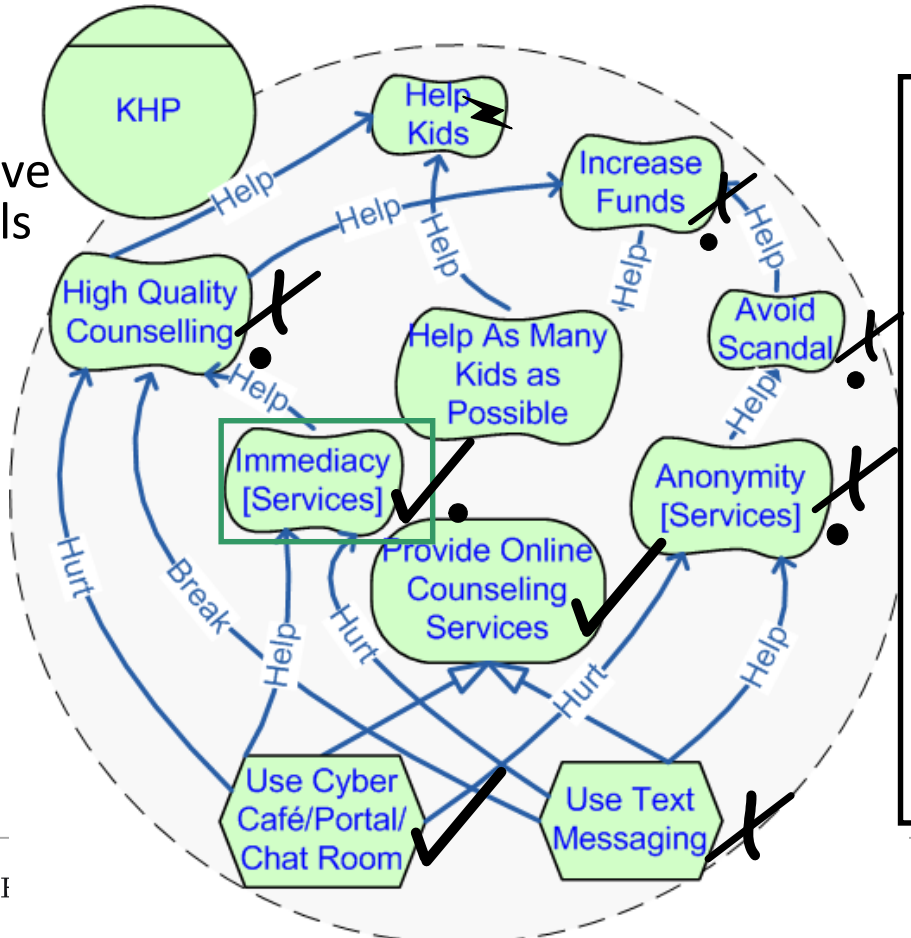
- Tradeoff between information and precision



Reasoning Example Procedure

- Evaluation based on an analysis question:
  - If the Organization implements Chat Room, but not Text Messaging, what effect will this have on goals?
- Place Initial Labels reflecting Analysis Question

- Propagate labels
- Resolve labels
- Iterate on the above steps until all labels have been propagated
- Analyze result



Human Intervention

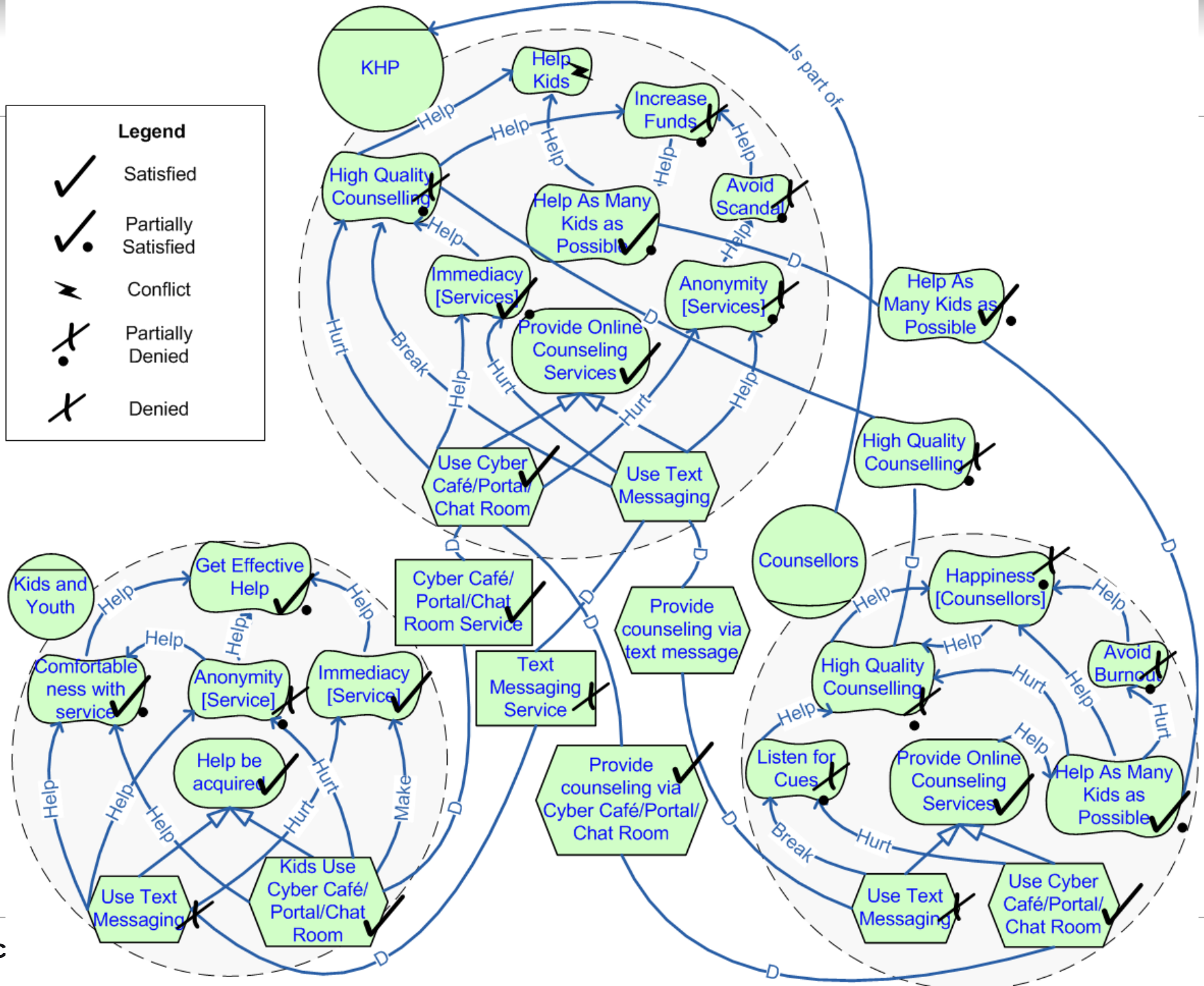
Immediacy Receives the following Labels:

Partially satisfied from Chat Room ✓

Partially satisfied from **Text Messaging** ✓

Select Label…

Select partially satisfied


Legend
Satisfied
Partially Satisfied
Conflict
Partially Denied
Denied
KHP
Help Kids
Increase Funds
High Quality Counselling
Help As Many Kids as Possible
Avoid Scandal
Immediacy [Services]
Anonymity [Services]
Provide Online Counseling Services
Use Cyber Café/Portal/ Chat Room
Use Text Messaging
Is part of
Help As Many Kids as Possible
High Quality Counselling
Counsellors
Happiness [Counsellors]
Avoid Burnout
High Quality Counselling
Listen for Cues
Provide Online Counseling Services
Help As Many Kids as Possible
Use Text Messaging
Use Cyber Café/Portal/ Chat Room
Kids and Youth
Get Effective Help
Comfortable ness with service
Anonymity [Service]
Immediacy [Service]
Help be acquired
Use Text Messaging
Kids Use Cyber Café/ Portal/Chat Room
Cyber Café/ Portal/Chat Room Service
Text Messaging Service
Provide counseling via text message
Provide counseling via Cyber Café/Portal/ Chat Room
Help
Hurt
Break
Make
D

Reviews, Walkthroughs & Inspections

- "Management reviews"
  - E.g. preliminary design review (PDR), critical design review (CDR), …
  - Used to provide confidence that the design is sound
  - Attended by management and sponsors (customers)
  - Often just a "dog-and-pony show"
- "Walkthroughs"
  - developer technique (usually informal)
  - used by development teams to improve quality of product
  - focus is on finding defects
- "(Fagan) Inspections"
  - a process management tool (always formal)
  - used to improve quality of the development process
  - collect defect data to analyze the quality of the process
  - written output is important
  - major role in training junior staff and transferring expertise

(Easterbrook & Campbell)

Reviews, Walkthroughs & Inspections

- Walkthrough definitions are not widely agreed!
- Other terms used:
  - Formal Technical Reviews (FTRs)
  - Formal Inspections
- "Formality" can vary:
  - informal:
    - meetings over coffee,
    - regular team meetings,
    - etc.
  - formal:
    - scheduled meetings,
    - prepared participants,
    - defined agenda,
    - specific format,
    - documented output

(Easterbrook & Campbell)

Benefits of Formal Inspections

- Formal inspection works well for programming:
  - For applications programming:
    - more effective than testing
    - most reviewed programs run correctly first time
    - compare: 10-50 attempts for test/debug approach
  - Data from large projects (note: data is old now)
    - error reduction by a factor of 5; (10 in some reported cases)
    - improvement in productivity: 14% to 25%
    - percentage of errors found by inspection: 58% to 82%
    - cost reduction of 50%-80% for V&V (even including cost of inspection)
- Effects on staff competence:
  - increased morale, reduced turnover
  - better estimation and scheduling (more knowledge about defect profiles)
  - better management recognition of staff ability
- These benefits can also apply to requirements inspections

(Easterbrook & Campbell)

Structuring the Inspection

- Checklist
  - uses a checklist of questions/issues
  - review structured by issue on the list
- Walkthough
  - one person presents the product step-by-step
  - review is structured by the product
- Round Robin
  - each reviewer in turn gets to raise an issue
  - review is structured by the review team
- Speed Review
  - each reviewer gets 3 minutes to review a chunk, then passes to the next person
  - good for assessing comprehensibility!

(Easterbrook & Campbell)

Why Use Inspections

- Inspections are very effective
  - Code inspections are better than testing for finding defects
  - For Specifications, inspection is all we have (you can't "test" a spec!)
- Key ideas:
  - Preparation: reviewers inspect individually first
  - Collection meeting: reviewers meet to merge their defect lists
  - Log each defect, but don't spend time trying to fix it
  - Reviewers learn from one another when they compare their lists
  - Defect profiles from inspection are important for process improvement
- Wide choice of inspection techniques:
  - What roles to use in the meeting?
  - How to structure the meeting?
  - What kind of checklist to use?

(Easterbrook & Campbell)

Summary

- Validation checks you are solving the right problem
  - Prototyping - gets customer feedback early
  - Inspection - domain experts read the spec carefully
  - Formal Analysis - mathematical analysis of your models
  - ...plus meetings & regular communication with stakeholders

- Verification checks your engineering steps are sound
  - Consistency checking - do the models agree with one another?
  - Traceability - do the design/code/test cases reflect the requirements?

(Easterbrook & Campbell)

Questions?