

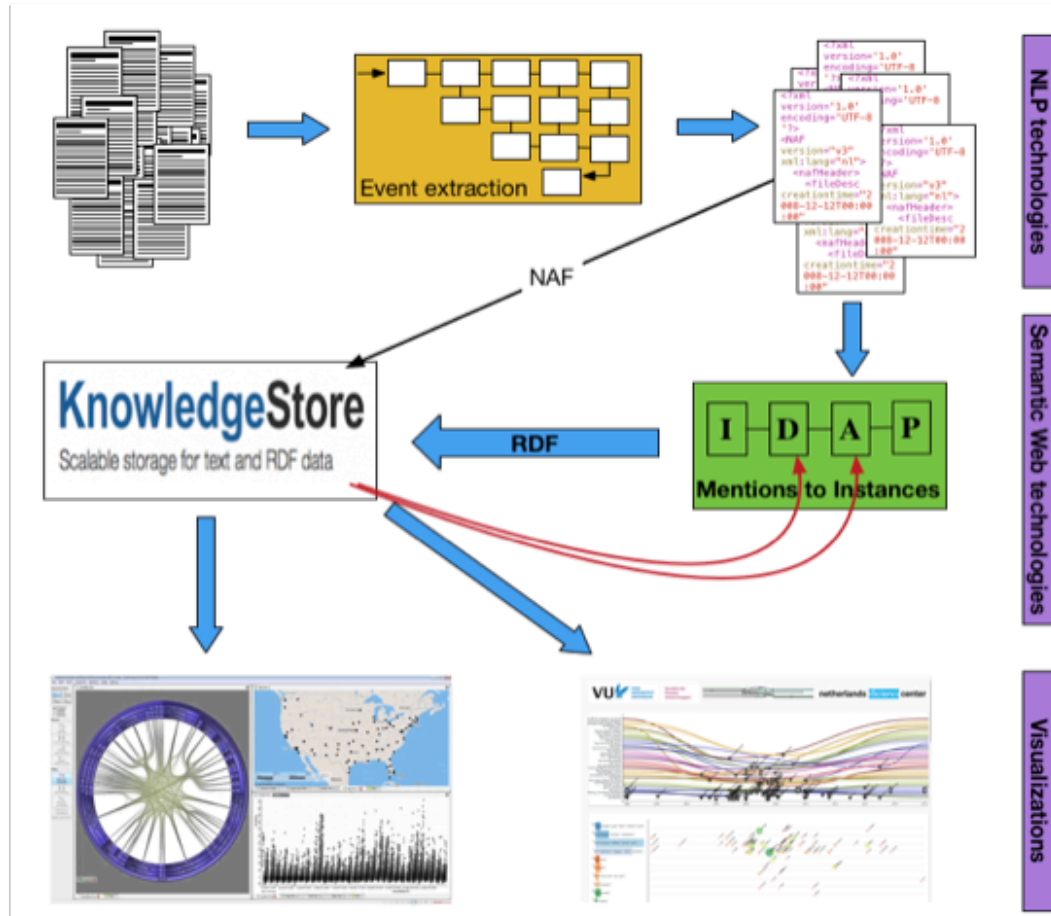


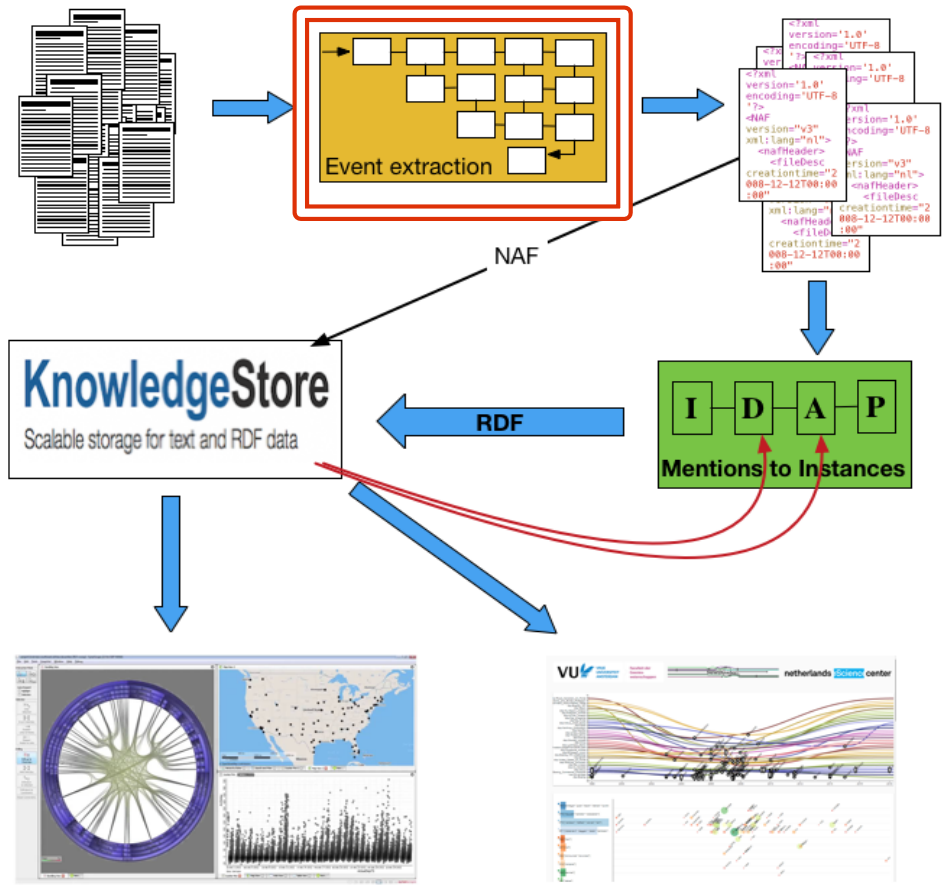
The Event and Implied Situation Ontology (ESO)

ICT 316404, FP-7-ICT-2011-8

www.newsreader-project.eu



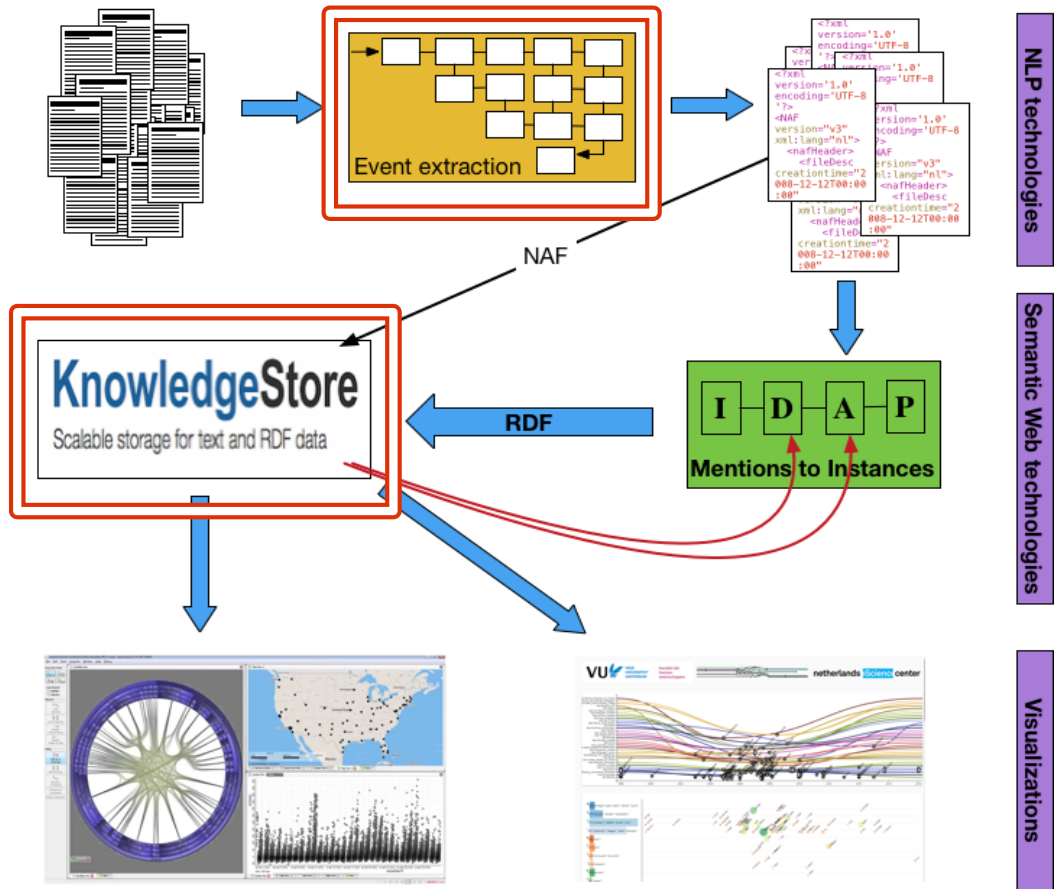




NLP technologies

Semantic Web technologies

Visualizations



NLP technologies

Semantic Web technologies

Visualizations

ESO in short!

- The **E**vent and Implied **S**ituation **O**ntology (ESO)
- Models the implications before, after and during an event and the roles of the entities involved in the event.
- Manually built event ontology with manual mappings to SUMO, FrameNet frames and Frame elements
- Manual mappings to WordNet synsets
- Written in OWL
- Freely available (CC BY SA license)

Application of ESO

- Typing of events, e.g. eso:Transportation
- Models and defines a) the implications of events and b) the roles of the participants affected by the event.
- Runs on Semantic Role Labeled (SRL) text
- Relies on Semantic Web techniques

The problem we had

- Millions of events in all kinds of lexicalizations, and with a variety of e.g. Frames and semantic roles.
- We know something has happened, but not what the *implications* or the pre and post situations of an event are.

Event implications?

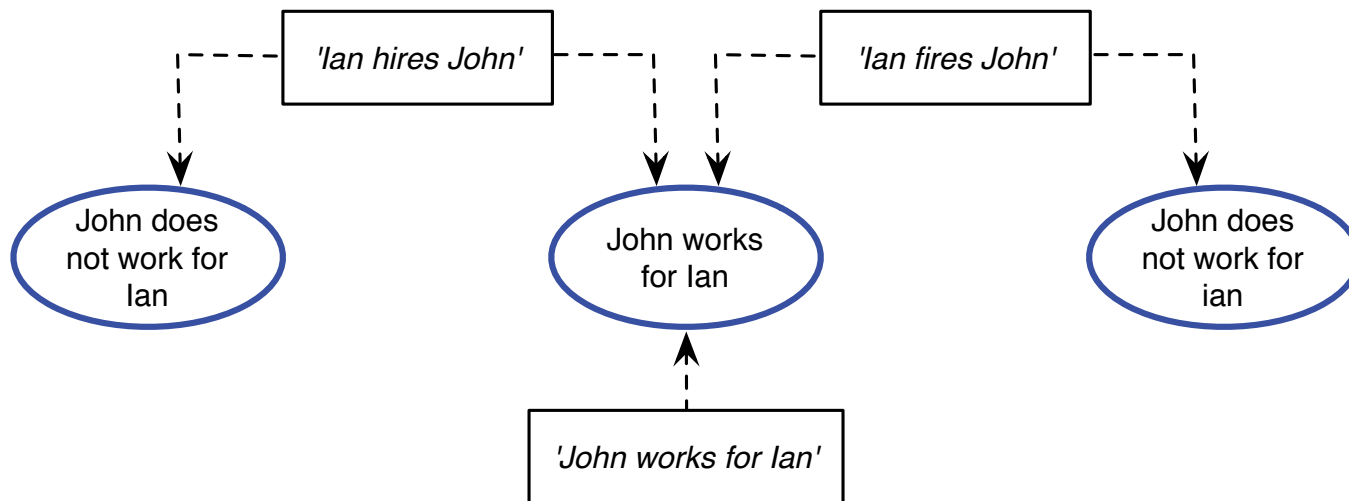
- “Apple hired Steve as their new CEO to save the company.”

Property change

- “Apple hired Steve as their new CEO to save the company.”
 - Before: Steve *notEmployedAt* Apple
 - After: Steve *employedAt* Apple
 - Steve *hasFunction* CEO
 - Steve *hasTask* save the company
 - Steve *isEmployed* true

Implications of Static and Dynamic Events

- Derive sequences of states and changes over time, regardless if the information is explicitly mentioned in text, or inferred by a reasoner:



Simple Semantics!

You (X) might be fired, quit your job, leave, resign or retire, but in the end, you no longer work for some employer.

eso: LeavingAnOrganization (skos:closematch fn: Quitting, fn: Firing)

X notEmployedAt Y

You might be a fn:Donor, a fn:Victim or a fn:Seller, but first you owned something and now you don't.

eso: ChangeOfPossession

X notHasInPossession Y

You may travel to Bucharest while smiling and with a certain speed, but in the end, you're in Bucharest and not where you were before.

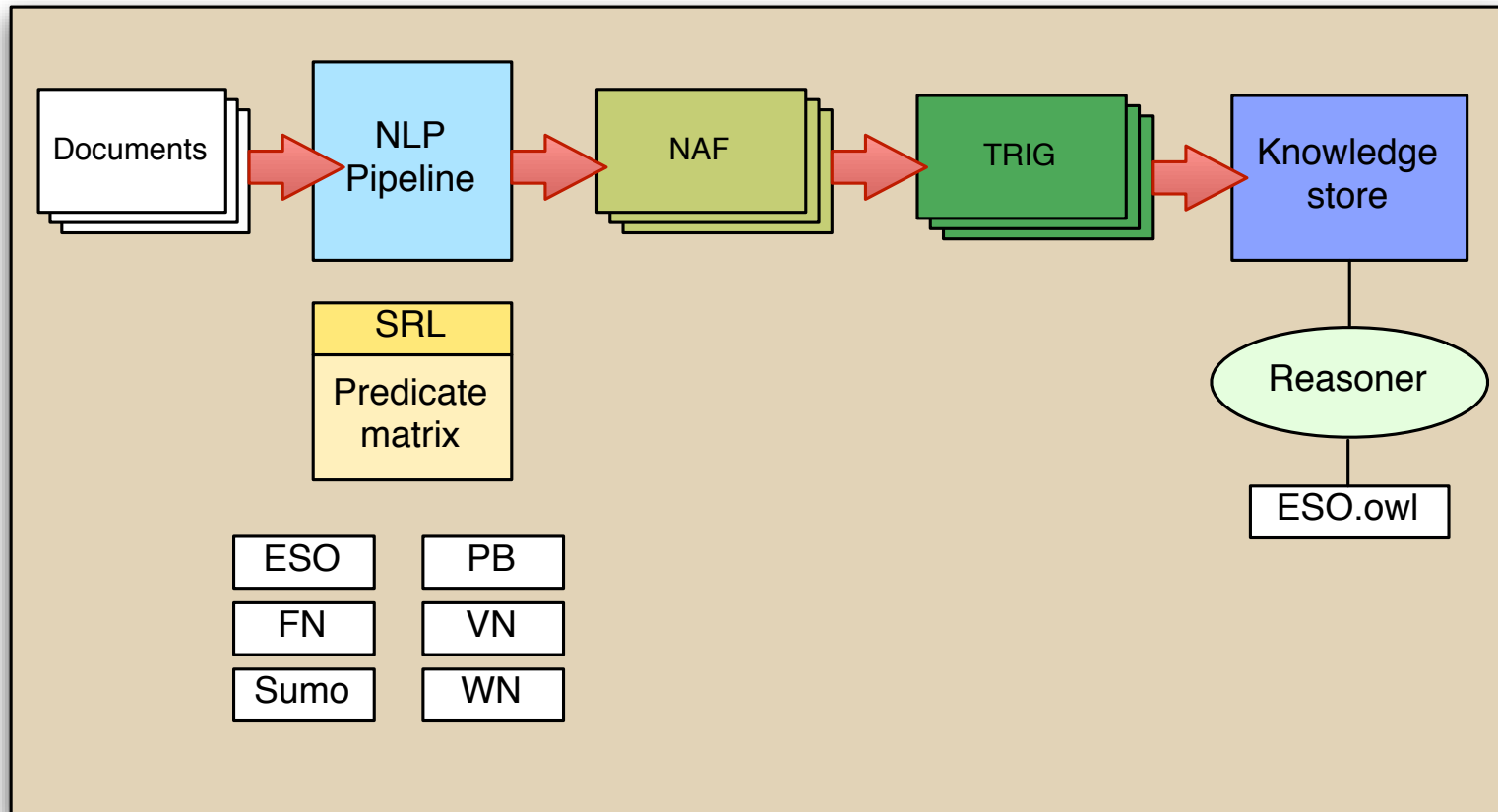
eso: Translocation

X atPlace Y

ESO approach to event modeling

- In ESO, we focus on modeling the implications of events, not on the semantics of the events themselves.
- And we don't define all possible implications of an event.
(Though the model is open to extensions)

How do we do it: ESO in our NLP and Knowledge Suite



Text to RDF

“John generously gave the book to Ian.”

John (**fn:donor**/**eso:possession-owner_1**) generously (**fn:manner**) gave (**fn:Giving**/**eso:ChangeOfPossession**) the book (**fn:theme**/**eso:possession-theme**) to Ian (**fn:Recipient**/**nwr:possession-owner_2**)

obj-graph-eventX

:eventX

a eso:ChangeOfPossession;

eso:ChangeOfPossession_possession-owner_1 :John;

eso:ChangeOfPossession_possession-owner_2 :Ian;

eso:ChangeOfPossession_possession-theme :book;

sem:hasTime :time_eventX.

Instantiating the pre and post situations

Situation rules:

eso:pre_ChangeOfPossession

eso:hasSituationRuleAssertion pre_ChangeOfPossessionAssertion1;

eso:hasSituationRuleAssertion pre_ChangeOfPossessionAssertion2.

Assertions:

eso:pre_ChangeOfPossessionAssertion1

eso:hasSituationAssertionSubject eso:possession-owner_1;

eso:hasSituationAssertionProperty eso:hasInPossession;

eso:hasSituationAssertionObject eso:possession-theme.

eso:pre_ChangeOfPossessionAssertion2

eso:hasSituationAssertionSubject eso:possession-owner_2;

eso:hasSituationAssertionProperty eso:notHasInPossession;

eso:hasSituationAssertionObject eso:possession-theme.

Instantiation:

:eventX_pre (*John gave the book to Ian*)

:instanceX(John) eso:hasInPossession :instanceZ(book)

:instanceY(Ian) eso:notHasPossession :instanceZ(book)

Contents of ESO

| Component | Number |
|-----------------------------------|--------|
| Event classes | 63 |
| Dynamic event classes | 50 |
| Static event classes | 13 |
| SUMO class mappings | 46 |
| FrameNet Frame mappings | 103 |
| Situation rule assertions | 123 |
| Properties | 58 |
| ESO roles | 65 |
| Mappings of roles to FrameNet FEs | 131 |

-More on ESO in Deliverable 5.5.2:

<http://www.newsreader-project.eu/publications/deliverables/>

-ESO.owl, an extensive documentation, publications on ESO and the manual FrameNet LU to PWN 3.0 mappings:

<https://github.com/newsreader/eso>

Contact:

r.h.segers@vu.nl

rosPOCHER@fbk.eu

