**KRO Scaffold**

KRO Scaffold is designed to mitigate impact of Contextual bias (shaped by multi-faceted extraneous influences) and its influence on Competency Goal driven AI knowledge change management and Outcome improvement efforts. It provides clarity when the variations in a wide range of contextual conditions and factors across different settings can impact the improvement process, influence implementation, skew effectiveness scores, limit semantic scaffold spread and the forecast of sustainability for complex adaptive systems in dynamic environment(s).

**Finite, Facts & Factors**

The Competency Goal driven AI approach has three broad types of knowledge representation objects.

Imperative knowledge is exercised by *an algorithm* as a sequence of steps that are performed during processing (*Imperative programming focuses on describing how a program operates* and is often represented as a partial or complete [finite-state machine](https://en.wikipedia.org/wiki/Finite-state_machine) ). A well-known example is the [procedural reasoning system](https://en.wikipedia.org/wiki/Procedural_reasoning_system), which might, be a smart-guide that provides a path to part of a building, using procedural actions such as "navigate to a room"

Declarative knowledge contains domain-related concepts and data oriented facts (schema, model, taxonomy, ontology, class, category..) such as a map of the building and a model 0f actions that can be done in a building (like moving forward, turning, and stopping)

Contextual factor knowledge content is used (by a domain-independent [planning algorithm](https://en.wikipedia.org/wiki/Computer_planning)) to discover how well the outcome of actuated actions achieved the stated goals. A context-mechanism-outcome (CMO) pattern configuration is a proposition stating the value for ’ what it is about an action outcome which works for whom in what circumstances’.

**KRO Scaffold** **BASI:E: --> F L O model**

KRO Scaffold is designed to be used with a BASI:E: --> F L O model.

* Employing an appropriate data ontology, in a systematic approach for extracting and codifying critical information
* Examine and report the (semantic-space) relationships between multiple documents, messages and their sources by content "meaning"- regardless of the form of expression and degree of completeness and level of obfuscation and temporal factors (e.g. changes over time).

**KRO Scaffold Mapping (KROSM)**

**KROSM** is a semantic workflow matrix which can consist of KR Objects, that in turn, can employ many cell-bound objects, whose location-specific-roles are annotated as *action-based* ‘rows’ and *objective-based* ‘columns’. It should be noted that it is possible for the information contents of a single column to comprise the total object mapping of a simplistic KR Object. Alternatively a KR Object may require the contents of many columns to completely (as in the cells of the Zachman Information Systems Architecture Framework) map the relationships between a set of objects, that are employed in the converting of raw data into Business Analytic. Further, as in a spreadsheet the contents of some KR Objecting cell-bound objects are referenced by other cell-bound objects. Moreover, the content of one ‘cell’ may be directly dependent upon the contents of two other ‘cells’. Finally, although the contents of a Semantic workflow Matrix is not limited to a finite set of rows, it is recommended that as a minimum, the set of objects used in each KR Objecting column identify :

* **What** it is - the information scope of the objective being mapped;
* **When** it is (actively) to be used, e.g., start date - end date;
* **Who,** owns it, needs to (actively) use the Context-mechanism-outcome (CMO) - people, workgroups;
* **How** to use (activate) it - protocol and parameters;
* **Why** (actively) use it - the objective, the value-added purpose;
* **Where** in the organization (network) is it registered for (active) use.

The Beginning Concept

It is a fact that when all the participants of a meeting have the same level of knowledge agreements are quickly reached. By centering upon the use of Business Terminology when articulating specifications, KROSM encourages the sharing of knowledge. However, even in its most simplest form the production of Business Analytic requires a mixing and matching of data assets that are sourced from many places. Enabling the sharing of knowledge, the preparatory phase of Objective Mapping is the cataloging of the data assets. Rather than simply listing the assets, KROSM requires that any use of cryptic computer descriptors for ‘source data’ items is embellished by the application of links to, one or more, realworld concepts, as defined in a Business Terminology lexicon.

When KROSM cataloging it is highly recommended that the ‘Business Terminology links’ are used to implement relationships with multi-token (compound-term) concepts that consist of nouns, pronouns, verbs, adverbs and / or predicates (after all data descriptors are primarily nouns and processing descriptors are primarily verbs). Further, in order, to maximize the contents of the Business Terminology lexicon, that mapping should be based upon : (a) one concept-token having an one-to-one relationships with metadata that is used to catalog the characteristics of data predominantly consists of nouns; and (b) another concept-token having an one-to-many relationship with verbs that populate the scripting actions of KR Objects that convert source data to Business Intelligence. For Business Terminology concept linking purposes a simple Business Analytic Ontology is in Appendix A.

The Data-types of the Workflow Matrix

As noted above each cell-based object of the Semantic workflow Matrix is populated with factors of knowledge that enjoy business relationships with each other. A knowledge factor can take many forms, however, the most complex are those that describe the utilization of many Business Rules, which in turn, employ definitions of activity-based and data-type objects. Typically those knowledge factors will be part of business processes that exploit multiple sets of ‘source’ data and also generate different types of working data. To aid the mapping of knowledge factors which are dependent upon many the data types the following definitions are provided :

* Raw Data - is a source data-type object that must be (is accepted as being) in existence as a pre-condition of the current objective-based activity that is provided by an external (to this KR Object) entity. Is often data that is captured at the place of origin;
* Embedded Data - is a data-type object that is created during the activity but is not stored in a persistent manner nor is it used (as source data) to satisfy any other objective;
* Derived Data - is a data-type object that is created during the activity. Is not be stored in a persistent manner. Is used as source data to satisfy any other objective;
* Virtual Data - is a data-type object that is created during the current activity. Is not immediately stored in a persistent manner. Is referenced as source data by another objective description;
* Intelligence Data - is a data-type object that is created during an activity. Is immediately stored in a persistent ‘engine’ and may or may not be directly utilized as source data to satisfy one or more Context-mechanism-outcome (CMO) objectives.

**The Activity-based Objects of the Matrix**

Additionally, in an attempt to reduce the number of semantic ambiguities related to activity-based objects (especially when the contents of a KR Object / Business Rule is being mapped), KROSM promotes the (exclusive) use of eight words - **Associate, Check, Filter, Prompt, Regulate, Transform, Queue, Warehouse.** Although these words are listed in the Business Terminology lexicon, it is highly recommended that the listing of Business Rule terminology in Appendix B is also introduced as a *guideline to synonym resolution* by all KROSM. A subset of that listing is produced immediately below :

N.B. It should always be remembered that in a natural language statement the use of some activity-verbs, such as REGULATE, will always cause other verbs to be included in the ‘map’.

* **A - Associate :** *Match, Compare, Join, Relate, Assemble* ....
* This activity-based object is utilized to identify new groups of Context-mechanism-outcome (CMO) data, as identified via the comparison of values.
* **C - Check sum :** *Audit, Re-Count, Accumulate, Add, Calculate, Compute, Summarize* ..
* This activity-based object is used to generate summary type values. When a KR Object includes a measurement of quality Checksum should be active, and vice-versa.
* **F - Filter :** *Partition, Replicate, Reformat, Bundle, Copy, Non-Warehouse CRUD (Create, Read, Update, Delete), Get, Hear (multi-media), Receive, Retrieve*...
* This activity-based object is the primary mechanism for changing the format and / or structure of existing data.
* **P - Prompt :** *Trigger, Event, Activate, Anticipate, Assign, Automate, Begin, Send, Transmit-Message, Deliver, Transport..*
* This activity-based object defines the WHEN of a KR Objecting. Every workflow objective has an active prompt.
* **R - Regulate :** *Control-actions, Adjust, Administer, Alter, Cancel, Reconcile* ..
* This activity-based object is used to facilitate the implementation of selective criteria - to make a choice. It often causes many of the other ‘verbs’ to become active.
* **T - Transform :** *Translate, Transliterate, Change, De-codify, (State-)Transition, Modify (fuzzy change*).
* This activity-based object is used to de-codify, make normal (cum de-normalization) existing data. It is often active when there is a need to create information that is destined for human-consumption.
* **Q - Queue :** *Wait (Non-State-Change*)
* This is a NON activity-based object. Its purpose is to explicitly record an in-active state.
* **W - Warehouse :** *Store-Append CRAD (Create, Read, Archive, Delete), Accept, Acquire, Attach, Build, Collect, Conserve, File, Keep, Memorize, Put aside, Remember, Save*.
* This activity-based object is a specialized FILTER that explicitly creates Intelligence Data. It should be active in the final KR Object of a Workflow Matrix.

**Business Terminology Lexicon**

Often during a JAD session there is a need to clarify exactly What type of data is being used and / or produced by a KR Object and exactly How an item of data gets converted into What is required. In particular, when the conversion of data assets produces a unique set of Business Analytic (that perhaps uses Virtual Data), the task of articulating the Context-mechanism-outcome (CMO) knowledge factors used in the process can initiate debates about semantics. The extent of such debates may be controlled with a judicious application of the Business Terminology lexicon. That is, whenever a new object (data-type or activity-based synonym) is discovered, the cataloging includes a mapping to the Business Terminology lexicon.

N.B.

If an object cannot be Business Terminology linked immediately THAT object must be considered as an ‘orphan’ that must be given a home before the KROSM session is allowed to terminate. That is, all objects must have been fully articulated before an agreement is reached.

**Objectively Mapping the Flow**

The overall intent of building a workflow matrix is to publish a fully detailed specification of how, the components of, one or more KR Objects are employed to produce Business Analytic that directly supports the objectives of consumers who have a Business Goal - as agreed by everyone who has a stake in the implementation. Thus, the mapping of a workflow is essentially the garnering of information about objective data manipulations and the linking of that to the needs of specific information consumers, i.e. KROSM maps metadata that are data-type objects and activity-based objects to metadata that are business objectives and have Context-mechanism-outcome (CMO) . For example :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***this is the -*** | **BUSINESS** | **GOAL** | **# 101 :** | ***CHANGE*** | ***PURPOSE*** |
| *this cell contains a unique ID number* | number 101. 1 | number 101.2 | number 101.3 | number 101.4 | number 101 .... |
| ***KR Object***  NAME | *This cell contains the name of a process* |  | *N.B. each concept / BR (business rule) must be explicitly defined if not known* | e.g.  IDENTIFY LOSS MAKING SALES | e.g.  IDENTIFY  LOSS MAKING SALES PERSON |
| ***Why ?***  OBJECTIVE |  | *This cell describes the reason for employing this process* | *N.B. each concept / BR (business rule) must be explicitly defined if not known* | e.g.  Identify What type of  SALES are  **not** **profitable**  **BR(**where  COST>SALES**)** | e.g.  Identify **who**  **BR(**HR#ID**)** is making SALES that are  not profitable |
| ***When ?***  PROMPT  (Trigger) |  | *This cell identifies the event that causes the*  *described processing to take place* | *N.B. each concept / BR (business rule) must be explicitly defined if not known* | e.g.  At the end of - Quarterly SALES  cycle | e.g.  At the end of -  (KR Object)  101.4 |
| ***What ?***  DATA-TYPE  part one  (*‘R’AW,*  *‘E’MBEDDED,*  *‘D’ERIVED,*  *‘V’IRTUAL,*  *‘I’NTELLIGENCE*) | *This cell describes what types of data are utilized or produced* | *This cell can also*  *identify the (unique) number of the process that generated the data* | *N.B. each concept / BR (business rule) must be explicitly defined if not known* | *e.g.*  *‘***D***’* (ERIVED) **86.1 :**  *PRODUCT SALES CODE* | e.g.  ‘**R**’  **who**  (defined in why) |
| *How ?*  ACTION  on DATA-TYPE  part one  (*ASSOCIATE, CHECKSUM, FILTER, REGULATE, TRANSFORM, QUEUE, WAREHOUSE*) | *This cell identifies the actions that are applied to the* DATA-TYPE  *part one* |  | *N.B. each concept / BR (business rule) must be explicitly defined if not known* | e.g.  **ASSOCIATE**  *Each PRODUCT*  *SALE CODE- with*  *PRODUCT COST CODE and*  **CHECKSUM**  ***not profitable*** | e.g.  **TRANSFORM**  HR#ID |
| *What ?*  DATA-TYPE  part two ....  (*‘R’AW,*  *‘E’MBEDDED,*  *‘D’ERIVED,*  *‘V’IRTUAL,*  *‘I’NTELLIGENCE*) |  | *This cell and other DATA\_TYPE cells are used to differentiate between the different types of data that are utilized and / or produced* | *N.B. each concept / BR (business rule) must be explicitly defined if not known* | *e.g.*  ‘**R**’(AW) **:**  *PRODUCT COST CODE* |  |
| *How ?*  ACTION on  DATA-TYPE  part two ...  (*ASSOCIATE, CHECKSUM, FILTER, REGULATE, TRANSFORM, QUEUE, WAREHOUSE*) |  | *This cell identifies the actions that are applied to the* DATA-TYPE  *part two ...* | *N.B. each concept / BR (business rule) must be explicitly defined if not known* | *e.g.*  **FILTER**  ***not profitable***  & **TRANSFORM** *PRODUCT COST CODE* |  |
| ***Who ?***  Consumer,  Owner,  Keeper,  Agent | *This cell identifies the people and the role they play in this process* & *Context-mechanism-outcome (CMO)* |  |  | e.g.  **Consumer** = VP MARKETING | e.g.  **Consumer** =  VP MARKETING |
| *Where ?*  ADDRESS  (logical and / or physical) |  | *This cell identifies the location of the process* |  | e.g.  email=vpmkt@  global.com | e.g.  email=vpmkt@  global.com |
| *Quality Metric*  *CHECKSUM* | *This cell identifies the form of measurement that can be applied to assess Quality* |  | *N.B. each concept / BR (business rule) must be explicitly defined if not known* | e.g.  **CHECKSUM**  (+/-difference)  = $‘-’ | e.g.  **CHECKSUM**  who =  alphanumeric |
| *Association ?*  KR OBJECT ID |  | *This cell is the link to another KR Object* |  |  |  |

Granularity of Workflow

An KROSM mapper can use varying levels of granularity (details) when populating a matrix. Indeed as a general guideline, in KROSM JAD sessions especially, a matrix should be developed at a high level at first, then decompose as required. That is, the intention is to gain a general agreement on the overall workflow, and as required, the contents of each KR Object is scrutinized for accuracy. In the same way new objects are Context-mechanism-outcome (CMO) cataloged, whenever KROSM participants are unable to agree upon the mapping of a Business Rule, then it should be tagged as an ‘orphan’ and resolved at a later point (often the mapping of other items can resolve the delay) - that is before the session is completed.

**Parallel Processing**

Often a decomposition will be introduced when there is a possibility of performing KR Objects in parallel. Indeed, parallel processing can also be introduced whenever a Business Rule has conditions that support the employment of two or more actions at the same time. Indeed, although parallel processes can rejoin at a later point, they do not have to. However, parallel processes do not ‘progress’ to points that, are before, or are the current step.

**Context-Mechanism-Outcome (CMO) Pattern Configuration**

All systems of business are reactive systems triggered by changes (real or estimated) in the environment of that business. the automation of that business, the actualizing of its processes, can be accomplished through observations of the current environment in a context. the intent of this paper is to prove that accomplishment.

**BASI:E: --> F L O model**

the full collection of artifacts and operations of a business is embedded in its ontology. We hold that an ontology by its definition\*\* is the collection (OC) of Operations Tokens (OPT:), Data Points\*\* (DP) in the most traditional sense of that concept, Eigen values (EV) and a matrix of relationships amongst all of these (MR). said another way an ontology is a model of data points, value points and execution (operation) tokens AND their inter-relationships. Said yet another way An ontology contains all of the information items in our business we care about AND the operations performed on these components.

<<why do we actualize - benefits>> ever increasing speed of technology, the competition's advances, market climate, skyrocketing fixed costs

*the way we actualize an ontology* is by the use of another ontology. an ontology devoted to the business of actualizing an ontology. the business of this second ontology is to perform those operations found in the business ontology in accordance with the matrix of relationships. given the abstract nature of any ontological discussion, we have opted (as many researchers have done before us)\*\* to use the subcategory names (metadata names), of which there are many, to aid in our discussion. the exposition should be clear enough to allow any reader the ability to associate these metadata names into their proper ontological partitions.

this other ontology (the actualizing ontology) may be located anywhere. it may be embedded in the primary (business of the business) one, or may be part of a larger federation of ontologies. the components of this ontology are described as the BASIE model. seen this way this WHOLE ontology (the combination), though it is a platform to be sure, is more than just a platform. seen in this way, though it is a model of the business, it is more than just a model. it is a platform for the actualization of a business that has been modeled.

A FORMAL DEFINITION

the BASI:E: --> F L O model states that the final (exposed) states (F), the output alphabet language (L), and the post production output operations (O) are produced by the set of beliefs (B) as tested at the intersection (I:) of the input alphabet (A) and the current set of states (S) where the status of S may have been produced by one or more executions of a permutation of the pre-process execution set (E)

where

B is the set of beliefs (sets of rules of behavior regarding truths)

A - the language (Alphabet) at the time of input.

S - the set of states (non-zero) before rule production

I: - the intersection where events, and conditions occur for a specific collection of states (Sx) AND for a specific element of the input alphabet (Ax) Ax in A. (I: can be considered the rule )

E: - the set of executions that may have preceded rule production.

F - the set of exposed states after production

L - the set of alphabet after production

O - the set of processes invoked (executed) after production

OBSERVATIONS AND ANALYSIS

in the real world there are time space events ([TSE]) that have a finite set of values. business demands that we make decisions based on the current status (values) of these events, yet all we have in our databases, data warehouses and stores are snapshots of these outside time space events at some past point in time.

for our discussion we declare that time is cyclical. t0 proceeded by t1 proceeded by t2 proceeded by t0.

where

t0 is that interval of time before we have captured the status (value of) some time space event.

t1 is the interval of time where we can use our collection of current values to make decisions AND influence our environment.

and where t2 is a period of time where we have allowed the environment to settle from the effects of our actions.

it is impractical and often impossible to carry around the real world item (TSE) to test its value. we hold a representation of it instead. these data points, as they are called, are representations of the value of this TSE. these data points are timeless in that they might not reflect the current value of the event they represent BUT when we perform some capture function immediately preceding the testing. this removes the temporariness restriction from these data points. for all of the interval of t1 they ARE the current value.

we have designated B as our belief network. for our discussion a belief set B (basian or otherwise) is a collection of rules (I:) based on the expectations of the status S of the system. This differs from the traditional view of belief sets which are the resultant sets of the application of the rules. Here, because we are interested in governance, we will aim our conversation on the rules not their products.

these rules I: are themselves an aggregate conclusion of the truthfulness or falseness of a collection of observable truths [Ix] derived from an executable (performable) function (E:) obtaining the current status of some TSE, designated Sx and some expected logical comparison (Opx) to some supplied value (Vx)

we consider Ix to be an observation point in that some value Vx associated with the state Sx will be compared logically to the current value at Sx and only the resultant truth (the signature of Ix ) will be carried into the final aggregation of I:.

**THE CYCLE OF EVENTS**

rule-1 perform all E. E’s are the transducer processes that gather the status of the universe. These states are recorded into variables [S] that corrispond to the element being recorded.

Rule-2 perform all I. I’s are the gathered facts about environmental conditions. Is it true that S(1) (a reflection of the outside air tempreture) is greater than 32?

Rule-3 perform all Ax([I]). Ax([I])’s are the logical response of all the facts. If it is true that I(1) is true AND ( I(2) or I(3) is not true) then Ax is true

Rule-4 apply all A to produce FLO. If A(z) is true then activate Oz. The resulting conditions will be:

- there will remain a set of states (FA) that will be activated during the next rule-3.

- There might be changes to the environment ([TSE]) which might also include data stored values (breadcrumbs). These will reflect changes during rule-1.

where do we start? (Scenario placement)

we truly begin with (ie our first determination is from) the set of operations

[Ex(TSEx)->Sx,Vx,Opx] -1-

which can be seen as a gathering of truths. this is the logical operation of Opx on the data point Sx which reflects the execution of a transducer operation (Ex) on some Time Space Event (TSEx). it should be noted that there is no limitation on the logical operation Opx, and that Vx can be a full formulaic observation yielding any conclusion. the observation may be concerning the numeric, character or logical factors of the result. further Vx may be of the form Vlow <= Vx <= Vhigh where Vlow and Vhigh may also be full observations.

using -1- we can deduce a collection of states [Ax] where

Ix([Ax],Sx,Vx,Opx) yields true or false. -2-

here Ix is the truthfulness or falseness of the operation of Opx on Sx and Vx. running this forward we can apply each state (Ax) to the rule

I:(Ax,Oz([Is-z])) -3-

where Ax is a set, Oz is a logical operator relating all of its parameters (all of which are truth indicators ([Is-z]) to yield a "Go" or "No Go" signal (the yield of Oz) which will produce (or not) the outputs [Fx],[Ox],[Lx].

after the application of I: we have phased into t1. at t1 the exposure functions will produce:

Fx --> Ax at t0 -4-

Ox(TSEx) --> Sx (at t0 viz-a-via Ex(TSEx) ) -5-

Ox(Lx) --> TSE(Vi) -6-

Ox(Lx)--> (TSEx) says that any new information exposed (created) will be in and of itself considered a time space event (TSE) and set to some initial value (Vi). bread crumbs marking a sequence of steps.

after the application of [Ox} we have phased into t2. a time period long enough to allow the universe (of discourse) to settle from any changes we (by application of Ox, or external sources may have created.

after t2 our cycle is complete and begins anew with t0. at each t0 for every exposed Ax we will collect the truth set [Ix] and apply them to the rule set (B) as above (I:(Ax,Oz([Ix]))