Entity Identifiers for Lineage Preservation

Julien Gaugaz and Gianluca Demartini
L3S Research Center
Leibniz Universität Hannover
– Project context: OKKAM
– Setting: Revision of Entity Identity Decisions
– Problem Statement
– Example Scenario
– Prime Number Labeling Scheme for DAGs
– Lineage Preserving Entity ID
– Discussion & Future Work
– OKKAM Goal:
  • Foster the re-use of entity identifiers to ease information integration
  • To create and manage a large collection of entity identifiers (EID)
  • Not to create a complete knowledge base
    – Only discriminative information is stored
Operations during the Entity Lifecycle:

- Creation (ID Issuing)
- Split
- Merge

As a result of revising entity identity decisions
Goal: Resolve locally the lineage of entities

By providing Lineage Preserving EIDs

Need for supporting the following operations:

• Creation of EIDs
• Resolve whether a given EID A is an ancestor of an EID B
• Retrieve the list of all the ancestors of an EID
• Retrieve the list of all the descendants of an EID
Contribution

– EIDs that include its history
  • Content changes, history doesn’t!
  • Lineage Preserving EIDs allow to detect deprecated EIDs **locally**
  • No need for querying the OKKAM node
  • Definitive advantage in a fast evolving environment and as long-term solution
Outline

- Project context: OKKAM
- Setting: Revision of Entity Identity Decisions
- Problem Statement
- Example Scenario
- **Prime Number Labeling Scheme for DAGs**
- Lineage Preserving Entity ID
- Discussion & Future Work
Prime Numbers Labelling Scheme for DAGs

– DAG: G(V,E)

– Algorithm:
  • Assign a unique prime number $p$ to each $v$ in $V$
  • Label each $v$ with ($p$* the least common multiplier of its ancestors’ label) $\text{ancestor-label}$

Adapting Prime Number Labeling Scheme for Directed Acyclic Graphs
G Wu, K Zhang, C Liu, J Li - Database Systems for Advanced Applications, 2006 Springer
• The history of an entity can be represented as a DAG
• Prime number labeling can be used as a basis for creating the entity IDs

![Diagram](image)

- Deprecation
- Merge
- Split

• For each entity $e$ we assign $i=(e_{\text{self}}, e_{\text{ancestor}})$
Application of LPID

Sender

Local Entity Repository

E-mail

Receiver

Local Entity Repository

isAncestor(7^{120}, 3) = \text{rem}(120, 3)

OKKAM node
If the system knows A and receives C,
- It can replace A with C.

If the system knows A and receives C,
- It only knows that A is deprecated.
Being able to identify locally deprecated EIDs allows to:

- Reduce the number of requests to OKKAM nodes
- Assure that an entity is represented by only one EID in the local repository

EIDs based on prime numbers are not intuitive for end-users
Using DNS for resolving EIDs

- We can encode LPID in a DNS node for associating user-friendly name to entities
- The max number of unique ancestors we can encode is, at least, 483 (estimation)

[RFC1035 (DNS standard)]
Future Work

– In-depth analysis of space/time requirements
– Comparison with other approaches
– Simulation for studying the growth of LPID size with the number of entities and operations considered
– Simulation for studying the space limitation while using DNS
Thanks