|  |
| --- |
|  |
| **Title\*:** | TDL Roadmap Brainstorming Call Report |
|  |  |
| from **Source**\*: | Institut für Informatik |
| Contact: | Philip Makedonski  |
|  |  |
| input for **Committee**\***:** | MTS |
|  |  |
| Contribution **For\*:** | Decision |  |  |
|  | Discussion |  |  |
|  | Information | **X** |  |
|  |  |
| Submission date**\***: | 2020-04-24 |
|  |  |
| Meeting & Allocation: | **MTS-MTS#80-Online** -  |
| Relevant WI(s), or deliverable(s): |   |
|  |

**ABSTRACT:***Summary from the TDL Roadmap Brainstorming Call*

* Attending
	+ Emmanuelle (ECT), Andreas (AU), Finn (FK), Martti (MK), Philip (PM), Jens (JG), Gusztav (GA), Daniel Ilie Georghe-Pop (DIGP), Michele (MC)
* Agenda
	+ Background
		- STF 577 status update
		- Roadmap available at MTS(20)080001
			* AU: Focus on external factors:
				+ RESTful API testing
				+ AI testing
				+ User requests
			* MC: Parameterised TPs and TDs
				+ Recommendation for REST context in particular for TPs
				+ See also resolution for TDs
				+ To be agreed offline before MTS#80
			* AU: Test coverage in particular for API specifications
				+ Methodology and guidelines for capturing and managing requirements, tool support
				+ STF 576 can provide initial input for future work on TDL
			* MK: Support from TTCN-3 WG for standardised mapping to HTTP
				+ JG: Submitting a CR via mantis is required, discuss internally to prepare contribution
		- Latest TDL WG meeting minutes are available in MTS(20)079035
	+ Potential directions:
		- Indentation-based syntax (see also BZ issue) with high priority
		- **REST API specification and testing needs (see separate note, included below)**
		- **Capturing and managing requirements**
		- Generalise behaviour specification (see separate note, included below)
		- **AI Testing initiative**
			* AU: Wide scope, focus on autonomic / self-adaptive systems
				+ Methodology (see related WIs)
				+ Test specifications, execution
				+ Testing needs to be adaptive, beyond predefined test sequences

Tester adapts based on input from SUT and decides what to do next

Design in TDL

* + - * + Ericsson experiences in adaptive testing

GA: Early stages at the moment, potential input in the future

* + - * AU: Start with requirements solicitation
			* PM: Early stage, to be kept in the pipeline for subsequent activities
		- NWM integration
		- Guidelines on integration of repositories and cross references of machine-readable assets, considering ownership, responsibilities, procedural frameworks
	+ Brainstorming details (see separate notes)
	+ Next steps
		- Reach out to (IoT) TST WG
			* DIGP: at the moment TDL is serving the work sufficiently well
		- Direction(s)
			* Short term focus / mid / long term outlook
			* Review uploaded roadmap offline
		- Initiatives
			* STF / TTF proposal to facilitate continuity of evolution
			* Parallel tracking of related activities
		- MTS / WG meeting contributions
			* Roadmap website (see also MTS / MTS-TDL notes)
			* ToR drafting
		- Deadlines
			* **PM: Share notes after the meeting (including related notes)**
			* **PM, AU: Update roadmap by next week**
			* **PM, AU: Start drafting ToR before MTS#80**
				+ Determine scope, resource estimates first
				+ To be approved by RC in June
			* **PM, AU: ToR needs to be finalized by August**

The following are related working notes which continue to evolve over time, copied her for reference:

* Mapping to (from?) OpenAPI for restful specification and testing
	+ In particular expressing flows for which OpenAPI is not well suited
	+ Investigate how resources and URIs can be mapped to data (types) and components for example
		- Pilot effort needed to collect input for potential funded initiative
		- Can potentially be integrated with output from STF 576 and NWM group
	+ Define a guideline / standardised mapping if necessary
	+ Add CR once this is more matured
* TDL for OpenAPI in broader terms
	+ Mapping / importing data and datatypes from OpenAPI
	+ Deriving TDL TDs and TPs from OpenAPI
		- Is that even possible?
		- What is the extent of the information that needs to be extracted?
			* Data types, values, instances
			* Requests, responses can be turned into when/then statements in TPs, interactions in TDs, with manual annotation / refinement embedded in them or defined externally so that an automated syncronisation does not impact the manual additions
	+ Derivation of executable TTCN-3 in a standardised manner
		- **Would also be a good idea to define standardised mapping to HTTP for TTCN-3**
		- Contribution from MEC STF
			* Headers
			* Payloads (JSON-schema-based)
			* Clear definition and recommendation for mapping
			* Tool support
	+ Data handling and mapping
		- Hiding members in derived types might be needed for PATCH operations
		- Data types describing alternative data types or combinations of data types
		- Need to investigate whether and how this needs to be supported within TDL
* Generalised behaviour specification
	+ Currently test descriptions are the main behavioural unit
	+ We may need a generic unit that captures a more open notion that can be used in broader context and then embedded in a test description
		- A test description would then become one possible container
		- This way it can be more convincing during specification (unless TDD is used)
	+ Does this stretch TDL too far beyond its scope?
		- Focus was initially on testing
		- Can still be a useful for broader purposes and audiences, however it needs a different argumentation and in particular how it compares to MSCs and UML sequence diagrams
		- Need to have better understanding of what is stopping users from adopting it - if it is just the naming, it shouldn't be too much of a concern