Training Programme Relationship Control

Updated 8-March-2013

Current CBT System

Our current CBT system of simple relations with minor sequencing ability does not fit with the requirements of Training in today's market. The CBT paradigm of Toolkit-Programme-Movie & Test, works well for un-sophisticated training requirements, but to compete with enterprise systems and SCORM packages as well as supporting External Training Courses, we need a more flexible approach.

Behavioural Modification Checks

This is based on Donald Kirkpatrick's level 3 of Training Evaluation. After Training has been taken, the trainee is observed to see if a behavioural modification has taken place and the training ideas put into practice. This is to be performed by an observer after a period of some days/weeks has passed since the training was taken, using a portable 'Workplace Observation Form'. This can be in the form of a printed page, an online web-page from the e-learning system or a dedicated 'App' on a smart-phone/tablet. The analysis of these forms (Kirkpatrick's Level 4) is beyond the scope of this system, although a simple scoring paradigm can be used as a guide to further action or in ROI discussions.

Workplace Observation Feedback

This ranges from formal pre-defined targeted form filling (work-place observation form), leading to a provisional score; through generalised quantitative feedback form (similar to 'check-a-trade'), to a qualitative feedback message used in the same way post-it notes are. These can be both internal and external observation, and formal or informal; for external observations, some possible editing will need to be available to aid dispute resolution.

Training Needs Requirement Analysis

This is the domain of the CMS, and will not be covered here in any detail. Suffice to say that this involves Training Goals and External Training Certification compared with Risk Assessment findings resulting in recommendations of required trainee training.

Compatibility with SCORM

Whilst reviewing our structures, methods and material; we need to take into account SCORM package requirements to future-proof our training systems. This may involve both SCORM package content and training-system control accessing external content.

Proposed Extensions

Equivalent Training Programmes and Training Goals

At present, Training Goals for a trainee can often be fulfilled by a number of equivalent training Programmes, due to foreign language and industry versions; for example the various 'Manual Handling' programmes in our system. This obstacle has become apparent when implementing 'Training Sets' based on Training Programmes, these need to change to use Training Goals instead. The Training Goals would consist of a hierarchical relationship between Goals and Programmes, in much the same way as Tool-kits are implemented.

Another way of tackling foreign language versions of programmes is to use multiple sound tracks with our video-clips and/or different language sub-titles provided on-the-fly; both of these options are possible with the new Flash Player and to some extent with HTML 5 Video, but only need be implement where required.

The concept of equivalent training of HFI programmes and External Training Certification is beyond the scope of our system, but we can have a generic dummy HFI 'programme' that can be 'force passed' to provide a mechanism for Goal Achievement.

Training Course Hierarchical Relations

To support the presentation of External Training Courses using our e-learning system as a delivery method, we need to add an extra layer of hierarchy to our training programmes and/or goals to implement this paradigm.

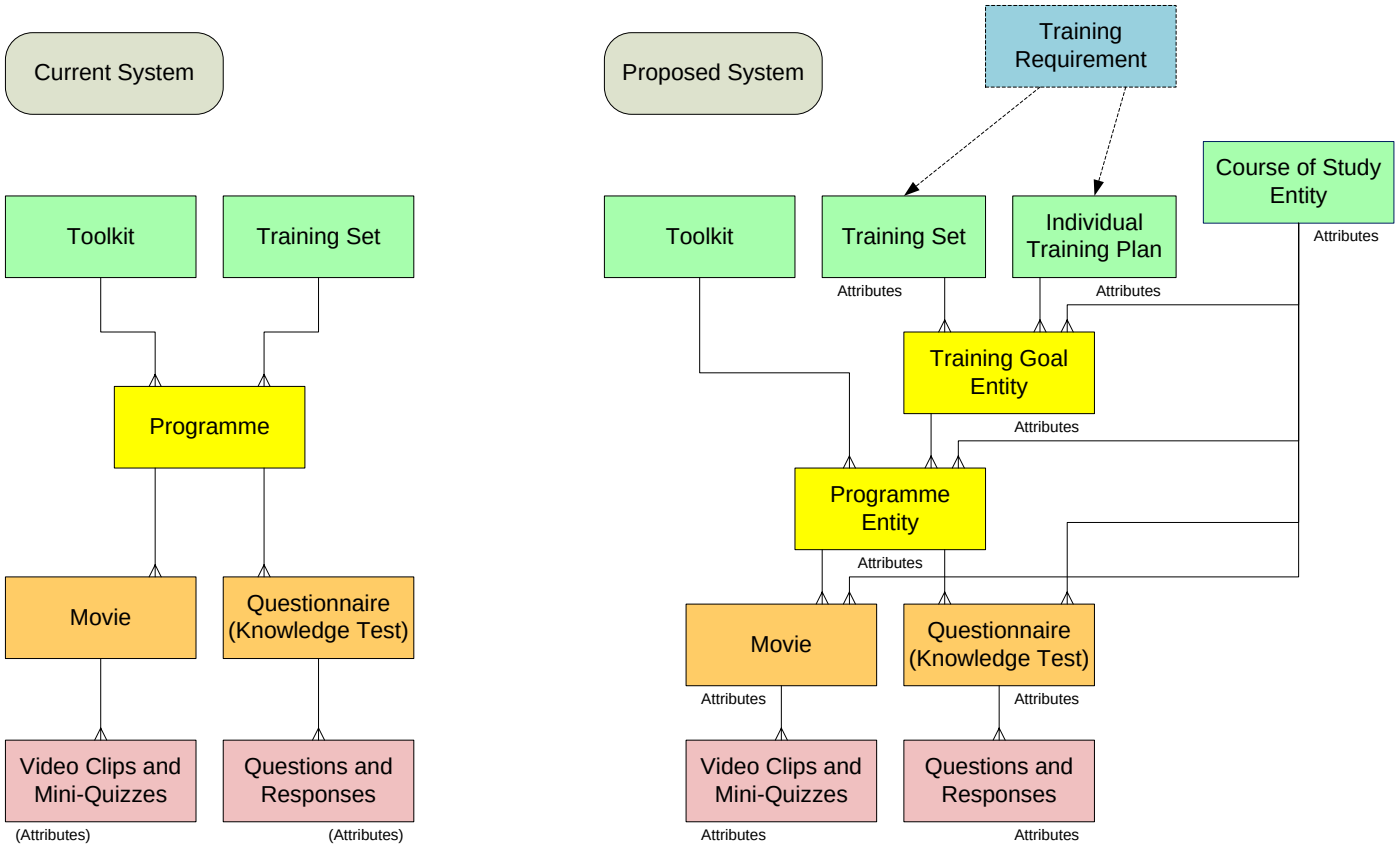
These 'Course of Study' units comprise a Syllabus of Programmes and a Praxis of Goals/Events. The programmes outlined in the Syllabus can be normal programmes that are treated differently depending on the requirements of the C-of-S. Similarly the Praxis of the goals will define the path of the constituent programmes with regards to sequence, equivalence and achievement. For example programmes may not provide a 'pass' certificate, and achieving a goal or Praxis movement, may provoke the sending of an e-mail to stake-holders. Praxis movement may involve external events such as expiry dates or risk-assessments, as well as peer-review of progress.

Training Set Expansion

Training Sets need revising in-line with the new concepts, the Syllabus exists but needs updating for Goals; and they also need the inclusion of Praxis, to help define dated events such as timed introductions, peer review, workplace observations, and refresher programmes. A set of training will contain a collection of Training-Requirements specific to a job type (Training Set) or for an Individual (Individual Training Plan).

Automated Training Content Review

This would allow targeted training content review based on feedback; or more simply video-clip study based on knowledge test wrong responses. This would entail using meta-data from both questions and video-clips to produce a review list; either hard coded or semi-automated. This is more appropriate for courses-of-study, where review material can be difficult to determine otherwise.



Comparison of e-Learning Systems

As can be seen the major impact on the system is at the 'Programme' level, but this should not cause too much interference with the 'Live' web-site as the changes will be by a top-down approach, with the higher Entity handling requiring new Pages. However the majority of the reporting pages will need to be re-written to show the different type of report data; training goals as well as training programmes.

The main changes to the system will be the introduction of Entity-Modelling with Entity-Instantiation and Tracking on a per-Trainee basis. This will include the attachment of 'Attributes' to all unit-stages of the model to control the transit of the trainee through the model; these Attributes will be modifiable by higher processes to change the default behaviour of the units. These Attributes are to replace the current ad-hoc approach of hard-coded web-page variation.

Enhanced Specification – Reserved Words

Assets

Also known as content, comprises: Movies, video-clips, questionnaires, mini-quizzes, knowledge tests, surveys, supporting documents; plus possible additions of slide-show presentations, audio streams, PDF documents, spreadsheets, external projects and pre-defined events.

Programmes

Programmes are generally a collection of assets, strung together sequentially made up of Parts, with sub-divisions of Sections, Items and Elements. A programme can have an outcome, a status and be certified, but can also be empty acting as a marker.

Goals

A Goal is generally a completed Programme from a list of Programmes, but may also be an Event. Goals can be achieved or forced.

Syllabus

A syllabus is a collection of Goals and Programmes, that may be partitioned by other collections, i.e. 'Tool-Kits'. A Syllabus can be completed (Course of Study), and scored (Set of Training) for comparison.

Events

Events are goal achievements, external influences and dated/timed spans or objects. Work place observations, Peer reviews.

Praxis

A Praxis is a controlling container that defines sequences, selection, repetition and options; for a Syllabus. Responding to Events, and defining User-Interfaces and reporting surfaces.

Certificates

Both Programmes and Courses of Study can be certified, whereas a Goal can only be supported by an underlying certificate from one or more Programmes. Certificates have both a Pass date and an Expiry date.

Time-Line

The attainment of a Certificate is not the end of training, there is also the checking of Behavioural Modification and refresher Programmes to consider; some training requirements do not end. For the more hazardous activities re-certification will be required on a regular basis.

Entity

Entities have attributes, and a life. They are created from an entity-model on a per Trainee basis.

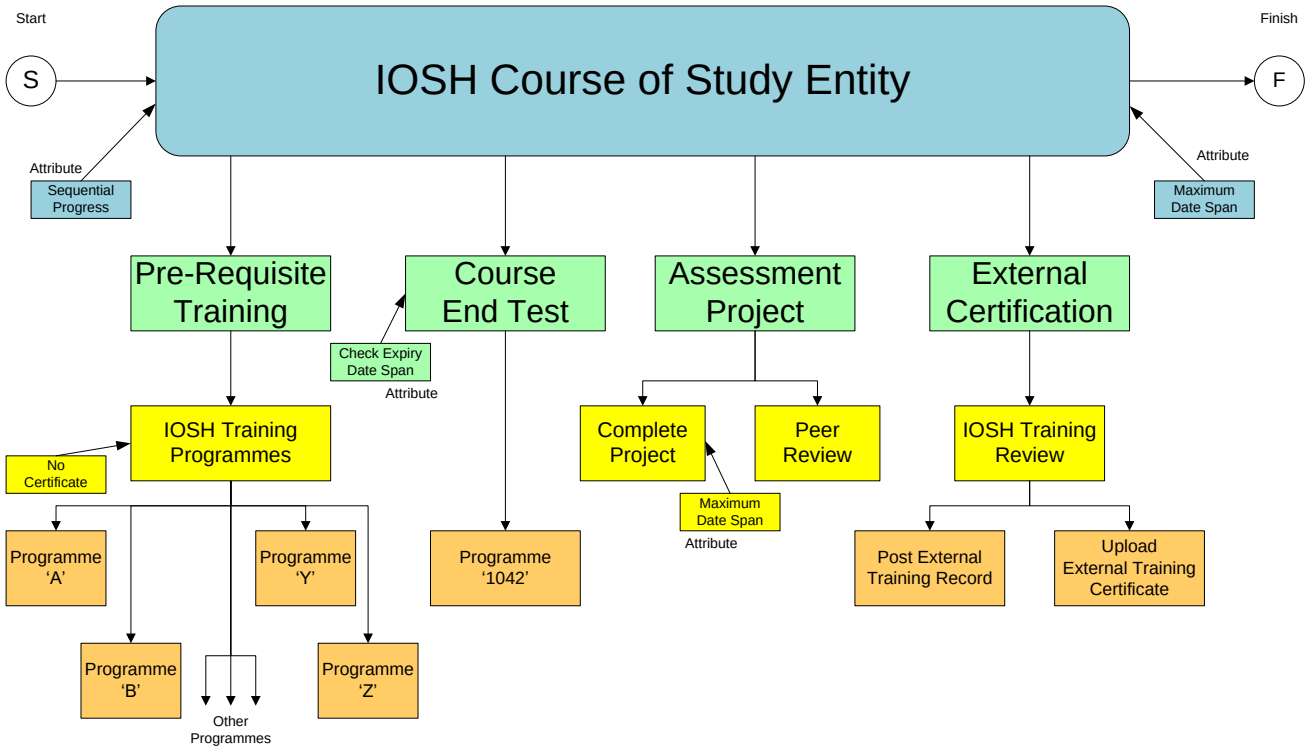
An entity records the progress through Praxis with regards to a Syllabus and a Timeline.

The life-time of a training requirement starts when an exposure to a hazard is identified for a Trainee, and runs until the Trainee is no longer exposed to the hazard; this is composed of a number of Training Goal Entities. Other types of entity exist, for example the Programme Entity and the Course of Study Entity.

An entity can be expressed as a tree diagrammatically, and constructed using XML structures physically with a model stored in the database, which is instantiated for a trainee when needed. Besides our current Response-Log and Training-Results tables, we will need an Entity-Praxis table to record the events and path taken through the Entity-Models by Trainees. Entity Models can also be called 'Classes' and the creation of Entities can also be called 'Object Instantiation' with the Entity then referred to as an 'Object'.

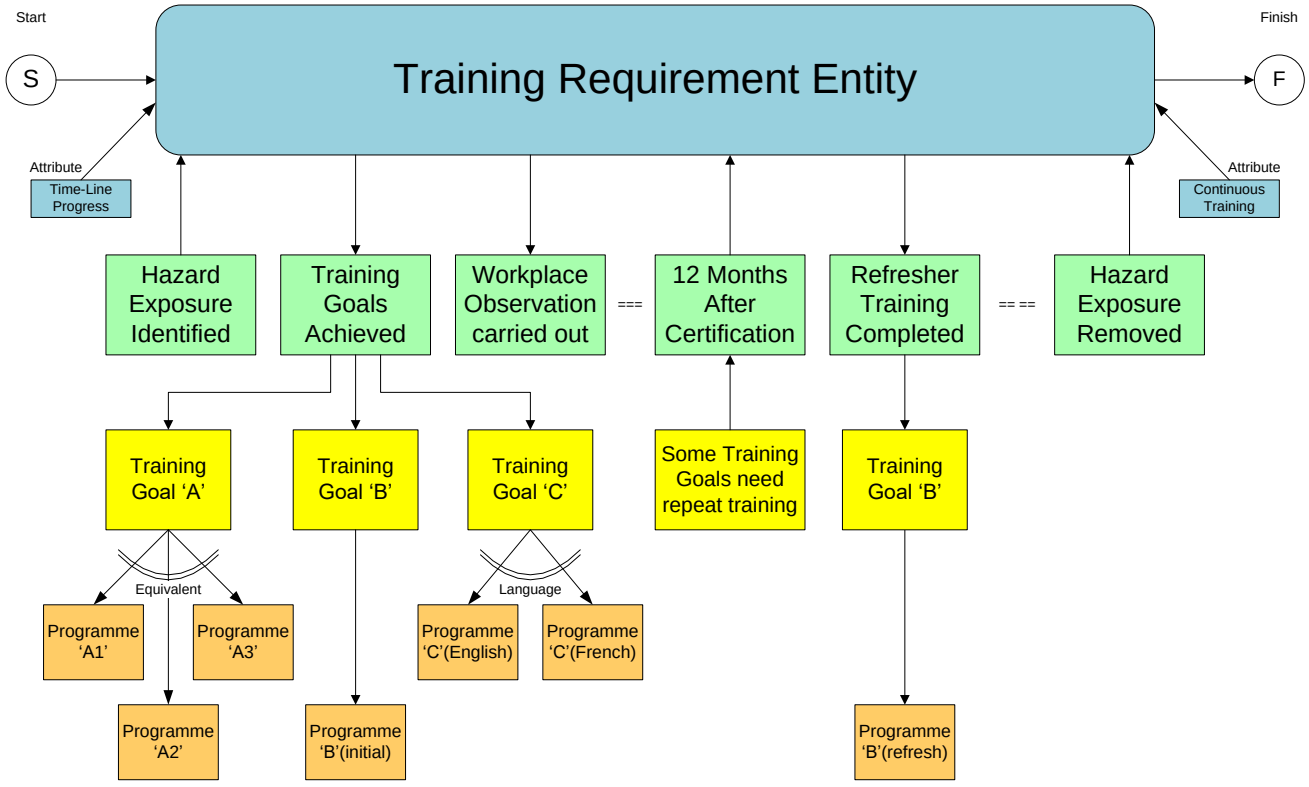
Examples of Entities

Example Course of Study Entity (IOSH)



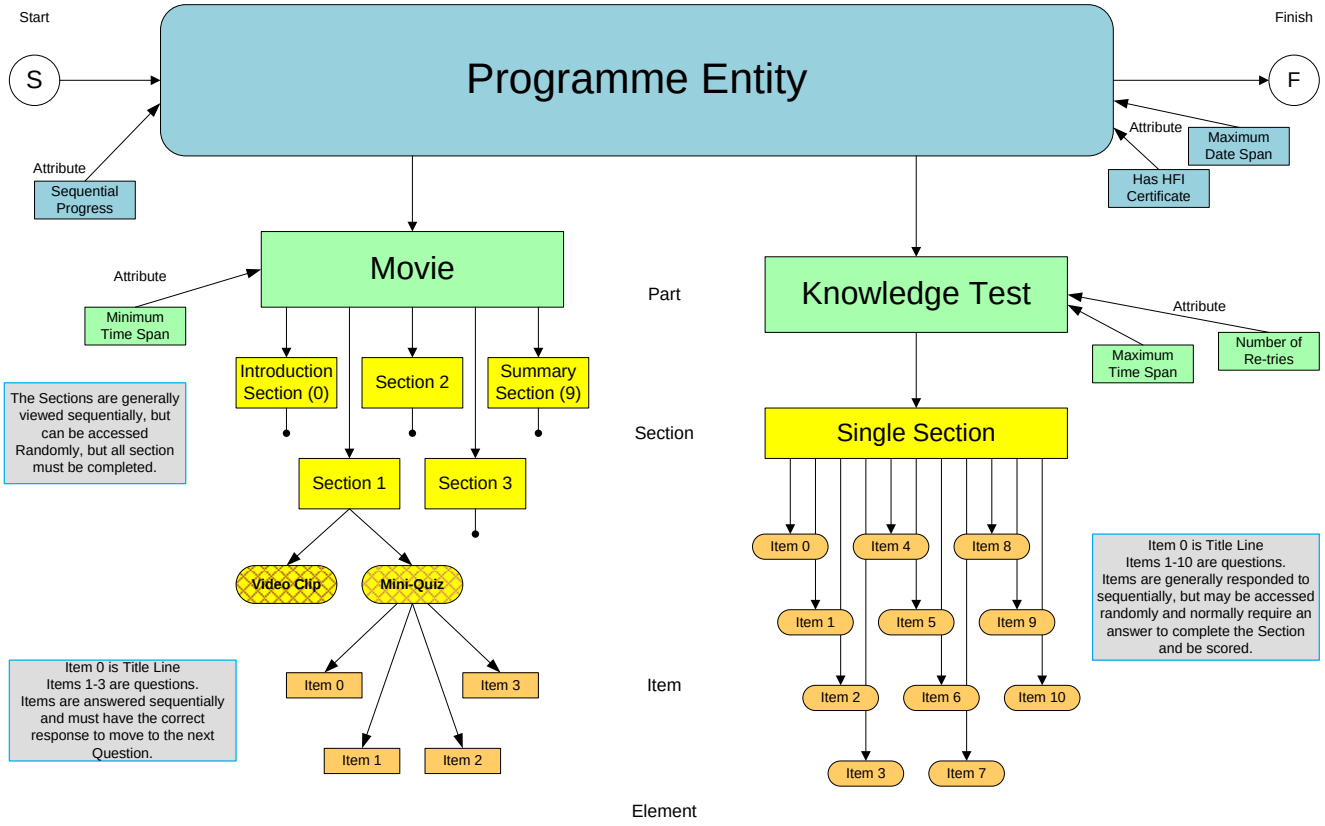
IOSH Course of Study Entity – Child Relations

Example Training Requirement Entity

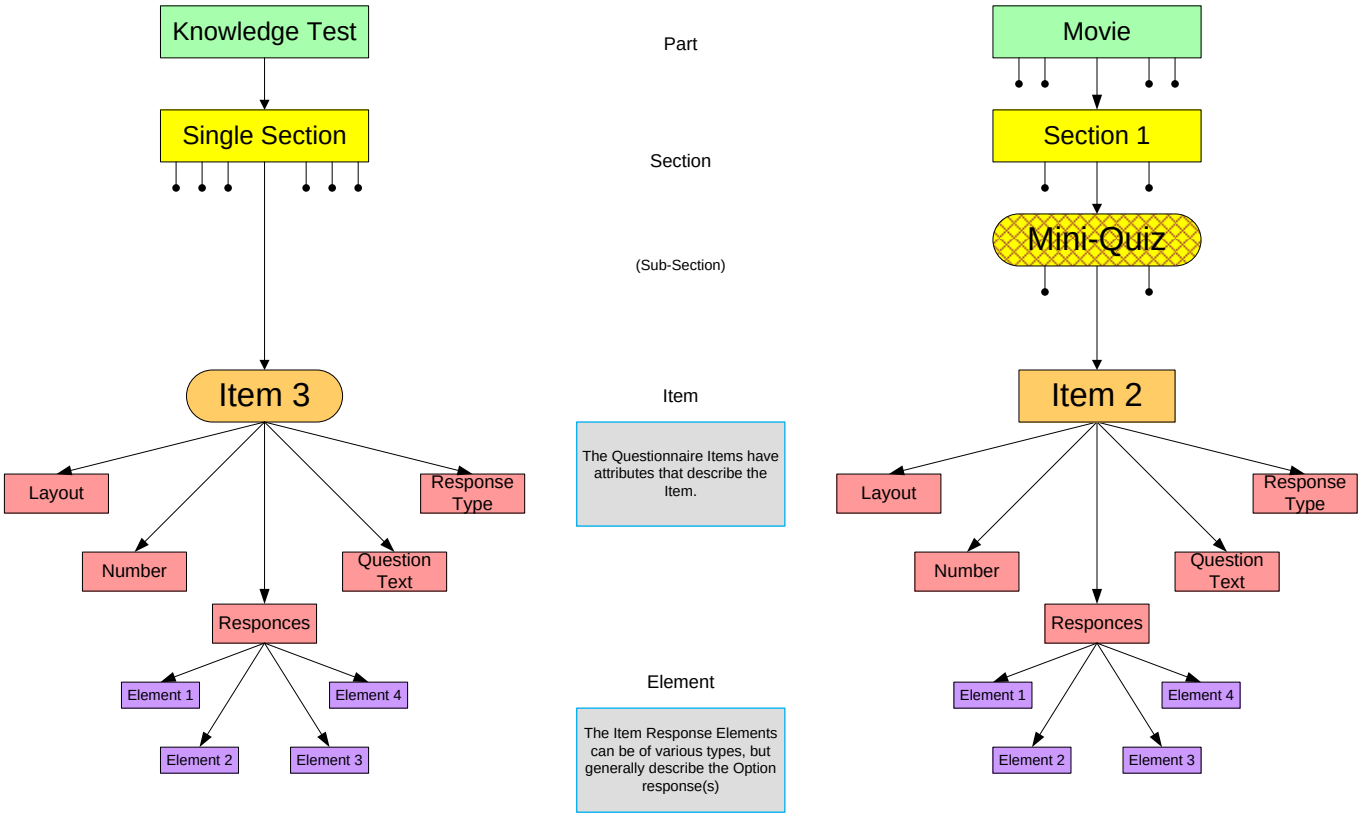


Training Requirement Entity – Child Relations

Example Programme Entity



Programme Entity – Child Relations



Programme Entity – Child Relations (Questionnaire Details)

Implementation of Entity Models

Entity Attribute Inheritance

Entity-Models have default behaviour that can be over-ridden by their attributes. Lower entity attributes can be over-ridden by higher entity attributes. Attributes have importance flags, higher importance attributes override lower importance attributes.

The Training Set

The current Training Set is just a list of training programmes, or for augmented clients a list of training programmes and external training courses. The new Training Set will also include training goals, and client modifiable attributes.

New database table 'CBT-List-Attr' to store the client modifiable attributes, the current table 'CBT-List' will be modified to handle Goals.

The Training Goal Entity Model

In its simplest form this is the same as a training programme, but can also define equivalent programmes for the purpose of achieving the defined goal, along with the ability to support on-going training. The 3 basic types of Training Goal entity praxis are, Training Goals (TG), Training Goals plus Workplace Observation feedback (TG+), and Training Goals plus Workplace Observation feedback plus Refresher Training Goals (TG++); a fourth type will use a model defined in XML to tackle more complex requirements. Training goals possess a status and a timeline, and have pre-defined models. Training Goals can semantically contain other Training Goals if required.

New database table 'Goal' to store the Entity-Models, 'Goal-Attrib' to store the Attributes, and 'Goal-List' to store the Programme links. A new table will be used to store the Trainee's Goal-Entity achievements, events and Praxis movements, 'Trainee-Goal-Praxis'.

The Programme Entity Model

The current CBT Programme consists of parts and sections, and is normally used for training with a movie part followed by a simple knowledge test part; but it can be used just to show a movie or to allow a survey to be taken or for a more complex knowledge test with a number of sections. The Movie part can be a single video-clip, or a number of video-clip sections interspersed with mini-quizzes to check attention is being taken of the material. The Programme Entity will do the same thing but will allow the overall control of the programme to be defined by Attributes; for example if the programme produces a Certificate or not, or the number of repeat tries allowed of a failed test, or a time-limit. A further expansion of this entity-model will be to allow access to its sectional objects for re-use by higher entities. Programmes can be certified. The basic Programme entity-models are hard-coded into a number of wrappers, with the required Programme entity-model varied by use of attributes stored in the database.

New database table 'CBT-Attrib' to store the Programme attributes, with table 'CBT-Part' to re-define the programme parts. The current results tables will be used to determine the Trainee's progress.

The Course of Study Entity

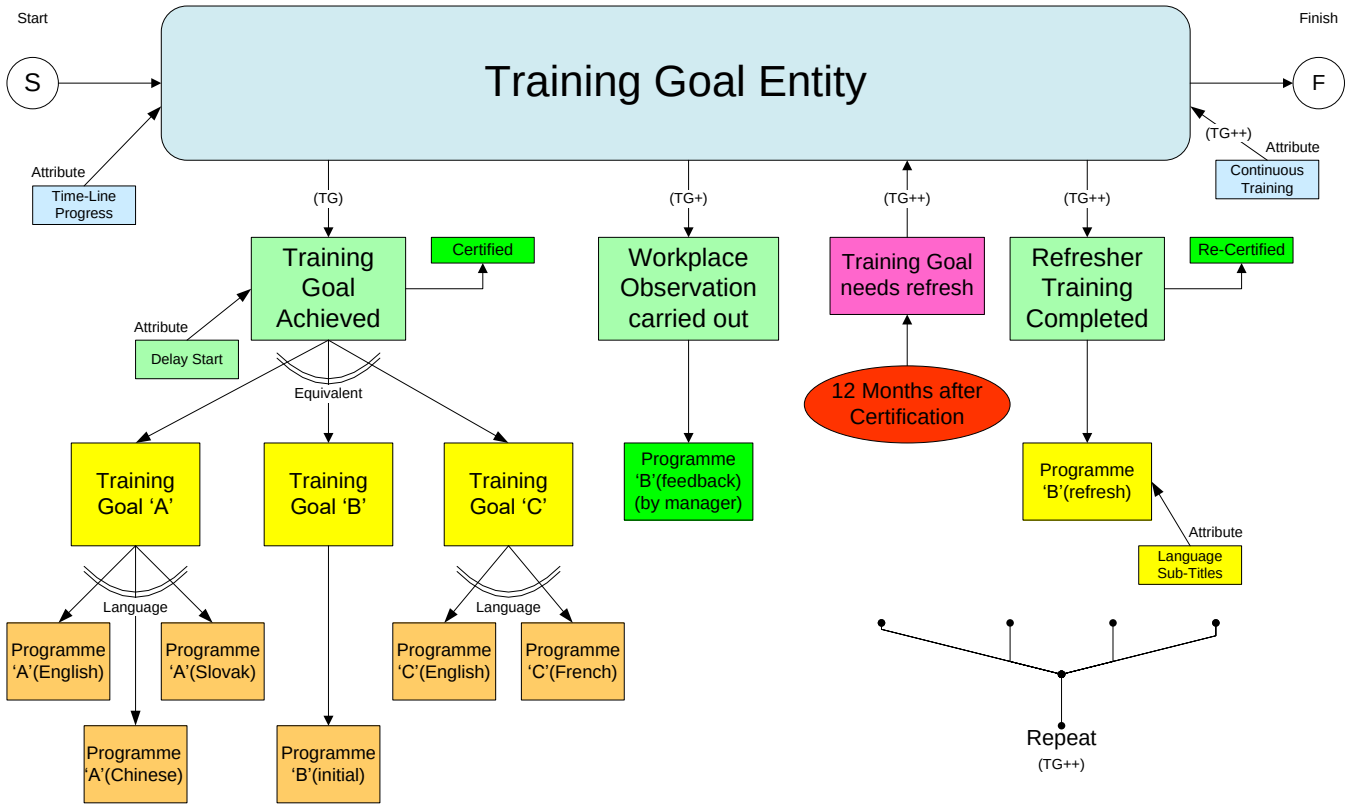
The Course of Study Entity is a pre-defined single use entity allowing complex control to be defined over the training process; it can be considered a 'Super' type of Training Set with sequencing and timeline elements. The entity-model can be constructed using XML and the entity-praxis by XML data or JSON data.

New database table 'CofS-Model' to contain the XML Model, and 'Trainee-CofS-Praxis' to contain the Trainee's progress.

The Training Requirement Entity

The Training Requirement Entity does not exist, it is an abstract idea of how the other entities co-exist and interact. It could however be used as a sales/marketing tool.

Training Goal Entity Example



Training Goal Entity – Child Relations and Actions

The status of this entity can change due to outside intervention as well as Trainee progress, and depends on viewer. Then the Entity is created, the trainee is first prompted to begin the training, and then the trainee carries out one of the available training programmes, achieves a Pass result and becomes 'Certified'. The manager is then prompted to perform a workplace observation programme a short time later, the results of which are digested elsewhere, but the certification becomes confirmed or approved. 12 months after the certification was obtained, the trainee is prompted to take the refresher training programme, and after getting a Pass becomes certified again. These processes can continue indefinitely or may come to a conclusion and the Entity destroyed; the certification is dependent on the programme results and not the entity. A new entity can be created by external assessments or training needs analysis as and when required.

Refresher training will only be available for a few pre-defined training goal entities, and not all training will require workplace observation feedback, for example introductions and participating Course-of-Study programmes.

Database Changes

New Data Dictionary Entries

GoalIdent	nvarchar(20)	primary key for Goal Entity table, often synonymous with CBT-Ident.	
CofSIdent	nvarchar(20)	Primary Key for Course of Study Entity table.	
ProgrammeIdent	nvarchar(20)	Primary key for Programme entity, may revert to CBT-Ident.	
TrainingGUID	uniqueidentifier		Workflow Instance identifier

New Fields for Existing Tables

tblTrainingGroup

TG_Attributes	nvarchar(max)		Client defined attributes as a packed string

tblCBTList

GoalIdent	nvarchar(20)	Primary Key	may be used to usurp CBT-Ident key.

tblCBT

CBTModel	nvarchar(max)	XML encoded Programme Entity Model

The majority of current CBT-Part fields in this table will be moved to the CBT-Part table.

Key Changes to Existing Tables

tblCBTList

OrganID	Int32	
TrainingSet	nvarchar(50)	
GoalIdent	nvarchar(20)	Insert new key field, default=N'none'
CBTIdent	nvarchar(20)	

New Tables

tblGoal

Goal Entity-Model representation along with the Goal-List table and their attribute tables.

GoalIdent	nvarchar(20)	primary key	often synonymous with CBT-Ident.
GoalParentIdent	nvarchar(20)	self-join	NULL or self, for top-of-tree
GoalTitle	nvarchar(250)		title of goal
GoalModel	nvarchar(max)		XML encoded Goal Entity-Model

tblGoalList

GoalIdent	nvarchar(20)	primary key	foreign key
CBTIdent	nvarchar(20)	primary key	foreign key

tblCBTPart

Programme Entity-Model representation along with existing CBT table and their attribute tables.

CBTIdent	nvarchar(20)	primary key	foreign key
CBTQuizID	Int32	Primary Key	foreign key
CBTPartNo	Int16	Primary Key	
CBTPartModel	nvarchar(max)		XML encoded Programme-Part Entity-Model

tblCofS

Course of Study Entity-Model representation along with the attribute table.

CofSIdent	nvarchar(20)	Primary Key	
CofSTitle	nvarchar(250)		Title of course
CofSModel	nvarchar(max)		XML encoded Course of Study Entity-Model

tblTrainingEntity

Trainee Workflow table, records the progress of training through the model, with up to 4 levels of hierarchy.

OrganID	Int32		foreign key
UserID	Int32	Primary Key	foreign key
EntityIdent	nvarchar(20)	Primary Key	foreign key
InceptDTS	datetime	Primary Key	for this instance only
EntityGUID	uniqueidentifier		Workflow Instance identifier, can be null
ParentEntityGUID	uniqueidentifier		Parent Workflow Instance identifier or null for root
EntityType	nvarchar(20)	(look-up?)	foreign key? (CofS, Goal, Programme, Prog-Part)
EntityPraxis	nvarchar(max)		XML encoded Training Entity Praxis

tblTrainCofS Trainee Course of Study, records Trainee progress through the Entity's model and other events. Replaced by tblTrainingEntity table.
tblTrainGoalTrainee Goal achievement, records Trainee progress through the Entity's model and other events. Replaced by tblTrainingEntity table.

New Attribute Tables

These are Fifth-Normal fields belonging to their associated tables. Takes the form of key-fields followed by attribute fields, in a similar way to Extra Data table fields. Attributes are used to modify the behaviour of the Entity-Model. Attribute flags bits 0 and 1 are used to determine the importance of the attribute relative to other attributes from other entities.

...		Primary Keys	Parent primary keys
...A_Name	nvarchar(20)	Primary Key	attribute name
...A_Type	nchar(10)		attribute data-type
...A_Data	nvarchar(max)		attribute data
...A_Flags	int16		attribute flags

tblCBTList-Attr

OrganID	Int32	Primary Key	
TrainingSet	nvarchar(50)	Primary Key	
GoalIdent	nvarchar(20)	primary key	
CBTIdent	nvarchar(20)	Primary Key	
CBTListA_Name	nvarchar(32)	Primary Key	attribute name
CBTListA_Type	nchar(10)		attribute data-type
CBTListA_Data	nvarchar(max)		attribute data
CBTListA_Flags	int16		attribute flags

tblCBT-Attr

CBTIdent	nvarchar(20)	Primary Key	
CBTA_Name	nvarchar(32)	Primary Key	attribute name
CBTA_Type	nchar(10)		attribute data-type
CBTA_Data	nvarchar(max)		attribute data
CBTA_Flags	int16		attribute flags

tblCBTQuiz-Attr

CBTQuizID	Int32	Primary Key	
CBTQuizA_Name	nvarchar(32)	Primary Key	attribute name
CBTQuizA_Type	nchar(10)		attribute data-type
CBTQuizA_Data	nvarchar(max)		attribute data
CBTQuizA_Flags	int16		attribute flags

tblQuizSect-Attr

CBTQuizID	Int32	Primary Key	
QuizSection	Int16	Primary Key	
QuizSectA_Name	nvarchar(32)	Primary Key	attribute name
QuizSectA_Type	nchar(10)		attribute data-type
QuizSectA_Data	nvarchar(max)		attribute data
QuizSectA_Flags	int16		attribute flags

tblGoal-Attr

GoalIdent	nvarchar(20)	primary key	
GoalA_Name	nvarchar(32)	Primary Key	attribute name
GoalA_Type	nchar(10)		attribute data-type

GoalA_Data	nvarchar(max)		attribute data
GoalA_Flags	int16		attribute flags

tblGoalList-Attr

GoalIdent	nvarchar(20)	primary key	
CBTIdent	nvarchar(20)	primary key	
GoalListA_Name	nvarchar(32)	Primary Key	attribute name
GoalListA_Type	nchar(10)		attribute data-type
GoalListA_Data	nvarchar(max)		attribute data
GoalListA_Flags	int16		attribute flags

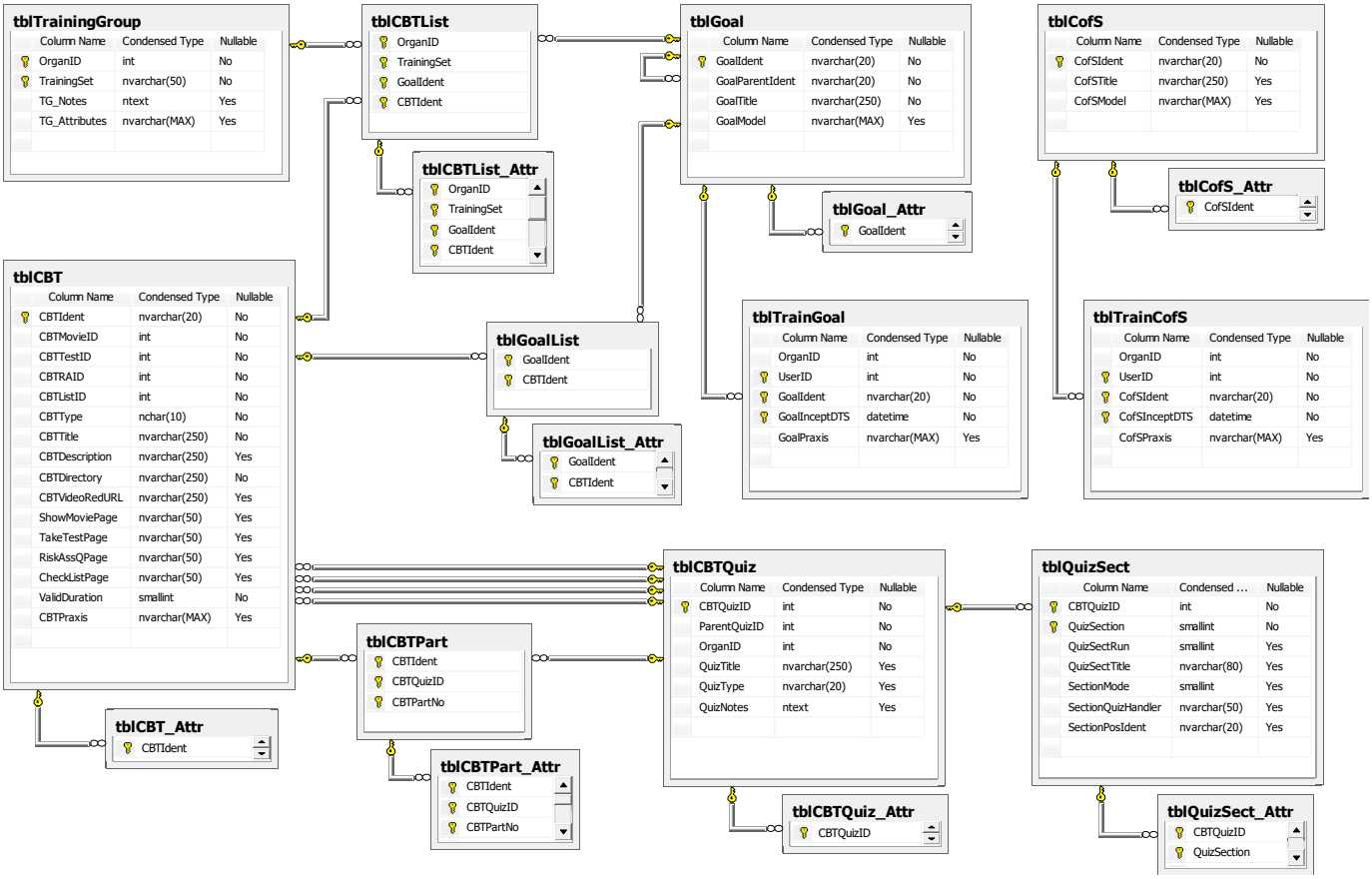
tblCBTPart-Attr

CBTIdent	nvarchar(20)	Primary Key	
CBTQuizID	Int32	Primary Key	
CBTPartNo	Int16	Primary Key	
CBTPartA_Name	nvarchar(32)	Primary Key	attribute name
CBTPartA_Type	nchar(10)		attribute data-type
CBTPartA_Data	nvarchar(max)		attribute data
CBTPartA_Flags	int16		attribute flags

tblCofS-Attr

CofSIdent	nvarchar(20)	Primary Key	
CofSA_Name	nvarchar(32)	Primary Key	attribute name
CofSA_Type	nchar(10)		attribute data-type
CofSA_Data	nvarchar(max)		attribute data
CofSA_Flags	int16		attribute flags

Diagram of Entity Tables



Events Log

This will be stored in the database as a collection of linked tables, to form an audit trail of Entity's events, and will be referred to by the Entity's Praxis fields.

Physically the Events Log will be made up of 3 tables, tblEventLog, tblEventLog-Extra and tblEventLog-Reason; with an auto-generated EventLogID number. The first table may be split into two to follow database 3rd normal rules, adding tblEventLog-EntityStack. The Entity-Stack refers to all the 'entity-praxis' partaking in an event.

First Cut Database Tables

tblEventLog

Separate combined field unique key 'x'. Entity-Stack of fields 'y', from 0 to 5 fields in use for any event.

EventLogID	int32	primary key	autonum
OrganID	int32	foreign key	
UserID	int32	foreign key	unique combo key x
EL_ReasonCode	int16	secondary key	unique combo key x, look-up table with event titles
EventLogDTS	datetime	secondary key	unique combo key x
CofSIdent	nvarchar(20)	foreign key	y, from CofS Entity-Model, with object data in ELX or blank
GoalIdent	nvarchar(20)	foreign key	y, from Goal Entity-Model, with object data in ELX or blank
EL_ProgIdent	nvarchar(20)	foreign key	y, Programme Entity-Model, object data in ELX, CBTIdent?
EL_ProgPart	int16		y, from CBTPart table or derived from CBT table or null
EL_ProgSection	int16		y, from QuizSection table or null

tblEventLog-Extra

This table contains extra data fields relating to the event, for example relevant Entity Inception DTS key fields.

EventLogID	int32	foreign key	combined primary key
ELX_Group	nvarchar(32)		combined primary key
ELX_Name	nvarchar(32)		combined primary key
ELX_Type	nchar(10)	foreign key	to look-up table
ELX_Data	nvarchar(MAX)		
ELX_DTS	datetime		auto-fill

tblEventLogReason

This is a look-up table for the Event-Log table with details of the type of event.

EL_ReasonCode	int16	primary key	
ELR_Title	nvarchar(32)		the name of the event
ELR_Description	nvarchar(250)		a description of the event

tblEventLog-EntityStack

This table may be used to contain the Event-Log Entity-Stack from the Event-Log table, 3rd normal form.

EventLogID	int32	foreign key	combined primary key
ELES_RunNo	int16		combined primary key
ELES_EntityIdent	nvarchar(20)	foreign key	from relevant Entity-Model/Object
ELES_InceptDTS	datetime	foreign key	from relevant Entity-Object, with object details in ELX
ELES_Type		look-up?	Entity type+ (CofS, Goal, Programme, Part, Section)

Classes

Top Entity (C of S)

Goal Entity

Programme Entity

CBTbyB Class

Methods

Initialise, Prog-Constructor () return object [prog] from session

Create a new trainee object, Prog-Constructor (UserID, CBTIdent) return object [prog], overloaded for higher entity

Continue an existing trainee object, Prog-Constructor (UserID, CBTIdent, InitialDTS) return object [prog], overloaded for higher entity

Done something (object, event) return object [next-step]

Kill Prog (UserID, CBTIdent, InitialDTS) return bool [success], overloaded for higher entity

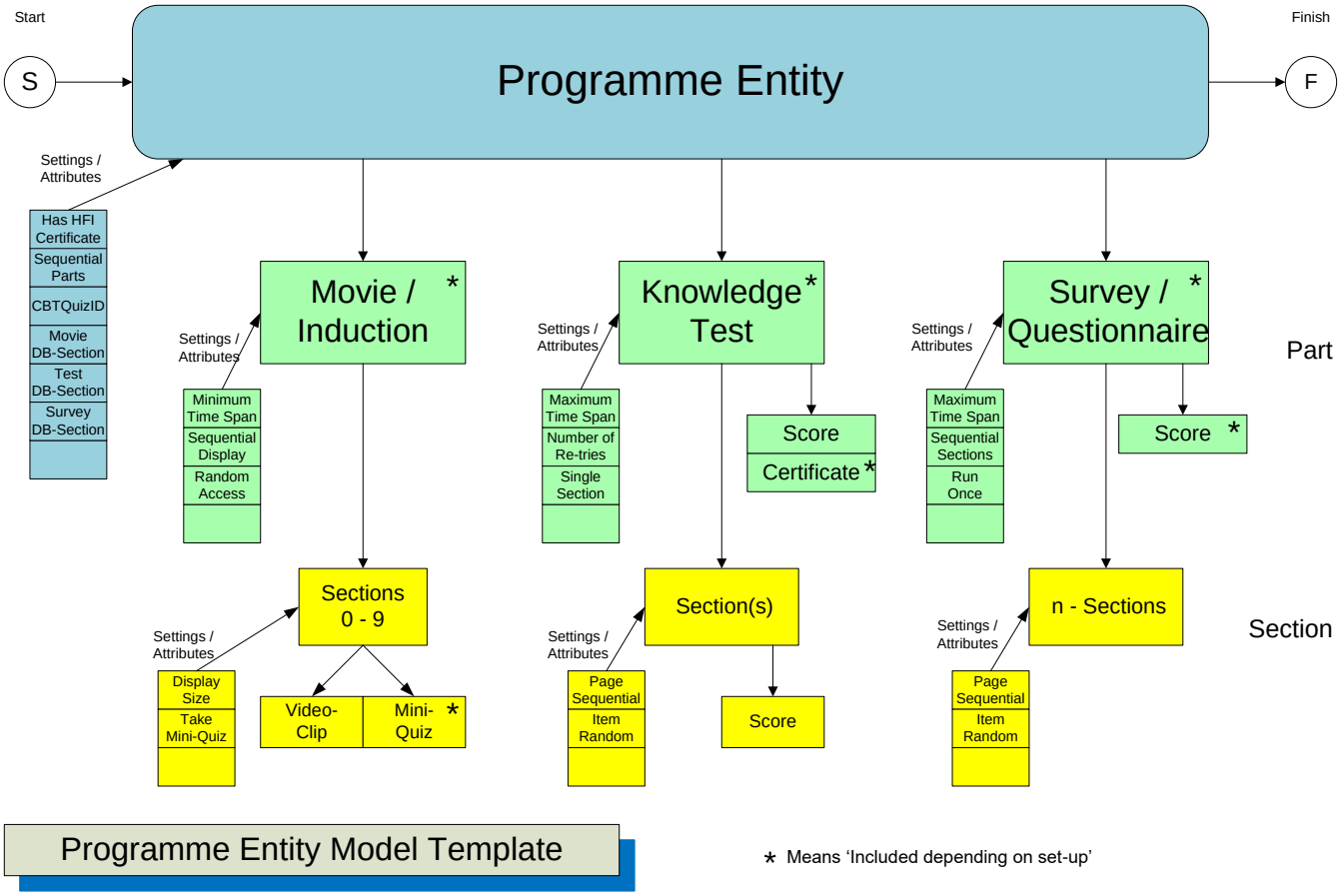
Find Prog (UserID, CBTIdent, InitialDTS) overloaded various versions, return object [prog] / list of missing param

Do something(object) return object [next-step]

Implementation of Part-Wrapper

Programme Entity Model Template

Movie part + Knowledge Test part + Survey part



Use same QuizID, movie data starts at section=10, test data starts at section=1, survey data starts at section 1, single section test, produces a certificate, movie has a minimum time = 10min, number of retries allowed = 2, identifier = CBTIdent.

Programme Workflow

start,

redirect to first part page (movie),

do first part (movie),

done first part (movie),

redirect to second part page (test/survey),

do second part (test/survey),

done second part (test/survey),

redirect to third part page (risk),

do third part (risk),

done third part (risk),

redirect to fourth part page (checks),

do fourth part (checks),

done fourth part (checks),

finish

Workflows have a GUID to identify object instance 'Instance-ID', we will need to obtain this from the Workflow Instance database store or store it in the Trainee's records, possibly with the Entity-Praxis. WorkflowInstanceGUID : WFInstanceGUID uniqueidentifier Workflow Instance identifier