



Matti Vuori, 20.6.2013

RATA project report

Architecture of models in testing – how models of various abstraction levels relate to each other

Contents

1. Introduction	2
2. Generic architecture of the models	2
3. Model types at various levels, for interactive systems	4
APPENDIX: What are models good for and why things should be “modeled” more?	10



1. Introduction

As the technology and practices of model-based testing are developed further and are used in more varied ways, we need to understand a wider array of “models” that can be utilized in various abstraction levels of the system and the testing. This paper presents a list of such models and some ideas of the relations of the models.

Note that the descriptions of this paper mostly deal with automatic test generation (usually model-based testing). For exploratory testing, the same information is used, but in more non-formal ways.

2. Generic architecture of the models

First, what is a model – in this context? We could define it as: “A presentation of a system or an element of a system that mimics the characteristics of it that someone is interested in”. The characteristics may include:

- Defining elements and their relations.
- Structure.
- Behavior.
- Visual appearance – a drawing, a mockup.

Of course, many of the artifacts used in designing and documenting a system are a “model” in varying degrees, by fuzzy logic... Still, we need to realize that there is more variety to models and modeling than the traditional things that are “labeled as models”.

There are various uses for the models. Some of the models can be used directly to test generation, but some of them will guide just test planning and test execution. And sometimes models are validated by testing. Some models just help in sharing understanding between stakeholders and inside the team about what they are dealing with an aiming to achieve. Those models build and align the mental models of people – which is the most important thing in development and quality assurance – but outside the scope of this paper.

A rough figure of the “positioning” of models is shown in Figure 1.

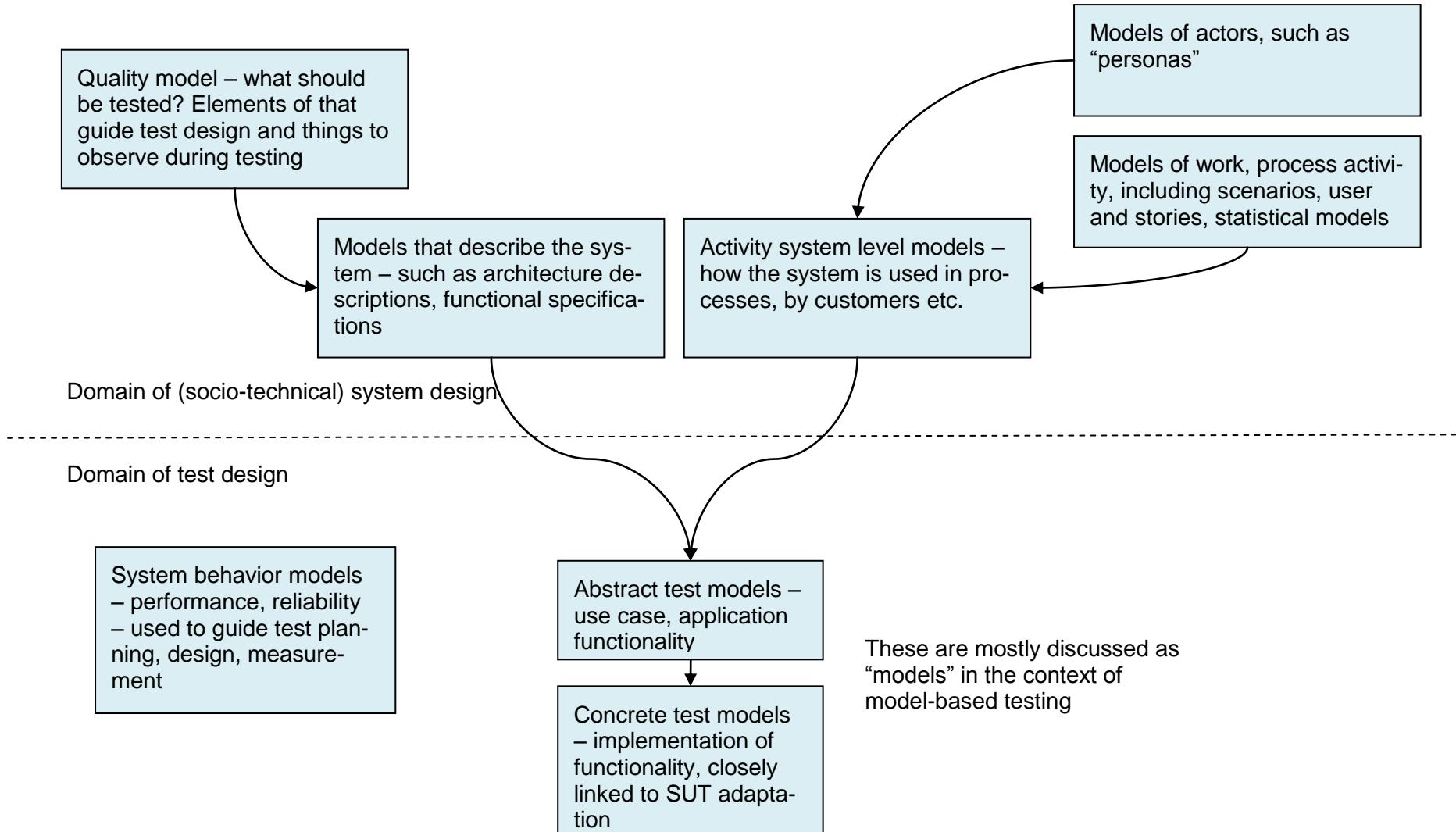


Figure 1. Architecture of models in testing (simplified conceptual presentation)



3. Model types at various levels, for interactive systems

In Table 1 we present a long list of “model-kind-of-artifacts”, aiming to show how they relate to testing. The descriptions in the table are necessarily simplistic and some types or models are more relevant in some domains and cultures than some others. Most likely, some types are missing from the table too.

Table 1. Various model types at various levels, for interactive systems.

Level	Model	Purpose and content	Model type	Usage in testing	Adaptation (relation with lower level test assets)
Life of an individual	Persona, archetype definition http://en.wikipedia.org/wiki/Persona_%28user_experience%29	What typical users are like	Descriptive, static Structured, detailed description of an imaginary individual	Test planning, configuration Sets a frame for a test (testing for this type of users) Direct basis for usability testing	–
	User (consumer) segments http://en.wikipedia.org/wiki/Market_segmentation	Division of users into classes based on their demographic characteristics or relation to the system	Descriptive, static Classification model	One segment sets a frame for a test All segments considered in analysis of observations & defects	–
	Daily activity for consumer	Daily activity of an individual	Descriptive, dynamic What activities and in what order) and statistical (how often)	Test planning (scenarios, stories, use cases) Gives priorities for testing of tasks, user stories Frame for usability testing & maturity testing Customer risk analysis	Test selection Test run parameters



Level	Model	Purpose and content	Model type	Usage in testing	Adaptation (relation with lower level test assets)
	Work profile for an occupation	Daily activity of an individual	Descriptive, dynamic What activities and in what order) and statistical (how often)	Test planning (scenarios, stories, use cases) Gives priorities for testing of tasks, user stories Frame for usability testing & maturity testing Customer risk analysis	Test selection Test run parameters
“Business” processes and tasks	Process flow	End-to-end description of a process flow; of activity or of data	Activity, dynamic Process specification languages	Test planning Test design Test execution Business risk analysis	Via test execution system – may be a batch run of tests, end-to-end tests or something else
	System activity metrics	Description of amount, volume, duration etc. of how system elements are used. See also Markov model	Descriptive, dynamic	Test planning Test design Test execution	Via test execution system
	Use(r) scenario	Description of why, what and how a user acts to achieve something	Activity, dynamic Freeform description	Use for usability testing used as such For MBT, extract use cases	–
	User story http://en.wikipedia.org/wiki/User_story	1) Describes what a user does in an activity; 2) description of a requirement	Activity, dynamic Free-form or structured description; scope may vary	For exploratory testing, used as such For usability testing, used as part of scenario or as such For MBT, transform to a more technical model	Extract low-level use cases & implement them



Level	Model	Purpose and content	Model type	Usage in testing	Adaptation (relation with lower level test assets)
	Behavioral Design specification (BDD) http://en.wikipedia.org/wiki/Behavior-driven_development#Behavioral_specifications	Description of requirement based on user behaviour in a role One form of a user story	Activity, dynamic (mostly) Structured textual description (As <a role> I want to <do something> (simplified))	Input for BDD testing tools in agile acceptance testing	BDD tools
	Task definition	“Formal” definition of a user task	Activity, dynamic Instruction for user	For manual testing (incl. usability testing), can be used as such For MBT, extract use cases	–
	Use case http://en.wikipedia.org/wiki/Use_case	Structured definition of a phase in process	Activity, dynamic Structured, descriptive; start state, end state, actions inbetween	Basis for modeling applications Basis for set of test cases	Lower level models



Level	Model	Purpose and content	Model type	Usage in testing	Adaptation (relation with lower level test assets)
System / product concept	Definition of product concept (operational concept)	Description of the main defining characteristics of the system, such as purpose, users, use environment, rough architecture, defining technologies, risks, key benefits	Desctiption, static List, mindmap Drawings NABC	Promotes shared understanding of the thing under development Absolutely essential to understand	(Lower level descriptions of usage, technical system)
Technical system models	Quality model http://en.wikipedia.org/wiki/ISO/IEC_9126	Description of the main quality attributes (requirements) of the system (See ISO 9126, 25010)	Descriptive, static	Overall test planning Selection of product metrics	–
	Architecture diagrams http://en.wikipedia.org/wiki/Software_Architecture_styles_and_patterns	Showing the system architecture from various viewpoints	Descriptive / activity; static / dynamic Varies	Informal use for test planning, test design Coverage assessment	–
	Configuration model	A look into the architecture from the viewpoint of configuring the system for customers and for tests	Descriptive, static	Generation of proper SUT for testing Configuring of tests (including test models) for test Preferably automatically through parametrization	Through test management & test environment management systems
	System lifecycle model	Describes the phases in system's lifecycle at user site from purchase to disposal	Descriptive, dynamic Linear	Test planning Test design	(Lower level plans)
	System performance model	Description of how the various system elements affect the system's performance	Descriptive, dynamic Empirically created or validated	Test planning Test design Evaluation of test reports and understanding observations (Especially for performance testing)	–



Level	Model	Purpose and content	Model type	Usage in testing	Adaptation (relation with lower level test assets)
	System reliability models	Description of factors affecting reliability of the system, including causal relations between things that happen	Descriptive / activity; static / dynamic Cause-consequence diagrams, fault trees, FMEA models	Test planning Test design Defect injection	–
Application design models	UI structure model	Hierarchical description of UI elements Part of UI specification	Descriptive, static Hierarchy HTML demo	Checklist for testing (coverage)	–
	UI design model	Presentation of content, layout of displays; description of control techniques, technologies	Visual, static / dynamic Prototype HTML demo	Information for test design	–
	Model-based design model http://en.wikipedia.org/wiki/Model-based_design	Model of application behaviour	Activity, dynamic State model	Simulation of application behaviour Generation of test cases Generation of test models	Modeling language Testing language scripts
	Model for UI element	Description of an UI element: type, inputs	Descriptive / activity; static / dynamic	Automatic test case generation Automatic test data generation	Test cases Test models
	Model of internal interaction of system elements	Description of how system components interact, for example in a protocol	Activity, dynamic UML diagrams or similar, for example: Sequence diagrams State diagrams	Test as such with demanding use cases and test data	Implement control with any test tools



Level	Model	Purpose and content	Model type	Usage in testing	Adaptation (relation with lower level test assets)
Application test models	Test model of an application http://en.wikipedia.org/wiki/Model-based_testing	Describes the functioning of an application in an environment, for testing purposes	Activity, dynamic State machine or similar; may consist of several sub-models	Use directly for functional testing, using an adapter	Adapter layer
	Test model of an application suite	Describes the functioning of an application and interactions between application in an environment, for testing purposes	Activity, dynamic Several state machines or similar	Use directly for functional testing, using an adapter	Adapter layer
	Markov models of activity	Empirical data of what probability applications and functions are used	Statistical model	Give priorities for functions and state transformation in testing Data to guide test runs	Via test models



APPENDIX: What are models good for and why things should be “modeled” more?

