

Cradle-SYS

The **Cradle-SYS** module provides an analysis, process, architecture and design modelling environment that, being linked into the systems engineering lifecycle, provides full traceability and coverage for all model information. You can use SysML, UML, ADARTS, SASD, eFFBD, IDEF, BPM and other notations to achieve your MBSE goals.

A model is an abstraction of an aspect of a system being developed. Therefore, models should not be separate from the needs, goals and objectives that the model seeks to satisfy, nor from the tests that validate the system's compliance. Hence Cradle integrates modelling into all requirements and other systems engineering data, so every component of every model is traced to the highest level user need and to the lowest level test result.

This applies to agile and phase-based processes. An agile process has no less need for models simply because its iterations are short. To neglect rigorous design in agile projects will ultimately compromise the system if a clear design is not modelled at the outset and maintained through each iteration.

Each Cradle database provides analysis and design domains. Each domain can contain any number of models, optionally organised in hierarchies. Models can be used to represent concepts such as:

- Alternative missions in a CONOPS
- Products within a range
- Regional variants of a product
- Comparative analysis of architectures

Each model contains any number of diagrams from a wide variety of notations. Each diagram contains symbols, and each symbol is described by a data definition or specification.

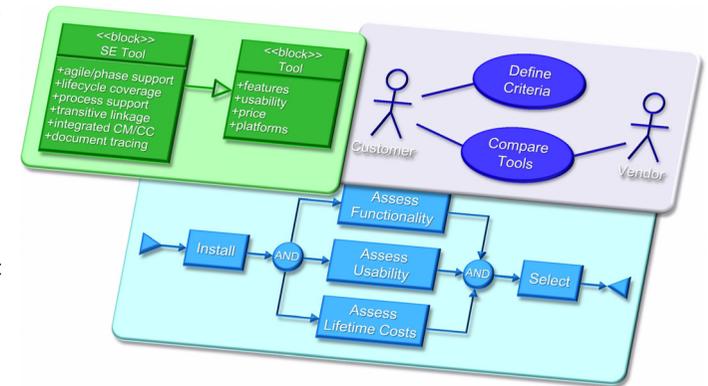
All diagrams, specifications and data definitions in

a model can be cross referenced to each other and to information in all parts of the lifecycle. So user requirements can be linked to use cases, that are linked to system requirements, that link to a logical model of system behaviour, that can be allocated to a logical architecture, which in turn can be allocated to multiple physical architecture models for assessment

A System Breakdown Structure (SBS) is useful as an abstraction of the system composition, and as a single structure to which all the requirements can be linked. The alternative system architectures and designs can be explored, each in its own model, all models linked to the SBS. This simplifies traceability for the requirements and the performance constraints, without restricting the modelling activities.

Models can link to a Product Breakdown Structure (PBS) to allow linking to a PLM environment.

Cradle has over 20 diagram notations from methods including UML, ADARTS, IDEF, SASD, SysML and data, process and architecture modelling. The notations can be combined when semantically viable. Cradle does not limit you to one method, nor constrain your choices for the notations that



will best express the system for the audience of that model.

Cradle provides a consistent, interface to building diagrams. It includes time-saving features to build diagrams in time-sequenced notations, such as Process Flow Diagrams (PFDs) and extended Function Flow Block Diagrams (eFFBDs).

In hierarchical notations, Cradle has a range of features to build both child and parent diagrams that are automatically consistent.

Cradle enforces diagram





syntax when editing. Completeness and I/O consistency checks are provided, both within a diagram and between diagrams to ensure conservation of data and function. Cradle can also check the consistency of diagram graphics and text descriptions. For notations that use it, Cradle provides a full Data Dictionary with a formal BNF notation to describe data composition.

In architecture models, Cradle supports data protocol descriptions across interfaces and can generate message formats (bitfields) that describe the formatting of the messages in all data exchanges.

Diagram symbols can be coloured and have embedded graphics, to ease users' understanding of the model.

Notations can be combined, such as using UML and other diagrams in the same model. Some notations can be used in many contexts. For example Sequence Diagrams (SQDs) can show a message protocol across an architecture model interface, their role in SDL before their use in UML.

All model elements can contain graphics, video, figures, tables, equations, URLs and integrate with desktop tools including Visio®, Word® and Excel®. Each diagram, data definition or specification is an item in the database and so can contain any number of attributes each containing, or referencing, up to 1 TByte of any type of data.

Models have change histories, discussions, comments, are formally reviewed in Cradle's CM system, and can be baselined.

Models can be printed to a variety of devices. They can be part of user-defined documents with requirements, tests and any other information. Models can be published into static, hyperlinked websites that provide links between diagrams and between symbols and the descriptions. All Cradle web UIs support viewing and navigation of models.

Models can be loaded from other tools by import or data conversion from other tools' data formats or XML.

Feature Summary

Feature	Benefits
Separate analysis and design domains	Separate implementation dependencies from essential behaviour
Any number of models	Define model structures suited to the project, such as mission analyses, analysis of alternative system architectures, or model multiple variants of multiple types of multiple products
Fully linked into the lifecycle	Link models to source requirements, features, tests, risks, issues, verifications and any other items, and link to a SBS and/or PBS to correlate information with non-modelling tools including PLM
Wide range of functional, object, data & architectural / physical notations	Not limited by SysML, UML, ADARTS, IDEF or SASD, use whichever notations are most appropriate and combine them when this is helpful
Multiple architectures and shared elements	Study multiple architectures and separate function allocations
Shared representation of diagram symbols	All uses of a data or behavioural component on any diagram share a common item in the database
Automated diagram construction	Create parent and child diagrams that are automatically consistent and balanced
Syntax checking editors	Diagrams and textual descriptions are immediately correct when created and edited
Comprehensive consistency checks	Absolute consistency of function, data and interfaces between diagrams and between diagrams and supporting text descriptions
Diagrams, specifications and data definitions have unlimited attributes	Model elements have unlimited attributes of any data type including plain and rich text, dates, numbers, single and multiple value lists, images, video, documents and any user-defined data types
Interfaces to other tools	Link to parts of PDF or Visio diagrams. Import/export from SVG, XML. Load from other tools including Teamwork, BPwin, RDD-100 and CORE.

Supported Notations

Methodology	Notations
SysML	Activity Diagram, Block Definition Diagram, Internal Block Diagram, Package Diagram, Parametric Diagram, Requirement Diagram, Sequence Diagram, State Machine Diagram, Use Case Diagram
UML	Use Case Diagram, Sequence Diagram, Collaboration Diagram, Class Diagram, Statechart, Activity Diagram, Package Diagram, Component Diagram, Deployment Diagram
Functional	Data Flow Diagram, IDEF0, extended Function Flow Block Diagram
Data	Entity Relationship Diagram, Data Structure Diagram
Dynamic	State Transition Diagram
Architectural	Physical Architecture Diagram, Architecture Interconnect Diagram, Software Architecture Diagram
Process	Process Flow Diagram
Source code structure	Structure Charts, Ada Structure Graphs
Organisation	Hierarchy Diagram