

Knowledge structure

Introduction

In order to facilitate the management of contents by Humane, two knowledge structures will be used. The first one is an hierarchy going from the general (Theme) to the specific (Affair) topic of study. Being an hierarchy implies contention: the level below is comprised on the upper levels. The second structure are contents and parts. They are independent objects and are connected by use, not ownership.

In the next sections all objects will be detailed to the data dictionary level.

Objects description

Attributes shared by all objetscs:

- Status: it allows for the object life cycle management. No object is ever deleted.
 - Value: it identifies the current status. The values are restrained by a value list. The value is defined based on the available status dates.
 - Creation date: date when the object was created.
 - Testing start date: date when the object started the testing phase.
 - Activation date: date when the object was released to use.
 - Retirement date: date when the object was retired.
- Detailed description: It allows for an in depth description of the object.
- Specific: When the object is specific for a client, this attribute will show the client name.

Hierarchy:

- **Level 1 Themes:** É o nível mais alto da hierarquia de conteúdos. Tanto pode representar um processo logístico (ex. picking), quanto uma área de conhecimento importante para a operação (ex. segurança). Os temas são concretizados como categorias no LMS. Can be updated after passing into "active" status if the change is minor. Otherwise, a new theme must be created.
 - **ID:** It identifies each object individually and is defined automatically by the system. Starts with the first letter of the object type ("T") and is followed by a sequential number.
 - **Name:** it is used by all systems when showing the object.
- **Level 2 Courses:** São subdivisões de um tema. Os cursos podem ser gerados por: métodos distintos para realizar o mesmo processo (ex. picking de caixa na posição palete por pedido cliente), implementações distintas pelos clientes de um mesmo process (ex. picking unitário na L'Oréal), especificidades de um tipo de mercadoria (ex. segurança no transporte de explosivos), necessidades específicas de uma população (ex.: iniciante, avançado). Um curso pode ter antecessores obrigatórios, criando um currículo. Para tanto, basta informar as variantes antecessoras na coluna "Dependência". Essa informação será utilizada no LMS. Can be updated after passing into "active" status if the change is minor. Otherwise, a new course must be created.
 - **ID:** It identifies each object individually and is defined automatically by the system. Starts with the first letter of the object type ("C") and is followed by a sequential number.
 - **Name:** it is used by all systems when showing the object.
 - **ThemeID:** The ID of the parent theme.
 - **Dependencies:** CourseID of courses that are pre-requisites to this one. When multiple values, use comma as separator.
 - **Category:** Indicates the reason for the course existence. It is constrained by a value list.
- **Level 3 Courses versions:** They are the actual course created at the LMS. After passing into status "active", versions can not be changed. Instead, a new version must be created. This includes changes to the version affairs

and its contents.

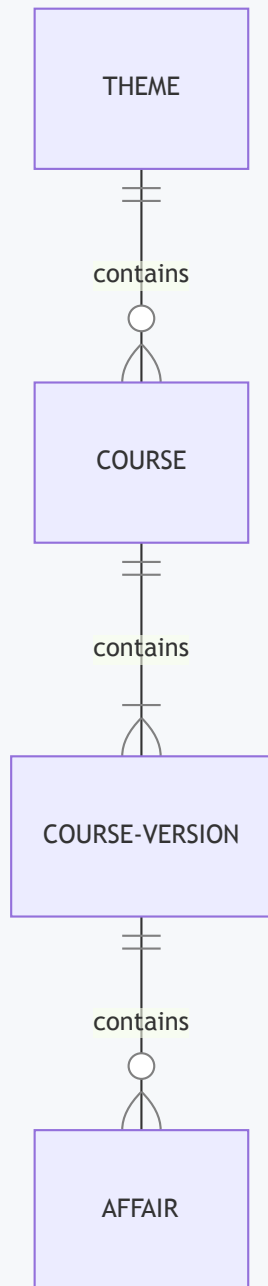
- ID: It identifies each object individually and is defined automatically by the system. Starts with the first letter of the object type ("V") and is followed by a sequential number. A different sequence is followed for each Course.
- CourseID: The ID of the parent course.
- Level 4 Affairs: Os assuntos são a trilha de aprendizado de uma versão de curso. Um assunto define um ponto a ser ensinado ao profissional (ex. Visão geral do processo de picking) no escopo do curso. Os assuntos de um curso devem ser ordenados, a fim de estruturar uma trilha de formação coerentemente. Para tanto, basta preencher o campo "Ordem" com o número ordinal de cada assunto dentro da variante. The object cannot be changed after achieving the "active" status. Instead, a new object must be created.
 - ID: It identifies each object individually and is defined automatically by the system. Starts with the first letter of the object type ("A") and is followed by a sequential number.
 - Name: it is used by all systems when showing the object.
 - CourseVersionID: The ID of the parent version.
 - Order: It defines the order in which the affair will be present inside a course version.
 - Bill of contents (BOC): The list of ordered comma-separated contentIDs used to taught the affair. This list, combined with the order of affaires on the course version will be used to create the course content list in the LMS.

Materials:

- Content: Um conteúdo é um veículo de informação concreto utilizado para ensinar parte de um assunto ao profissional (ex. O que é o picking por pedido?). Conteúdos podem ser de diferentes tipos, indo do texto simples ao vídeo, passando por testes. Um mesmo conteúdo pode ser utilizado por mais de um assunto. The object cannot be changed after achieving the "active" status. Instead, a new object must be created.

- ID: It identifies each object individually and is defined automatically by the system. Starts with the first letter of the object type ("C") and is followed by a sequential number.
- Name: It is used by all systems when showing the object.
- Media: It defines the type of media file (ex. video, audio, text, etc). It is constrained by a value list.
- Complete path: Path to access the file.
- Bill of parts (BOP): The list of ordered comma-separated PartIDs used to create the component.
- Parts: A parte pode ser entendida como peças pré-fabricadas utilizadas na montagem de conteúdos. Seu objetivo é reduzir o esforço na construção de conteúdos através do reuso (ex. um pedaço de uma animação, um slide de uma apresentação, uma imagem). The object cannot be changed after achieving the "active" status. Instead, a new object must be created.
 - ID: It identifies each object individually and is defined automatically by the system. Starts with the first letter of the object type ("P") and is followed by a sequential number.
 - Name: It is used by all systems when showing the object.
 - Media: It defines the type of media file (ex. video, audio, text, etc). It is constrained by a value list.
 - Complete path: Path to access the file.

Diagrams





is composed of



is composed of

