kompozition_
smarter together

2024

BUILD TRUE DIGITAL THREADS

Kompozition enables the development, maintenance and querying of a single, continuous and integrated Digital Thread defining your system and it's artefacts.

Who is Kompozition

Kompozition is an industry leading MBSE company championing the adoption of Digital Engineering practices and their benefits to organisations.

The Kompozition team is motivated by the need to reduce cost and risk on large programs of work by enhancing the way people work and communicate using their platform and methodology.

Based in Australia, our team comes with decades of experience delivering large complex systems to both Australian and International organisations.

RISK MITIGATION

Reduce risk to your system development, acquisition and sustainment through an integrated Digital Twin

ARTIFICIAL INTELLIGENCE

A proprietary AI engine connects data to accelerate the engineering, design and testing process

SYSTEM ANALYSIS

Analyse the impact of changes and integration of systems and interfaces before implementation

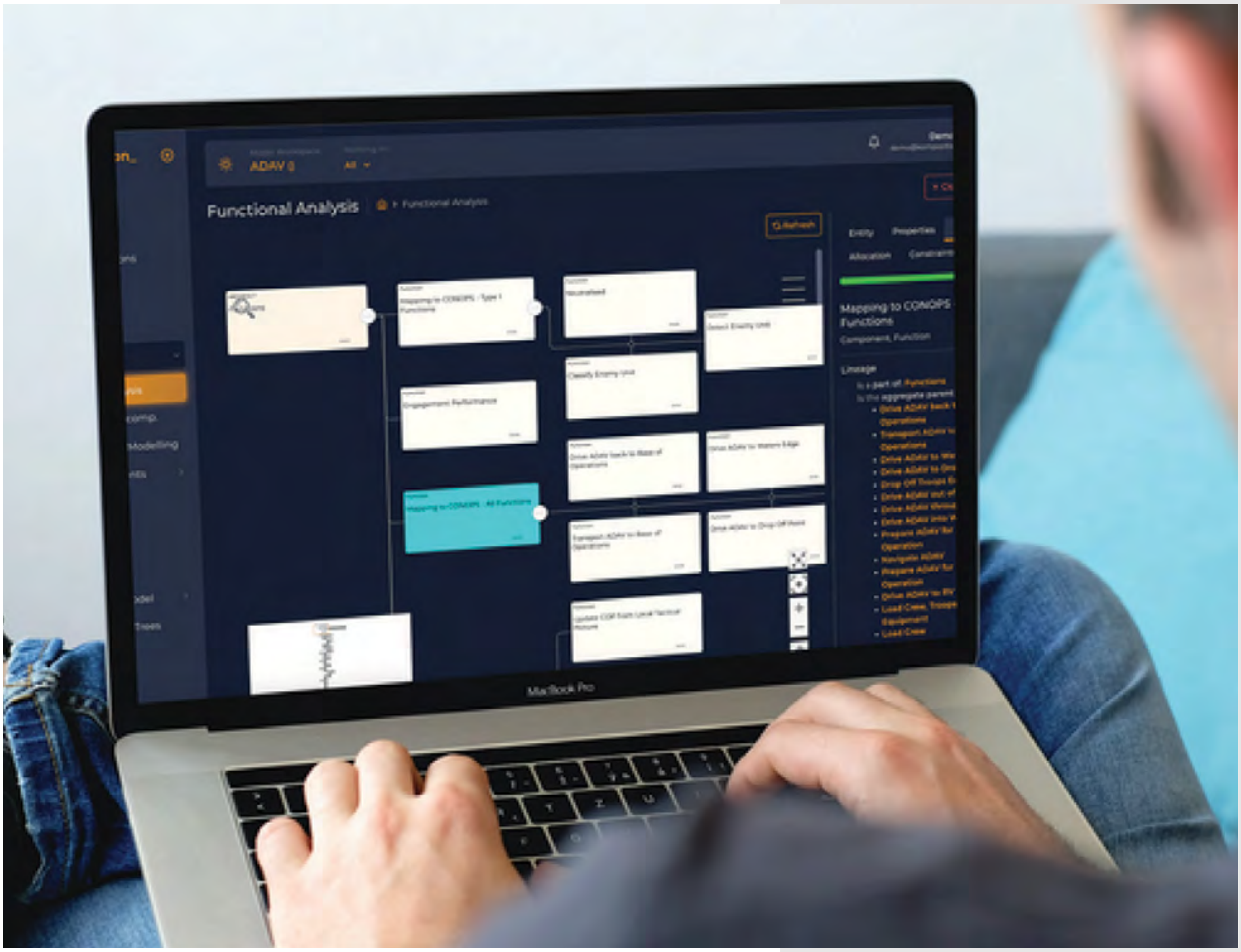
Our unique Digital Systems Engineering Platform has through-life traceability and connectivity helping organisations accelerate and de-risk technical projects and maintain the systems they deliver.

kompozition_
smarter together

Adoption of Digital Engineering

Kompozition enables the creation of digital data and digital connectivity that can be leveraged to integrate a system's design, development, delivery, and full life cycle support into a more agile, higher-performance product value stream than would be possible using a traditional Systems Engineering framework This is achieved due to Kompozition providing native support for Digital Engineering.

Model Based Systems Engineering with through-life traceability and connectivity



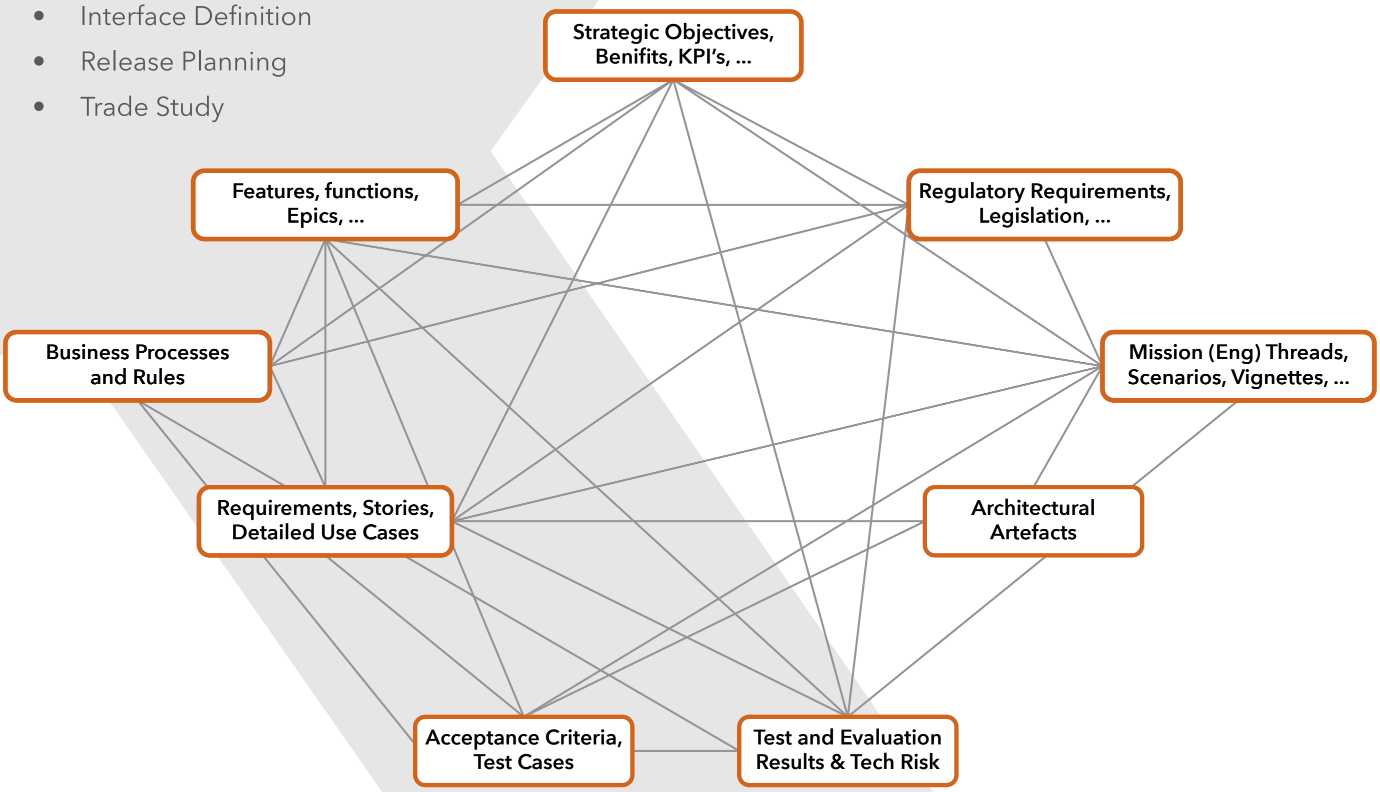
DATA DRIVEN INSIGHTS

Digital Engineering made easy

Kompozition does this in a single integrated model. Many of the artefacts in the list below can be synthesised from other artefacts in a model-based approach These models integrate the solution and operational domains (technical and human aspects of “the system”)

Kompozition provides the traditional features required to manage the Systems Engineering life cycle, and integrates information at the elemental level, including:

- The capture of Objectives, Assumptions, Dependencies
- Capability and Capability Gap Capture
- Requirements Management (Elicitation, Capture, Refinement, Trace, Allocation)
- Behaviour/Interaction/Process Modelling with Requirements Trace and Linkage to the Functional Architecture, Allocation, Domain Models, Interfaces, etc.
- Issue Identification and Management
- Test and Evaluation Capture and Trace
- Architecture Capture and Trace
- Domain Modelling
- Interface Definition
- Release Planning
- Trade Study



kompozition_
smarter together

TECHNICAL CAPABILITY OVERVIEW

Kompozition is a Model Based Systems Engineering (MBSE) platform that helps organisations accelerate and de-risk technical projects and maintain the systems they deliver.

The Kompozition Statement

Kompozition is a Digital Mission Engineering platform that helps organisations accelerate and de-risk technical projects and maintain the systems they deliver.

Kompozition enables the development, maintenance and querying of a single, continuous, and integrated knowledge graph (digital thread) of digital artefacts that represent the system, its verified design and architecture - including a representation of its expected behaviour in the operational environment, and the contracts/requirements that predicated its design and constrain its operation.

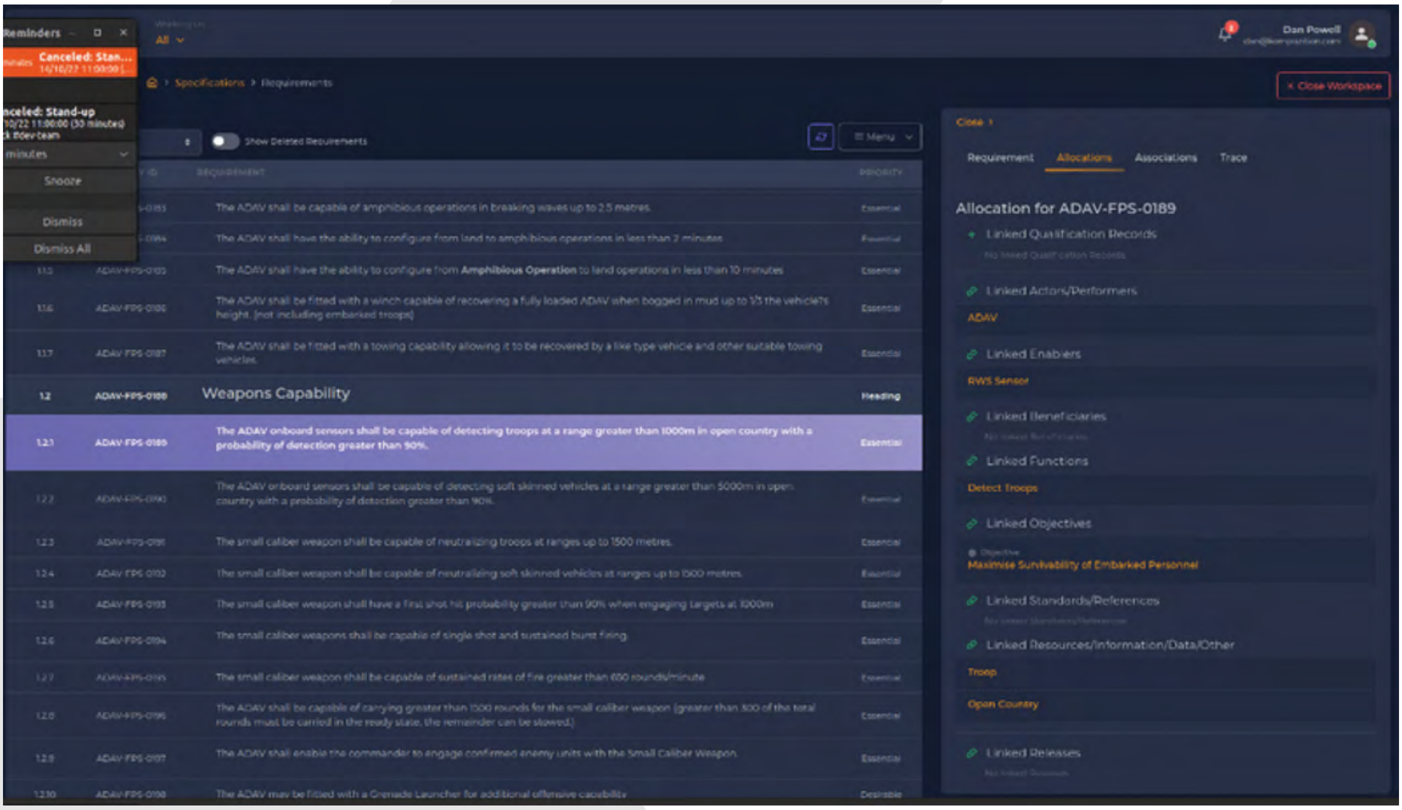
This digital thread can extend from the operational concept through to the detailed design representing the as-delivered solution, with links at all levels of detail to the records of verification and validation that qualify the intent and the solution.

The creation and querying of an integrated, detailed model describing intent and design are necessary to develop a deep, accurate and shared understanding of increasingly complex systems.
When delivered with the system (as a digital twin), this model subsequently enables the continuous monitoring and analysis of the impacts of change to, and on, the system, the operational environment, and any contract requirements or constraints that predicate its design and use. Using an extensible, integrated Kompozition knowledge graph, it would be possible to assess any such changes operational, technical, and contractual impacts.

DATA DRIVEN INSIGHTS

Requirements Management

The Kompozition platform has a full-featured requirement management capability providing all the features required to perform requirements definition and management; including tools to support requirements elicitation and analysis, requirements definition/authoring, trace management, allocation, verification and validation, requirements review and approval, baseline and delta management, and integrated uncertainty management.



An integrated mechanism

Kompozition provides an integrated mechanism for managing the information within a requirement in a consistent digital thread with architecture (operational through technical), design, evaluation and release planning information The difference between Kompozition and other requirements management and architecture/systems engineering tools (CSM, Sparx EA,) is that Kompozition uses AI, heuristic algorithms and an expressive modelling language to help synthesise and maintain a consistent integrated knowledge graph.

kompozition_
smarter together

Beyond Traceability

Like a traditional requirements database, Kompozition maintains an object for a requirement and maintains the trace and history of that object.

However, Kompozition does not just capture artefacts like requirements and traces at an object level, Kompozition uses a Natural Language Processing (NLP) framework to extract and relate the elemental information inherent within the requirement. In the example on the previous page, based on the NLP parse, Kompozition automatically:

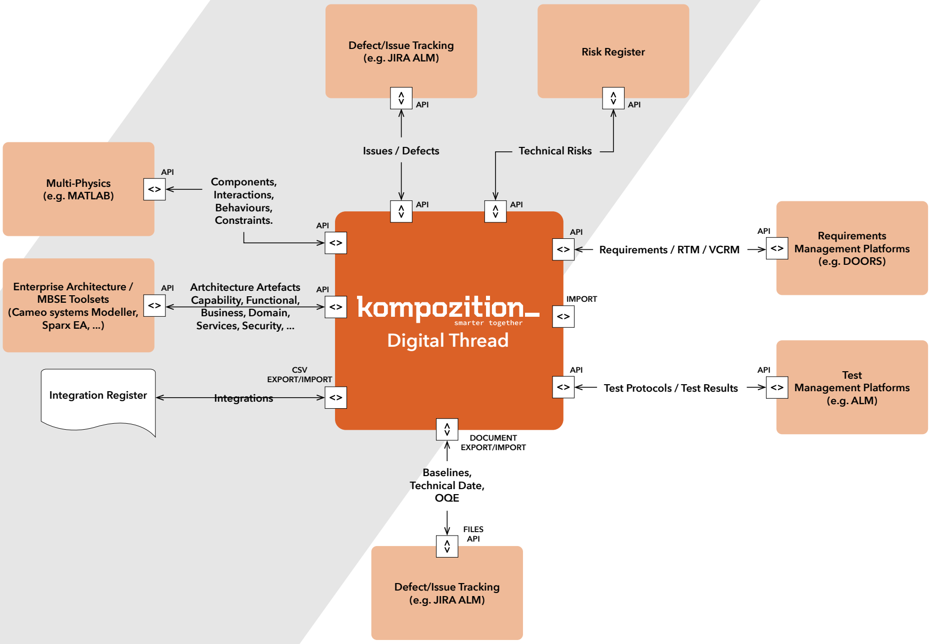
- **Identifies a function, Detect Troops (and captures it in the functional decomposition) and captures and associates performance properties and conditions (range, open country conditions) with this function.**
- **Allocates that function to the ADAV as the performer and the RWS Sensor as the enabler (creating either in the ontology and system breakdown, if required - noting that Kompozition automatically consults the knowledge graph that already records On-board Sensor as an alias of RWS Sensor).**
- **Extracts behaviour representing the Detect Troops event, with its pre-conditions and the resulting state (where the resources/information Troop and Open Country and the relationships between them are all captured in the ontology as part of the Domain Model).**
- **Identifies and classifies uncertainties around the fact that the knowledge graph doesn't have any information establishing what it means to be in open country and whether the ADAV and the Troops both need to be in Open Country (and links this uncertainty with the requirement, the function, the behavioural representation, and all relevant entities in the ontology), and links the requirement, through the functional decomposition to an objective (already in the platform).**

The knowledge graph links all these pieces of elemental information (compositional, structural, behavioural and uncertainty) with the requirement and with each other.

As a reference, the synthesis and integration of information from a set of requirements, managed within the Kompozition platform was most recently applied within Defence for JP9102 to create a high-fidelity model of their OCD and FPS.

Technical Information

Kompozition provides a native user interface to define and manage the digital artefacts within the knowledge graph (digital thread) Kompozition also provides an open API around the knowledge graph, enabling the integration of existing tool-sets to provide the knowledge graph with models one possible tool-set integration scenario, might look like the following;



Kompozition has been successfully integrated with DOORS NG and JIRA for requirements and backlog management and interfaces to Cameo Systems Modeller and Sparx EA have been implemented for a number of defence primes, based on the export and import of UML/SysML XMI from Kompozition's functional architecture, system breakdown, domain ontology, interface register and requirements databases (including their links) Limited behavioural information can also be exported to UML/SysML views - limited because the Behaviour Tree notation used within Kompozition is more expressive than the behavioural languages of the UML/SysML (BPMN, activity diagrams, sequence diagrams, etc.

Kompozition has been used to generate DODAF/xDAF views. A paper was presented at SETE in 2011 describing the fundamentals of the generation of many commonly produced DODAF operational and systems views from the Behaviour Tree modelling language used by Kompozition.

Kompozition can be deployed either on-premise or hosted in the cloud Kompozition is containerised and can be run completely isolated (unless integration with existing 3rd party tools is desired)

kompozition_
smarter together

Accelerating Model Development

Kompozition enables the lean development of digital models Kompozition is built on AI, a unique representation of intent, and graph algorithms that identify and parse information as it is being provided to Kompozition to extract and relate the compositional, structural, and behavioural elements of the digital artefacts necessary to design and assure systems. This tool support means that the development of large, high-fidelity models is achievable within the constraints of human short-term memory.

Kompozition can do this from multiple sources of information that define intent.

For example, Kompozition can be used in a model-first approach where the analyst captures, for example, a behavioural model of expected system behaviour and constraint as a starting point and then, uses the AI and algorithms within the platform to synthesise the functional architecture, ontology, interface register, and requirements and acceptance criteria implicit in that behavioural description of intent.

Alternatively, Kompozition could be used to ingest natural language requirements from a requirements management tool, like DOORS. The AI and algorithms within the platform could then be used to parse the natural language extracting/synthesising and li n king representations of the behaviour, functional architecture, allocation, ontology, interface register and acceptance criteria implicit in that requirement's natural language.

DATA DRIVEN INSIGHTS

The Importance of Modelling Behaviour

The Legacy of Prof. Geoff Dromey

Effective capability definition, development, acquisition and integration require the ability to not only reason about what operational outcomes need to be achieved but to reason about what courses of action could be performed (what needs to be done), to what levels of performance (how well do they need to be performed), and under what conditions (in what operational environment and state - when - do they need to be performed) to achieve those outcomes.

As engineers, our mission is to design and assure system solutions (solutions integrating technology, people, organisations, processes, etc.) to optimise the achievement of positive mission outcomes while reducing risks that threaten those outcomes.

What we need to be able to model

We must start with a high-fidelity understanding of the behaviour we are trying to design into the solution. We need a deep, accurate and shared understanding of that behaviour. We need a description of the deterministic and probabilistic courses of action and interaction possible under certain antecedent conditions and their likely consequences. To enable mapping of enabling solution elements to those courses of action, those courses of action must be defined with enough fidelity to answer questions of what is being done when (and to some extent by whom/what); what are the causal and temporal relationships between the things being done; what resources are being exchanged and what is being done with those resources; and how are those aspects interrelated.

Properties required of those models

There are two critical properties that those models must possess in order to optimise and assure system solutions:

- The models of behaviour must be valid. Currently, validation is performed by human subject matter experts. So, the models must be human readable and lend themselves to scenario-based walkthroughs; and
- The models must be formal enough to support mapping and reasoning about enablers, fundamental inputs to capability, and their predicted and observable performance and impact on the outcome.

kompozition_
smarter together

The Importance of Modelling Behaviour

The Behaviour Modelling Language

The Behaviour Models within Kompozition are not built around the behavioural/interaction views defined within the UML/SysML. These UML/SysML views are disparate. They require a large effort to keep consistent, let alone integrate, and even more effort is required to maintain consistency between behavioural views and the associated compositional and structural views. Walkthroughs, particularly walkthroughs with non-engineer subject matter experts, are ineffective. Maintenance is difficult. UML/SysML views do look good in PPT decks, though.

Kompozition is built around the Behaviour Modelling Language (BML) developed by the late Prof. Geoff Dromey as the language behind the Behaviour Engineering (bE) methodology in the late 19990s and early 2000s. The bE methodology and the BML were very successfully applied in industry to create high-fidelity models of large systems in the defence, aerospace, transport and government sectors to enable a deep, accurate and shared understanding of the needs of those systems and the uncertainty and risks associated with them.

The Behaviour Modelling Language allows the constructive (iterative and incremental) development of very large, high-fidelity integrated Behaviour Trees that are human-readable and proven effective for validation, at the same time as being formal enough to be useful for deriving and justifying design and for supporting simulation and execution.

The Behaviour Modelling Language is expressive enough and general enough to capture behaviour at the broadest operational/contextual level, down to the level of defining algorithms.

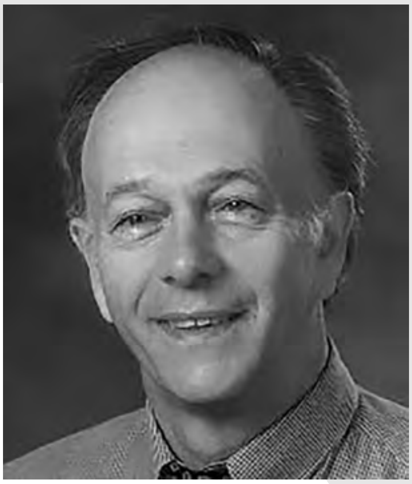
As opposed to maintaining consistency between behavioural, compositional and structural views, the BML representation of behaviour contains sufficient information to synthesise the compositional and structural information required of that behaviour - a significant chunk of the architecture at the level of detail of the model.

The Behaviour Modelling Language Cont.

For example, once we have defined behaviour using the BML, the Kompozition machine learning and heuristic modules can deduce what resources, information and data are required in the performance of that behaviour; what relationships between these things are established, broken or tested during the behaviour; what activities/functions are performed by what actors; how those actors interact and interface; what is exchanged between them; … Kompozition can automatically link the behavioural, compositional and structural aspects of the problem space (and provide the ability to architects and engineers to modify and refine that synthesised information).

If we create a Behaviour Tree at the operational level of detail in Kompozition, then Kompozition can automatically synthesise and link a very large proportion of the operational/functional architecture and functional requirements based entirely on the definition of the problem. This synthesis of digital threads of information within the Kompozition knowledge graph greatly reduces the effort and risk in MBSE/DME endeavours.

The behavioural, compositional and structural information in a Kompozition model of a problem space, provides "hooks" to be linked to the digital description of a solution to that problem to whatever level of depth is required, with a record of all verification and validation qualifying that solution (definition, trace and results), enabling and end-to-end model-based approach to capability definition, acquisition and integration that could also be used for mapping real-world observables to operational intent post-acceptance into service.



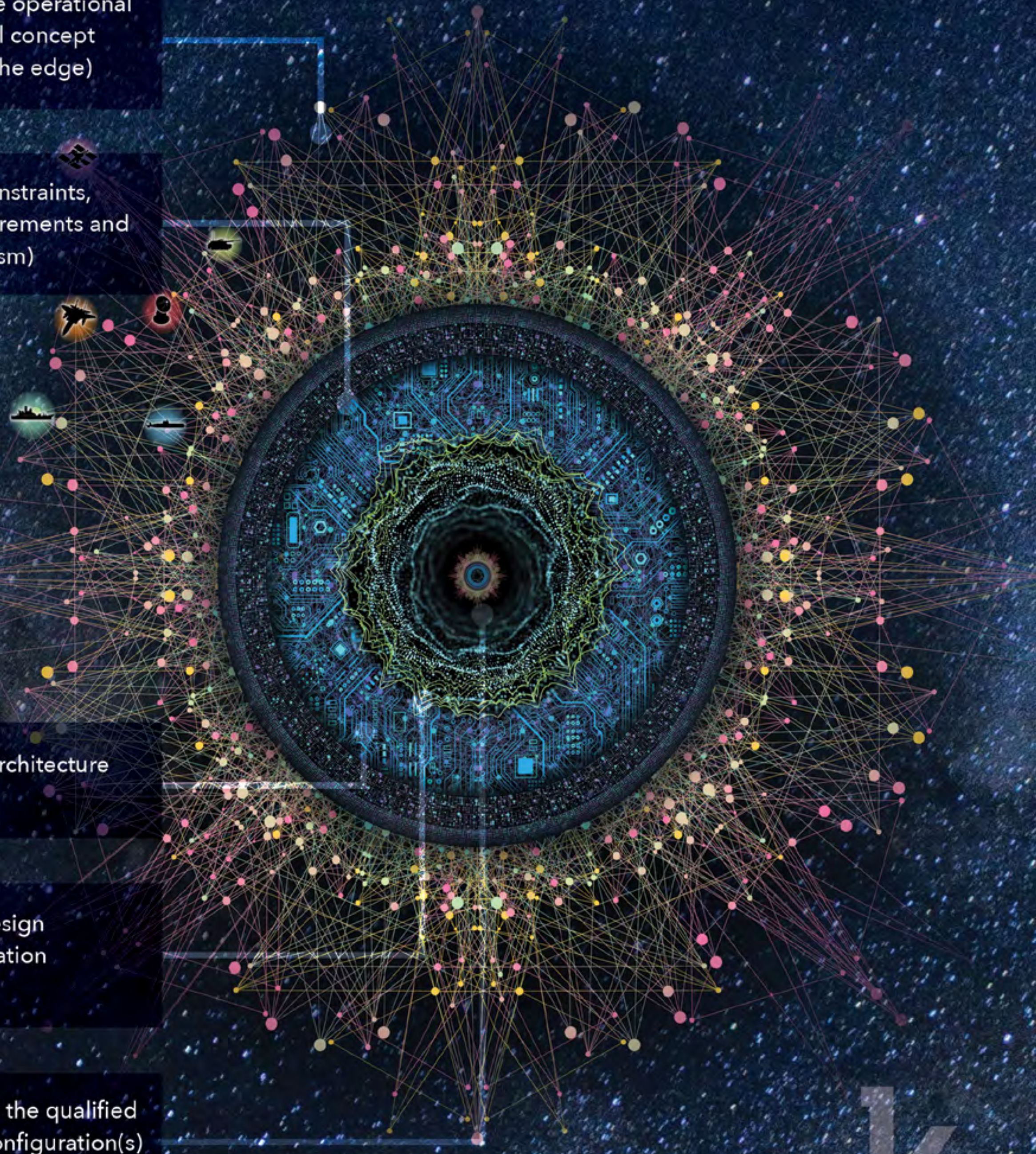
The enabler for all of this is, of course, the Behaviour Modelling Language developed by Prof. Geoff Dromey

This is **kompozition_**

smarter together

A digital thread that defines the operational problem space and operational concept (modelling mission threads at the edge)

A digital thread that defines constraints, specifications, contracts / requirements and functional architecture (formalism)



A digital thread that captures architecture and design

A digital thread that records design qualification (verification, validation & evidence)

A digital thread that represents the qualified as-built, as-deployed system configuration(s) in the operational environment

Kompozition creates and manages a knowledge graph

A single, continuous, integrated digital thread of information from the definition of operational concept in the operational environment (mission threads), through specification, contract/requirements, architecture & design and the qualification of that design against the intent.

enquiries@kompozition.com

kompozition.com

Let’s work smarter together