Email
info@discon.co.za
Website
www.disconspecialists.com
Phone
(+27) 12 667 5975
LinkedIn
www.linkedin.com/company/discon-specialists

DISCON Specialists

EA Enabling Techniques

Functional Effect Back Tracking (FEBT)

Reason for Existence

Architectural preference in terms of development order as well as the natural boundaries for business objects can be determined by applying the FEBT technique to an Attribute Dependency Diagram (ADD). The FEBT algorithm will deliver a number of useful things:

- Mathematically determined architectural development order of subsystems.
- Scope of all the systems.
- Mathematically determined system cohesion. System cohesion is how strongly systems are bound together by definition of mathematical rules.
- Context is expressed in terms of which building blocks the business consists of and which building blocks are naturally related.
- It provides a base from which mathematically derived objects are defined, as in the Object Oriented paradigm. These object definitions are in terms of which methods are associated with the objects, and which properties are associated with which objects.

Inference Rules

Transitive Functional Dependencies

If X determines Y and Y determines Z, THEN X determines Z by definition. Transitive Functional Dependencies get assigned a weight of -1 for the application of this algorithm. These dependencies are the weakest functional dependencies in terms of this algorithm and virtually make no contribution.

Ordinary Functional Dependencies

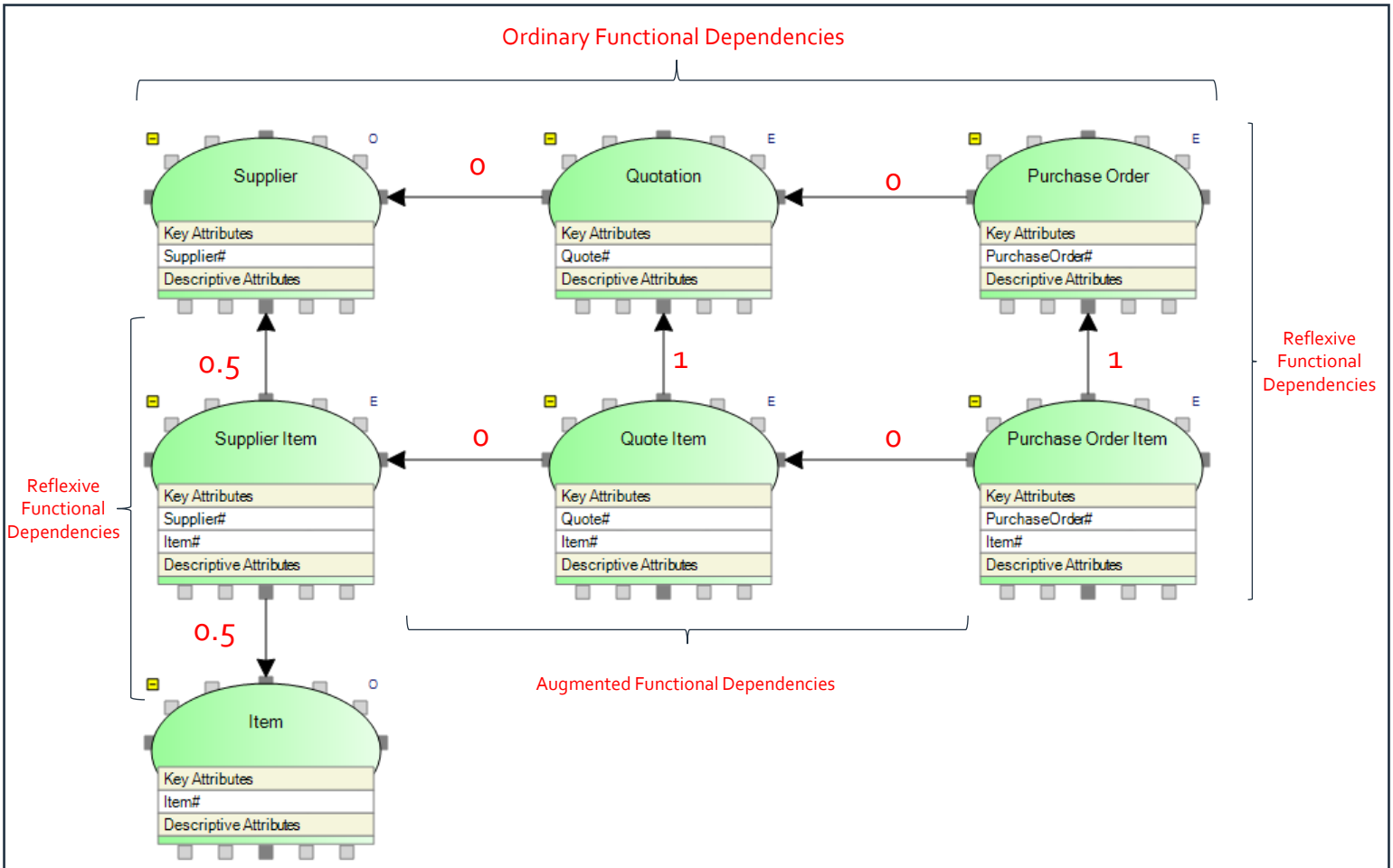
Ordinary Functional Dependencies do not conform to any of the inference rules. System boundaries must cross the Ordinary Functional Dependencies. Ordinary Functional Dependencies get assigned a weight of 0.

Augmented Functional Dependencies

If X functionally determines Y, and if Z is a true super-set or equal to K, and if XZ and YK exist somewhere together, then XZ functionally determines YK, by definition. These functional dependencies are not as powerful as Reflexive ones. Augmented functional dependencies get assigned a weight of 0. This means that system boundaries have to cross these functional dependencies.

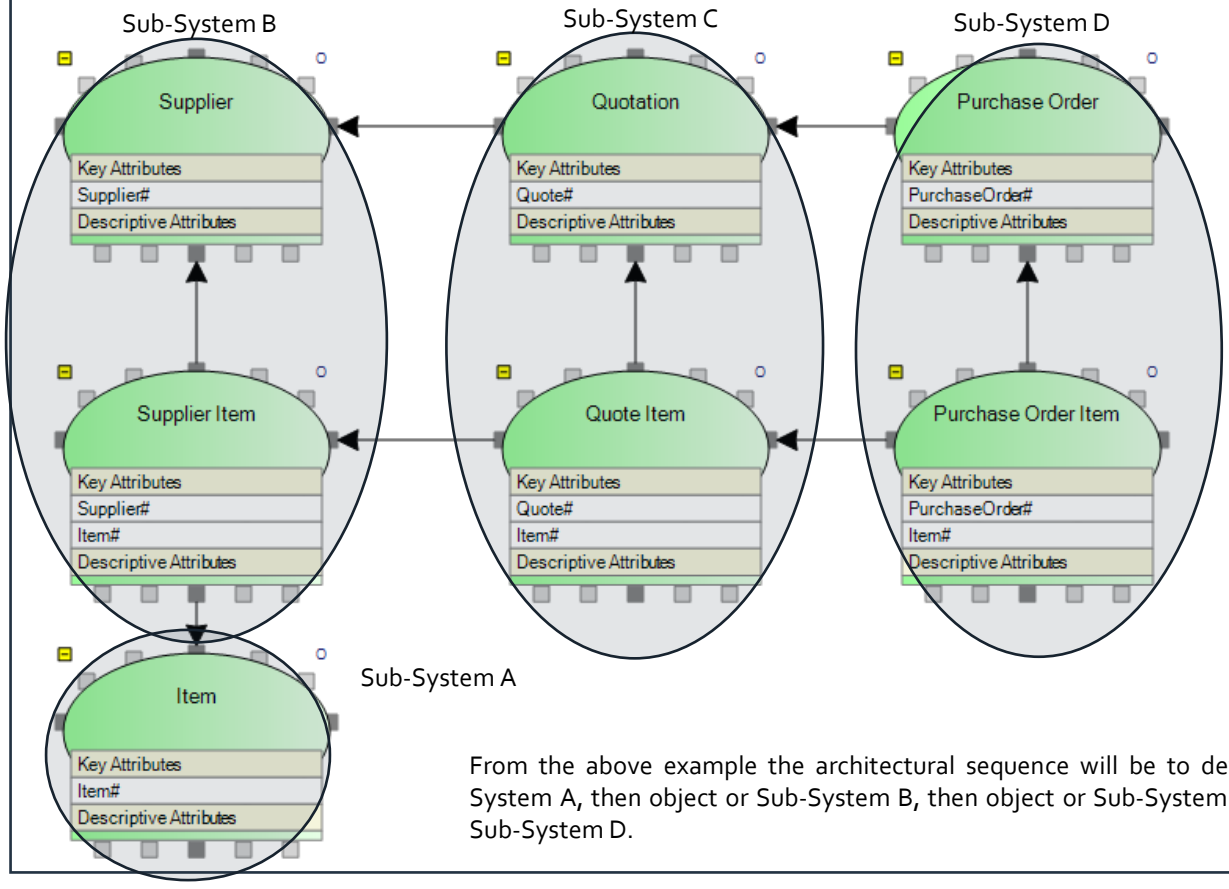
Reflexive Functional Dependencies

If X is true super-set or equal to Y, then X functionally determines Y.
Reflexive functional dependencies between the keys of two Event entities or two Object entities get assigned a weight of 1. They are the most powerful functional dependencies in terms of this algorithm. This means that a system boundary is never allowed to cross these functional dependencies.
Reflexive links between Object and Event entities are less powerful in terms of this algorithm. They get assigned a weight of 0.5. this means that they could be crossed for the sake of forming system boundaries.



Mathematically Determined Sub-Systems

The definitions of the dependencies and the diagram below show that we can mathematically determine the correct definitions of the boundaries of our sub-systems through the use of FEBT. Therefore the FEBT mathematical algorithm answers the question: 'Which building blocks does the business consist of and how are they inter-related?'.



Scope:
From where to where does a sub-system span?

Context:
How are my sub-systems related together?

Content:
Which elements do my sub-systems consist of?

From the above example the architectural sequence will be to develop object or Sub-System A, then object or Sub-System B, then object or Sub-System C and then object or Sub-System D.