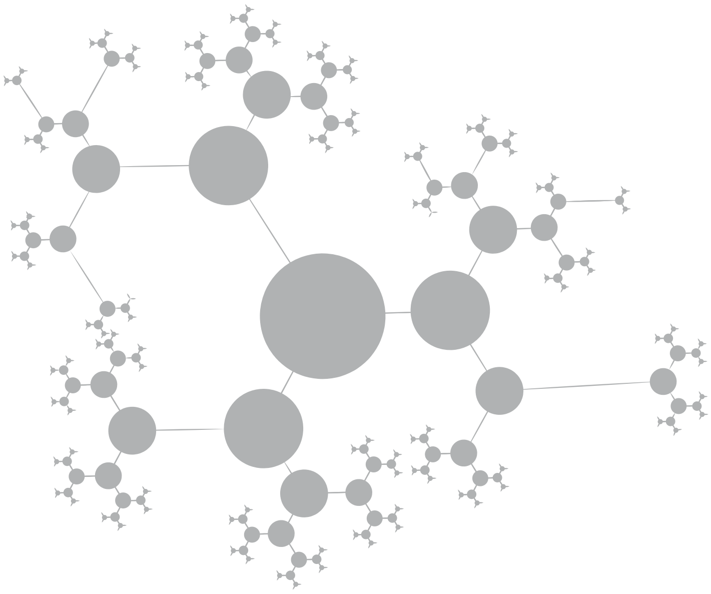
Vitech



four day training course

Model-Based Systems Engineering with CORE™

Model-Based systems engineering solutions to integrate systems engineering across your development team.

Integrate requirements management.

Model systems functionality.

Allocate behaviour to system architecture.

Trace design to verification and validation.

Letter27

Presented by Cobus Scannell

cobus@letter27.co.za

How will this training add value to my company or organization?

Increased competency and efficiency in Model-Based Engineering and use of CORE™ software.

Ensure shared understanding in your Systems Engineeirng CORE™ team.

Top up on knowledge, skill, and confidence in applying Model-Based Systems Engineering in the CORE™ environment.

Develop understanding of document generation in CORE™ .

Benefit from the support and insights of a mature, internationally developed course from Vitech Corporation.

What will the participants gain from this course?

Experience in solving a sample problem using a Model-Based Systems Engineering methodology. (STRATA™)

Insight into the CORE™ Systems Definition Language, Design Repository and Graphical View Generators.

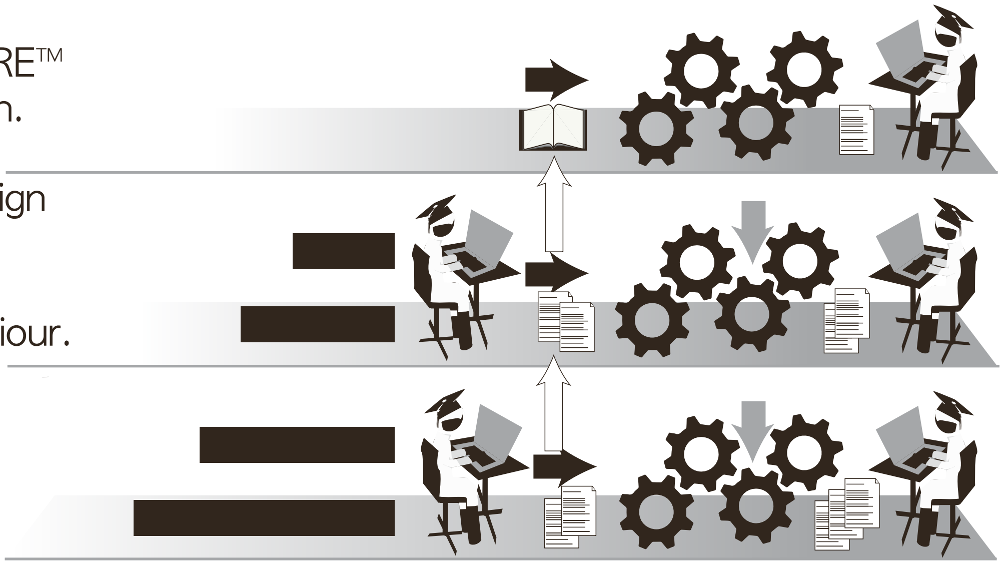
Skill to generate representative Systems Engineering documentation in CORE™.

Experience in applying CORE™'s modelling and simulation capabillity.

Learn to work with the CORE™ interface through application.

Skill to effortlessly build design traceablity in CORE™.

Insight into modelling bahaviour.



four day training course
Model-Based Systems Engineering with CORE™

Cobus Scannell cobus@letter27.co.za www.letter27.co.za

Course Outline	Key Concepts
1. Model-based Systems Engineering Introduction and Concepts Explanation of the basic concepts of Model-Based Systems Engineering.	System; systems engineering; system life cycle; return on investment; role of the Systems Engineer; systems engineering process; model; Model-Based Systems Engineering; synthetic thinking; MBSE layered process; views; diagrams and models
2. CORE™ Environment The participant will start the tool and explore the operating environment.	CORE™ ConOps; technology; system definition language; importing and opening projects; Project Explorer; project navigation; facilities; elements; Element Browser; attributes; relationships; versions; perspectives; schema; View Window; Exiting CORE; repository; view generation
3. Getting Started - Walkthrough Creation of new accounts and projects as well as setting project permissions and user preferences.	Logging in; new account; new project; project explorer properties; project permissions; user preferences; file locations; diagram preferences; preparing for development activities
4. Capturing Originating Requirements The participant will capture originating requirements beginning at the highest level.	Document parser; element extractor; elements; source documents; parsing; creating elements; parsed requirements; debris; component; editing; numbering and renaming requirements; composite requirements; child requirements; relationships
5. Parameters Discussion of the differences and similarities between attributes and parameters and their uses.	Attributes and parameters; versioning
6. Filter & Sort This section introduces Filtering and Sorting data in the tool.	Filters; creating and editing filters; sort blocks; creating and editing sort blocks
7. Saving Your Work This section covers the steps to saving the data and exiting the tool as well as the various types of files that can be created byCORE™ .	Save repository; exit CORE™ ; CORE™ file types

Course Outline	Key Concepts
8. Document Generation This section treats the powerful advantage available in CORE™ as a result of the capability to generate documents FROM the database directly. Using the Requirements Table and the System Description Document (SDD) as examples the participant will generate reports from the tool.	Reports; generating standard reports; requirements table; SDD
9. Concerns & Risks This section covers using the tool to capture, preserve, display and manage Concerns and Risks.	Concerns; risks; target; traceability; risk element; hiding elements on diagrams
10. Presenting Your Work The use and production of various diagrams is introduced and discussed.	Diagrams; hierarchy; spider and requirements diagrams
11. Behaviour Constructs Modelling the relationships between entities and their arrangement through control constructs to support the logical architecture of the system.	Behaviour; logic; logic constructs; graphical notation; control constructs; constructs in combination; items; execution and conditions; graphical representation; creating behaviour; CORE™ sim; constructs and behaviour; exits; loop; iterate; replicate; concurrency; sequence; data stores; triggers
12. Complete the Context Layer This section discusses the system and functional boundary definitions and continues the development of the requirements. It also introduces the Model Assistant as an aid to the modelling process.	System boundary; external systems; physical context; structure block definition diagrams; interfaces; interface diagrams; interface elements; N2 diagrams; Model Assistant; relating functional and physical; functional boundary; requirement types and relations; constraints; performance objectives
13. Bitmap Graphics The ability to use bitmap graphics from internal and external sources to enhance visualizations in CORE™ is discussed in this section.	External graphics path; associating graphics with elements; using graphics in diagrams
14. Use Cases and Threads The participant will construct use cases and threads for the example system.	External interactions; use case; use case relationships; use case diagrams; relating use cases to requirements; threads; behaviour diagrams

Course Outline	Key Concepts
15. Folders Here use of folders in the CORE™ tool is introduced.	Folders; creating and using folders
16. Derive Integrated System Behaviour The integration of threads into an integrated behaviour for the system is covered here. Requirements are linked through functions to the threads. Traceability through the integrated behaviour is established.	Integrated behaviour; relating threads; linking requirements to use cases and functions; trace ability diagrams
17. Executable Behaviour I This section introduces the methodology and tool (CORE™ sim) for analysing dynamic system performance and behaviour. The participant will apply the tool to the system model of the sample problem.	CORE™ sim; integrated simulation; kill branch; function decomposition
18. Allocation This section introduces and discusses the concept of allocation and aspects of allocation such as aggregation and determining when the allocation is complete.	Allocation; partitioning completion; decomposition; failure modes; physical architecture; system-level architecture; allocation decision; trail allocation; trade studies; primary design traceability
19. Interfaces and Links - Context Level Here the participant will identify interfaces, create the links that make them up and represent the interfaces and links in the tool using a number of diagrams.	External interactions; interfaces; interface termination; links; interface hierarchy diagram; physical block diagram; physical N2 diagram; flow block diagram
20. Verification Requirements and Test Planning This section introduces verification and test planning and it covers test threads and sequences as well as the use of the Dynamic Analyser. The participant will learn how to develop verification requirements and a test plan.	Verification; testing; test threads; verification hierarchy; verification property sheets; component interfaces; test sequences; Dynamic Analyser
21. Versioning The CORE™ tool capability for versioning is covered in this section.	Versioning; attribute version browser; purging; baseline control; attribute history; project compare; schema attribute properties; audit log

Course Outline	Key Concepts
22. CORE™ Scripts This section introduces the participant to ability to customize the behaviour of the CORE tool using scripts and a sample of its uses.	Script; external file; generating reports; html reports; data exchange; standard scripts
23. Formal Documentation This section covers generating formal documentation. The participant will generate documentation from the sample problem.	Formal documentation; documentation management; document production; templates; PUIDs; modifying documents
24. Executable Behaviour II The discussion of the definition and modelling of executable behaviour is extended in this section. Resources and function duration are introduced.	Resources; function duration; resource timing; resupply; replenishment; database queries; IF, THEN, ELSE constructs
25. Organizing, Navigating, and Focusing on Models The CORE™ tool capability for packaging as an organization strategy and the use of perspectives to focus model aspects are discussed in this section.	Package; package hierarchy; perspectives; navigation
26. SysML Diagram Representations This section discusses the SysML views available in CORE™ .	Requirements diagram; structure block definition diagram; activity diagram; use case diagram; interface N2 diagram; interface block diagram; flow interface block diagram; sequence diagram; physical block diagram; physical N2 diagram; flow internal block diagram; package diagram; classification block definition diagram
27. DoDAF 2.0 The Department of Defence Architectural Framework (DoDAF 2.0) is included in CORE™ as a separate schema. This section discusses the DoDAF and its use in modelling and the CORE™ tool in particular. Specific modelling and design challenges arising from DoDAF are covered as well.	DoDAF; architecture; software architecture; system architecture; enterprise architecture; integration and federation; frameworks; architecture definition; DoDAF views; DoDAF schema