A Glossary of Key Terms

auros
Active Knowledge System™

Retained Know-How

Essential knowledge that is systematically captured and preserved.

Knowledge Packet

A structured, granular unit of retained know-how; purpose built to efficiently capture, share, reuse, and provision knowledge.

Value Table

A structured element of a knowledge packet that encapsulates important logical relationships across relevant attributes, variables, or parameters resulting in a deterministic knowledge model.

Assessment Control

A comprehensive toolkit used to efficiently apply, evaluate, and track collections of knowledge packets within the flow of work.

Community of Practice

A structured organizational unit that allows a team with common interests to configure how they create, share, reuse, and provision knowledge packets.

Rule Processing Engine

A knowledge processing engine that is used to integrate and evaluate Knowledge Packets, with no effort by an end-user. It interprets the knowledge modeled within a Knowledge Packet, and then publishes it as a set of services into any digital environment that wishes to apply that knowledge.

Vitality

A statistical measure of the confidence of quality or health of a knowledge packet or a Community of Practice.

Proximity Search

An advanced semantic search technique that automatically connects related knowledge packets without the need for classification or metadata.

Active Knowledge

Retained know-how that performs work. In general, knowledge is 'active' when:

1. Provisioned or allocated. Knowledge is active in the sense that it moves to wherever, whenever it is needed based. For the end-user, this eliminates the need for 'searching'.
2. Knowledge autonomously reacts or influences in context. Knowledge is active in the sense that automatically evaluates status or calculates results based on the context.
3. Knowledge auto-connects. Knowledge is active in the sense that users do not have to hard wire connections from one piece of knowledge to another; relevancy of retained know-how are derived based on context.

Technical Memory

A general term used to describe an enterprise wide capability to collect and reuse Retained Know-how continuously.

Knowledge In-the-flow of Work

A general term used to describe a process that does not require the end-user to perform any searches for relevant retained know-how and where retained know-how participates in the flow of work in ways that are specialized to the context.

Bottom-Up Style Software Applications

Bottom-up systems have the following distinguishing properties:

1. Is comprised of fundamental 'units' that can build out to more complex systems (eg. Legos (unit=bricks), biology (unit=genes), Minecraft (unit=blocks of different types and properties)
2. 'Systems' built from 'units' can be assembled and adjusted in rapid and successive small cycles…without IT intervention… this allows continuous, adaptive, flexible, robust processes.
3. High level of distributed administration and configuration (vs centralized). Control is decentralized, but managed.

There are two important results of bottom-up style applications:

1. They are adaptive and robust to change. By definition, bottom-up allows small localized changes and will therefore adapt as environmental or needs change.
2. They are capable of producing 'emergent' properties of both process and effect… these impactful emergent properties evolve and could not be imagined in an up-front specification process. They are not a designed in feature, rather they are 'surprise' positive outcomes of adaptive systems. These bottom-up emergent properties are sometimes referred 'organic complexity'. That is, complexity that arises only from adaptation.