5S

Ricardo Torres

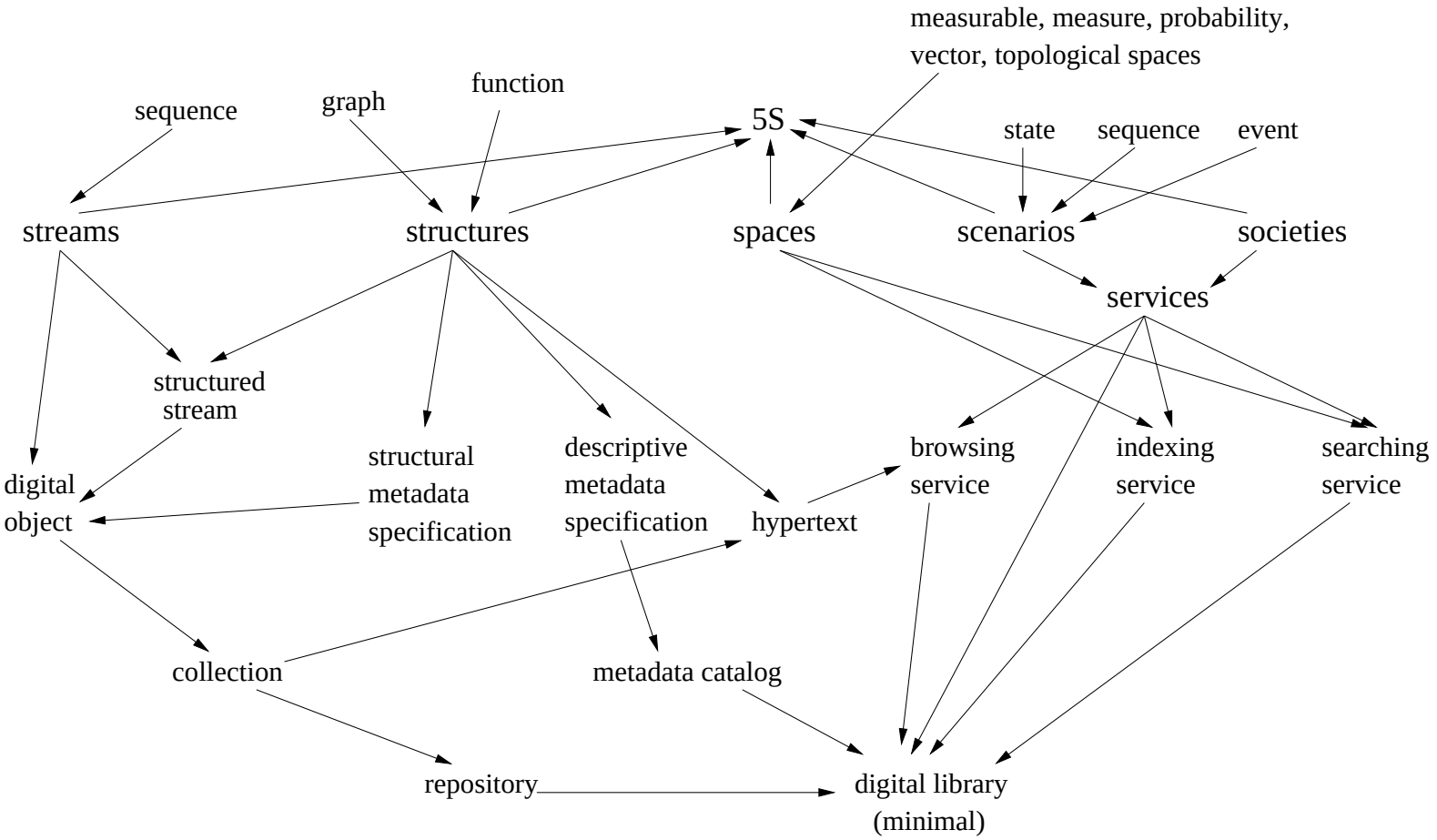
rtorres@ic.unicamp.br

Overview

Goal: 5S definitions
Main concepts:

- Streams
- Structures
- Spaces
- Scenarios
- Societies

5S Map of Formal Definitions



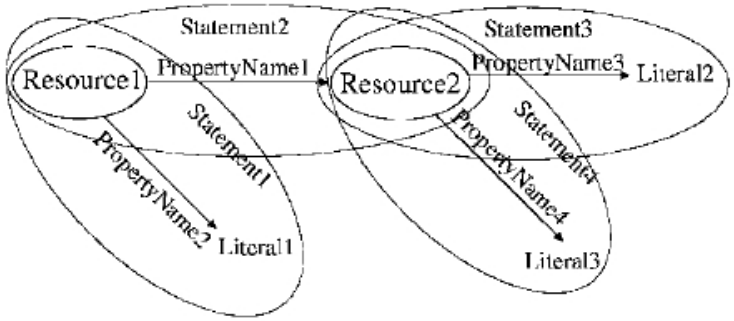
Stream

- A **stream** is a $sequence$ whose codomain is a nonempty set.
- A **sequence** is a function f whose domain is the set of natural numbers or some initial subset $\{1, 2, ..., n\}$ of the natural numbers and whose codomain is any set.

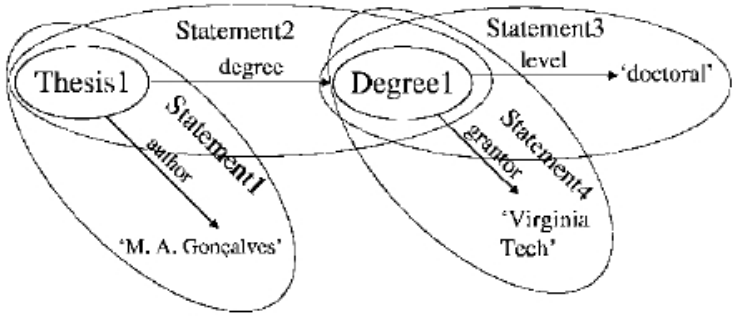
Structure

- A **structure** is a tuple $(G, L, \mathcal{F})$, where $G = (V, E)$ is a directed graph with vertex set V and edge set E, L is a set of label values, and $\mathcal{F}$ is a labeling function $\mathcal{F} : (V \cup E) \rightarrow L$.
- A **substructure** of a structure $(G, L, \mathcal{F})$ is another structure $(G', L', \mathcal{F}')$ where $G' = (V', E')$ is a subgraph of G, $L' \subseteq L$ and $\mathcal{F}' : (V' \cup E') \rightarrow L'$.

Structure



(a)



(b)

Structure

```
<?xml version="1.0" encoding="UTF−8"?>
−<feature_vector:Feature_Vector xmlns:feature_vector="http://feathers.dlib.vt.edu/~rtorres/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema−instance"
    xsi:schemaLocation="http://feathers.dlib.vt.edu/~rtorres/
    http://feathers.dlib.vt.edu/~rtorres/feature_vector.xsd">
    <feature_vector::ImageName>fish0.pgm</feature_vector:ImageName>
    <feature_vector:DescriptorName> ContourMSFractalDimension <feature_vector:DescriptorName>
    <feature_vector:Type> 1 <feature_vector:Type>
    −<feature_vector:Curve>
        <feature_vector:Nelements> 25 <feature_vector:Nelements>
        −<feature_vector:Curve1D>
            −<feature_vector:X>
                <feature_vector:value> 0.95105259594482394192 <feature_vector:value>
                <feature_vector:value> 0.98551214588154611995 <feature_vector:value>
                <feature_vector:value> 1.00415492765507829986 <feature_vector:value>
                <feature_vector:value> 1.00931032237937512441 <feature_vector:value>
                <feature_vector:value> 1.00583781572741104426 <feature_vector:value>
                                    (...)
                <feature_vector:value> 0.93810555611087775851 <feature_vector:value>
                <feature_vector:value> 0.87275204902189629230 <feature_vector:value>
                <feature_vector:value> 0.81066432563100665476 <feature_vector:value>
                <feature_vector:value> 0.75224263059381879515 <feature_vector:value>
            </feature_vector:X>
        </feature_vector:Curve1D>
    </feature_vector:Curve>
</feature_vector:Feature_Vector>
```

Space

- A **space** is a measurable space, measure space, probability space, vector space, topological space, or a metric space.

Example: Metric Space

A metric space is an ordered pair (M, d) where M is a set and d is a metric on M, that is, a function
$d : M \times M \to \mathbb{R}$
such that for any x, y and z in M

1. $d(x, y) \geq 0$ (non-negativity)
2. $d(x, y) = 0$ if and only if $x = y$ (identity)
3. $d(x, y) = d(y, x)$ (symmetry)
4. $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality).

State

- A **system state** (from now on, just state) is a function $s : L \rightarrow V$, from labels L to values V. A **state set** S consists of a set of state functions $s : L \rightarrow V$.

Transition Event

- A **transition event** (or simply **event**) on a state set S is an element $e = (s_i, s_j) \in (S \times S)$ of a binary relation on state set S that signifies the transition from one state to another. An event e is defined by a *condition* function $c(s_i)$ which evaluates a Boolean function in state s_i and by an *action* function p.

Scenario

- A **scenario** is a sequence of related transition events $\langle e_1, e_2, ..., e_n \rangle$ on state set S such that $e_k = (s_k, s_{k+1})$, for $1 \leq k \leq n$.

Service, Activity, Task, Procedure

- A **service**, **activity**, **task**, or **procedure** is a set of scenarios.

Society

- A **society** is a tuple (C, R), where
  1. $C = \{c_1, c_2, ..., c_n\}$ is a set of conceptual communities, each community referring to a set of individuals of the same class or type (e.g., actors, activities, components, hardware, software, data);
  2. $R = \{r_1, r_2, ..., r_m\}$ is a set of relationships, each relationship being a tuple $r_j = (e_j, i_j)$, where e_j is a Cartesian product $c_{k_1} \times c_{k_2} \times \cdots \times c_{k_{n_j}}$, $1 \leq k_1 < k_2 < \cdots < k_{n_j} \leq n$, which specifies the communities involved in the relationship and i_j is an activity (cf. Def. 8) that describes the interactions or communications among individuals.

Structural Metadata

- A **structural metadata** specification is a structure.

Descriptive Metadata Specification

- Let $\mathcal{L} = \bigcup D_k$ be a set of literals defined as the union of domains D_k of simple datatypes (e.g., strings, numbers, dates, etc.). Let also $\mathcal{R}$ and $\mathcal{P}$ represent sets of labels for resources and properties respectively. A **descriptive metadata specification** is a structure $(G, \mathcal{R} \cup \mathcal{L} \cup \mathcal{P}, \mathcal{F})$, where:

Descriptive Metadata Specification

1. $\mathcal{F} : (V \cup E) \rightarrow (\mathcal{R} \cup \mathcal{L} \cup \mathcal{P})$ can assign general labels $\mathcal{R} \cup \mathcal{P}$ and literals from $\mathcal{L}$ to nodes of the graph structure;
2. for each directed edge $e = (v_i, v_j)$ of G, $\mathcal{F}(v_i) \in \mathcal{R} \cup \mathcal{L}$, $\mathcal{F}(v_j) \in \mathcal{R} \cup \mathcal{L}$ and $\mathcal{F}(e) \in \mathcal{P}$;
3. $\mathcal{F}(v_k) \in \mathcal{L}$ if and only if node v_k has outdegree 0.

The triple $st = (\mathcal{F}(v_i), \mathcal{F}(e), \mathcal{F}(v_j))$ is called a **statement** (derived from the descriptive metadata specification), meaning that the resource labeled $\mathcal{F}(v_i)$ has property $\mathcal{F}(e)$ with value $\mathcal{F}(v_j)$ (which can be designated as another resource or literal).

Metadata Format

- Let $D_{\mathcal{L}_{MF}} = \{D_1, D_2, ..., D_i\}$ be the set of domains that make up a set of literals $\mathcal{L}_{MF} = \bigcup_{j=1}^{i} D_j$. As for metadata specifications, let $\mathcal{R}_{MF}$ and $\mathcal{P}_{MF}$ represent sets of labels for resources and properties, respectively. A **metadata format** for descriptive metadata specifications is a tuple $MF = (V_{MF}, \mathrm{def}_{MF})$ with $V_{MF} = \{\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_k\} \subset 2^{\mathcal{R}_{MF}}$ a family of subsets of the resources labels $\mathcal{R}_{MF}$ and $\mathrm{def}_{MF} : V_{MF} \times \mathcal{P}_{MF} \rightarrow V_{MF} \cup D_{\mathcal{L}_{MF}}$ is a property definition function.

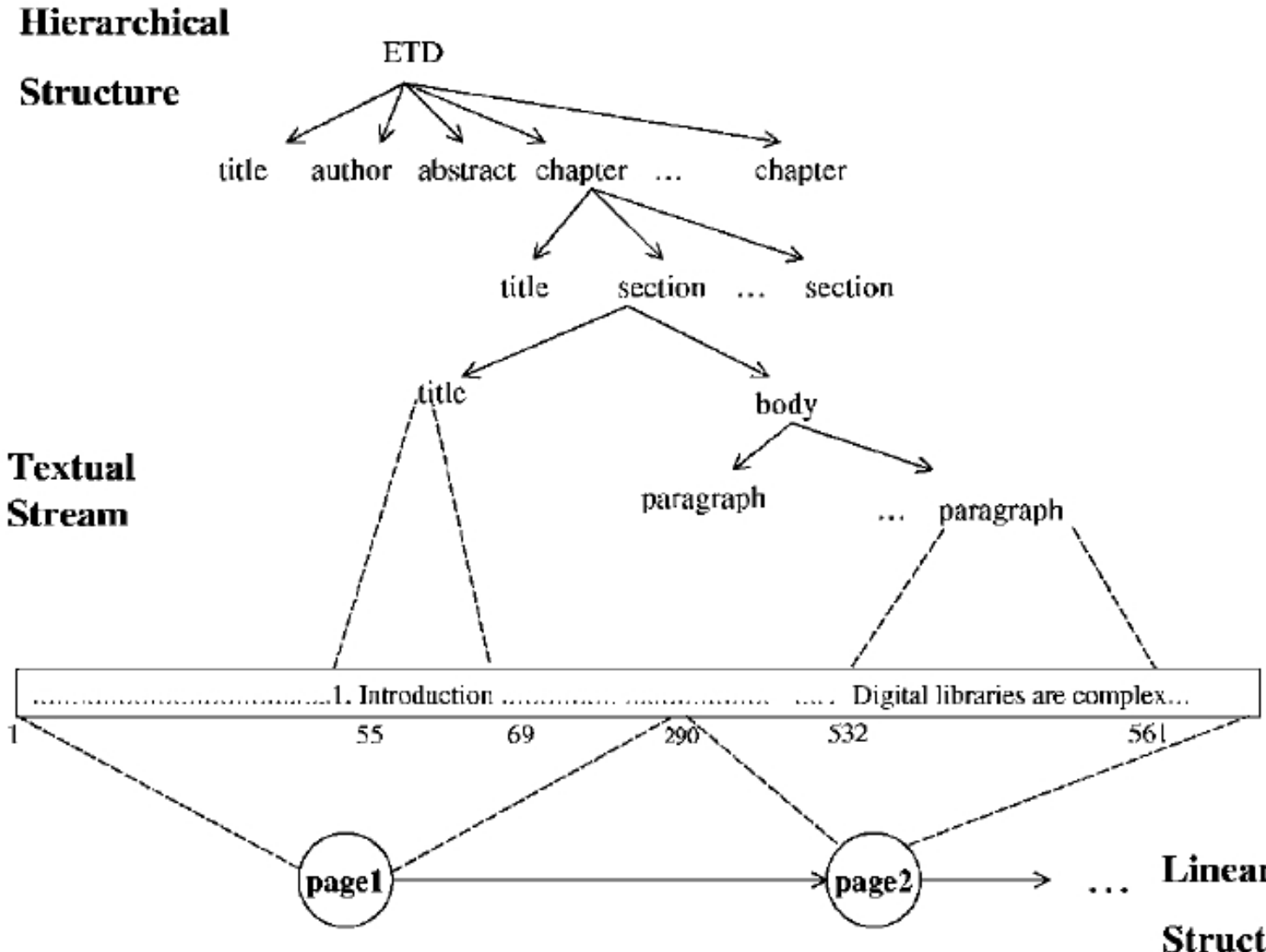
Metadata Format x Descriptive Metadata Spe

- A descriptive metadata specification $MS = (G_{MS}, \mathcal{R}_{MS} \cup \mathcal{L}_{MS} \cup \mathcal{P}_{MS}, \mathcal{F}_{MS})$ **conforms with** a metadata format $MF = (V_{MF}, \mathrm{def}_{MF})$ if $\mathcal{R}_{MS} \subseteq \mathcal{R}_{MF}$, $\mathcal{L}_{MS} \subseteq \mathcal{L}_{MF}$, $\mathcal{P}_{MS} \subseteq \mathcal{P}_{MF}$, and for every statement $st = (r, p, l)$ derived from MS, $r \in \mathcal{R}_k$ for some $\mathcal{R}_k \in V_{MF}$ and $p \in \mathcal{P}_{MS}$ implies $l \in \mathrm{def}_{MF}(\mathcal{R}_k, p)$.

Structured Stream

- Given a structure $(G, L, \mathcal{F})$, $G = (V, E)$ and a stream S, a **StructuredStream** is a function $V \rightarrow (\mathbb{N} \times \mathbb{N})$ that associates each node $v_k \in V$ with a pair of natural numbers (a, b), $a < b$, corresponding to a contiguous subsequence $[S_a, S_b]$ (segment) of the Stream S.

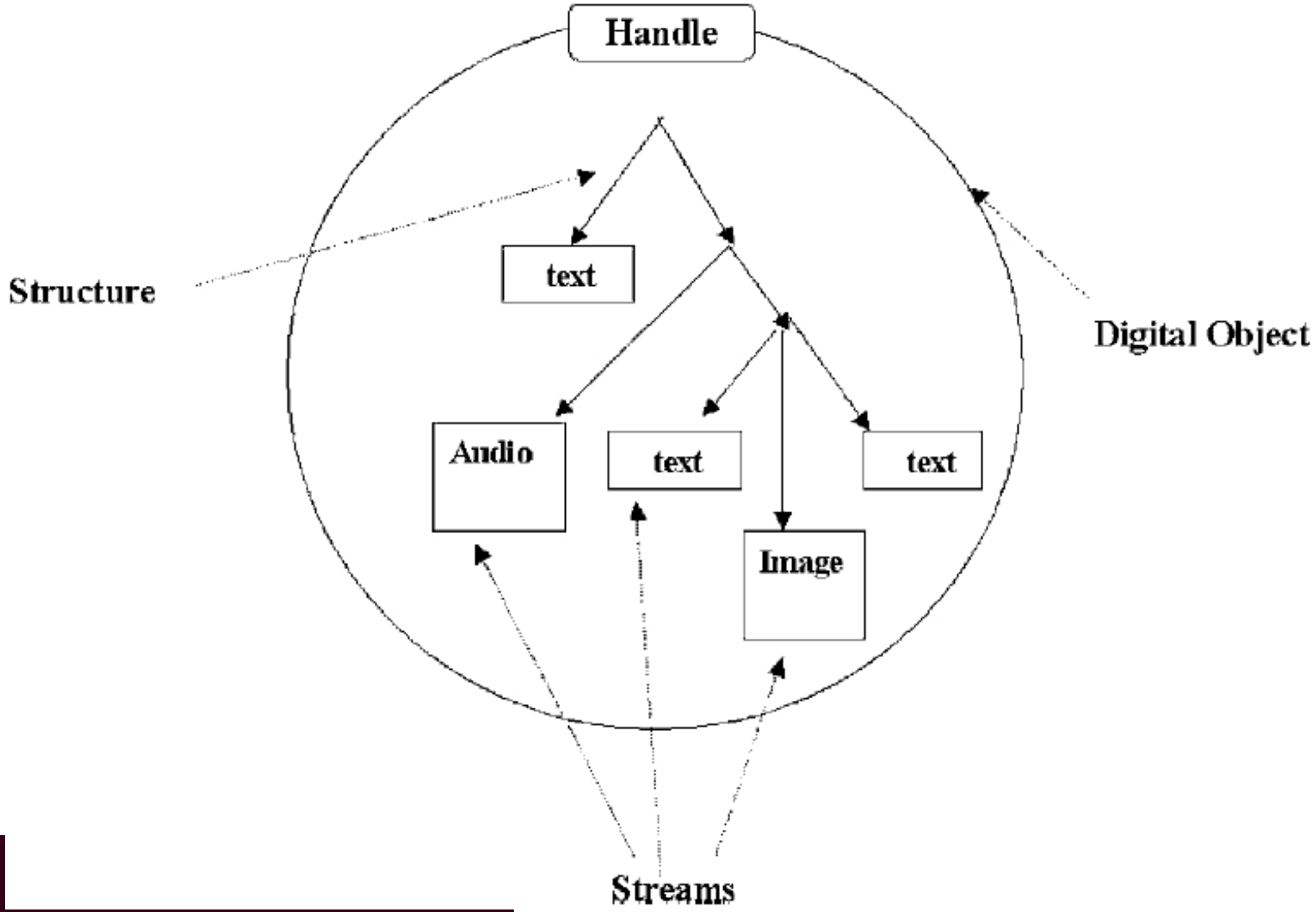
Structured Stream



Digital Object

- A **digital object** is a tuple $do = (h, SM, ST, StructuredStreams)$ where
  1. $h \in H$, where H is a set of universally unique handles (labels);
  2. $SM = \{sm_1, sm_2, \ldots, sm_n\}$ is a set of streams;
  3. $ST = \{st_1, st_2, \ldots, st_m\}$ is a set of structural metadata specifications;
  4. $StructuredStreams = \{stsm_1, stsm_2, \ldots, stsm_p\}$ is a set of StructuredStream functions defined from the streams in the SM set (the second component) of the digital object and from the structures in the ST set (the third component).

Digital Object



Collection

- A **collection** $C = \{do_1, do_2, \ldots, do_k\}$ is a set of digital objects.

Metadata Catalog

- Let C be a collection with k handles in H. A **metadata catalog** DM_C for C is a set of pairs $\{(h, \{dm_1, \ldots, dm_{k_h}\})\}$, where $h \in H$ and the dm_i are descriptive metadata specifications.

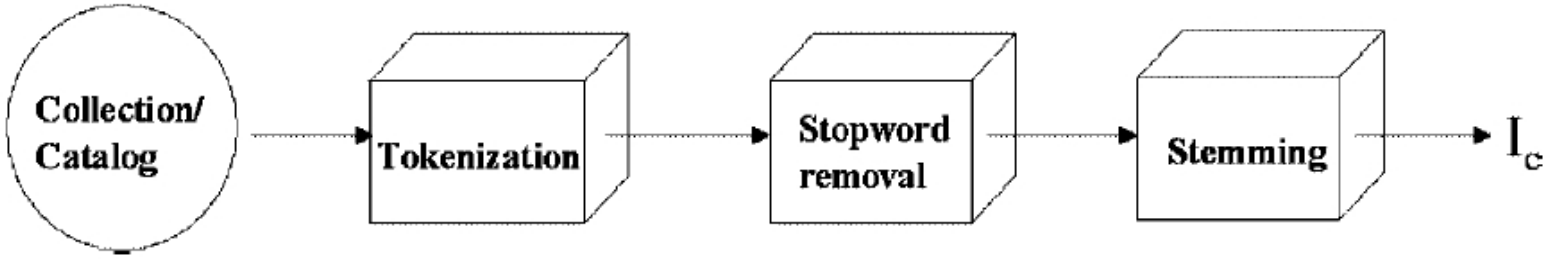
Repository

- Let C be a collection with handles H. A **repository** is a tuple $(R, get, store, del)$, where $R \subset 2^C$ is a family of collections and the functions “get”, “store,” and “del” satisfy:
  1. $get : H \rightarrow C$ maps a handle h to a digital object *get(h)*.
  2. $store : C \times R \rightarrow R$ maps $(do, \tilde{C})$ to the augmented collection $\{do\} \cup \tilde{C}$.
  3. $del : H \times R \rightarrow R$ maps $(h, \tilde{C})$ to the smaller collection $\tilde{C} - \{get(h)\}$.

Indexing Service

- Let $I : 2^{\mathcal{T}} \rightarrow 2^{H}$ be an index function where $\mathcal{T}$ is a set of indexing features and H is a set of handles. An **index** is a set of index functions. An **indexing service** is a single scenario $\{\langle is_1, is_2, ..., is_n \rangle\}$ comprised of pipelined scenarios $is_1, is_2, ..., is_n$ in which the starting state s_{k_0} of the first event of the initial scenario is_1 has a collection $s_{k_0}(K) = C$ and/or a metadata catalog $s_{k_0}(Y) = DM_C$ for collection C as its values and the final state s_{k_f} of the final scenario is_n has an index $I_C = s_{k_f}(Z)$ as its value (K, Y, and Z being labels of the respective states).

Indexing


Collection/
Catalog
Tokenization
Stopword
removal
Stemming
I_c

Searching Service

- Let Q be a set of conceptual representations for user information needs, collectively called *queries*. Let $M_{I_C} : Q \times (C \times DM_C) \rightarrow$ be a matching function, associated with an index I_C, that associates a real number with a query $q \in Q$ and a digital object $do \in C$ and possibly its descriptive metadata specifications $ms \in DM_C$, indicating how well the query representation matches with the digital object, structurally, by content, or regarding the descriptive metadata specifications.

Searching Service

- A **searching service** is a set of searching scenarios $\{sc_1, sc_2, \ldots, sc_t\}$, where for each query $q \in Q$ there is a searching scenario $sc_k = \langle e_0, \ldots, e_n \rangle$ such that e_0 is the start event triggered by a query q and event e_n is the final event of returning the matching function values $M_I(q, d)$ for all $d \in C$.

Hypertext

- Let $H = ((V_H, E_H), L_H, \mathcal{F}_H)$ be a structure and C be a collection. A **hypertext** $HT = (H, Contents, \mathcal{P})$ is a triple such that:
  1. $Contents \subseteq C \cup AllSubStreams \cup AllSubStructuredStreams$ is a set of contents that can include digital objects of a collection C, all of their streams (and substreams) and all possible *restrictions* of the StructuredStream functions of digital objects.
  2. $\mathcal{P} : V_H \rightarrow Contents$ is a function which associates a node of the hypertext with the node content.

Browsing Service

- A **browsing service** is a set of scenarios $\{sc_1, \ldots, sc_n\}$ over a hypertext (meaning that events are defined by edges of the hypertext graph (V_H, E_H)), such that traverse link events e_i are associated with a function $TraverseLink : V_H \times E_H \rightarrow Contents$, which given a node and a link retrieves the content of the target node, i.e., $TraverseLink(v_k, e_{k_i}) = \mathcal{P}(v_t)$ for $e_{k_i} = (v_k, v_t) \in E_H$.

Digital Library

- A **digital library** is a 4-tuple $(\mathcal{R}, DM, Serv, Soc)$, where
  - $\mathcal{R}$ is a repository;
  - $DM = \{DM_{C_1}, DM_{C_2}, ..., DM_{C_K}\}$ is a set of metadata catalogs for all collections $\{C_1, C_2, ..., C_K\}$ in the repository;
  - Serv is a set of services containing at least services for indexing, searching, and browsing;
  - Soc is a society.