ALPE as LT4eL processing chain environment

Dan Cristea, Ionut Cristian Pistol, Corina Forăscu
(dcristea,ipistol,corinfor@info.uaic.ro)
"Alexandru Ioan Cuza" University of Iasi, Romania
Faculty of Computer Science

Outline

- LP meta-systems
- LP augmented hierarchy : ALPE
- The LT4eL processing chain
- Questions, comments...

NLP meta-systems

- Make use of existing modules in building LP chains
- Make use of existing linguistic resources and allow the user to add/build new ones
- Allow the user to compare and choose between modules available
- GATE (http://gate.ac.uk/), (1998) CUE (http://www.dcs.shef.ac.uk/~hamish/dalr/granada/mason.final.html), ATLAS (http://www.nist.gov/speech/atlas/), UIMA (http://www.research.ibm.com/UIMA/), LiveTree (http://www.fxpal.com)

Linguistic Processing Chains

- Linguistic Processing (Chains) Flows – used by UIMA and ALPE
- Def: a series of sequential and/or parallel applications of processing modules on an initial input that produces a output. During the execution of the processing flow additional resources might be accessed (language models, lexicons, etc.).
- A processing flow can be executed locally, on a single computer or on a distributed network (GRID) or can be run remotely as applications of webservices and scripts.

What do they offer?

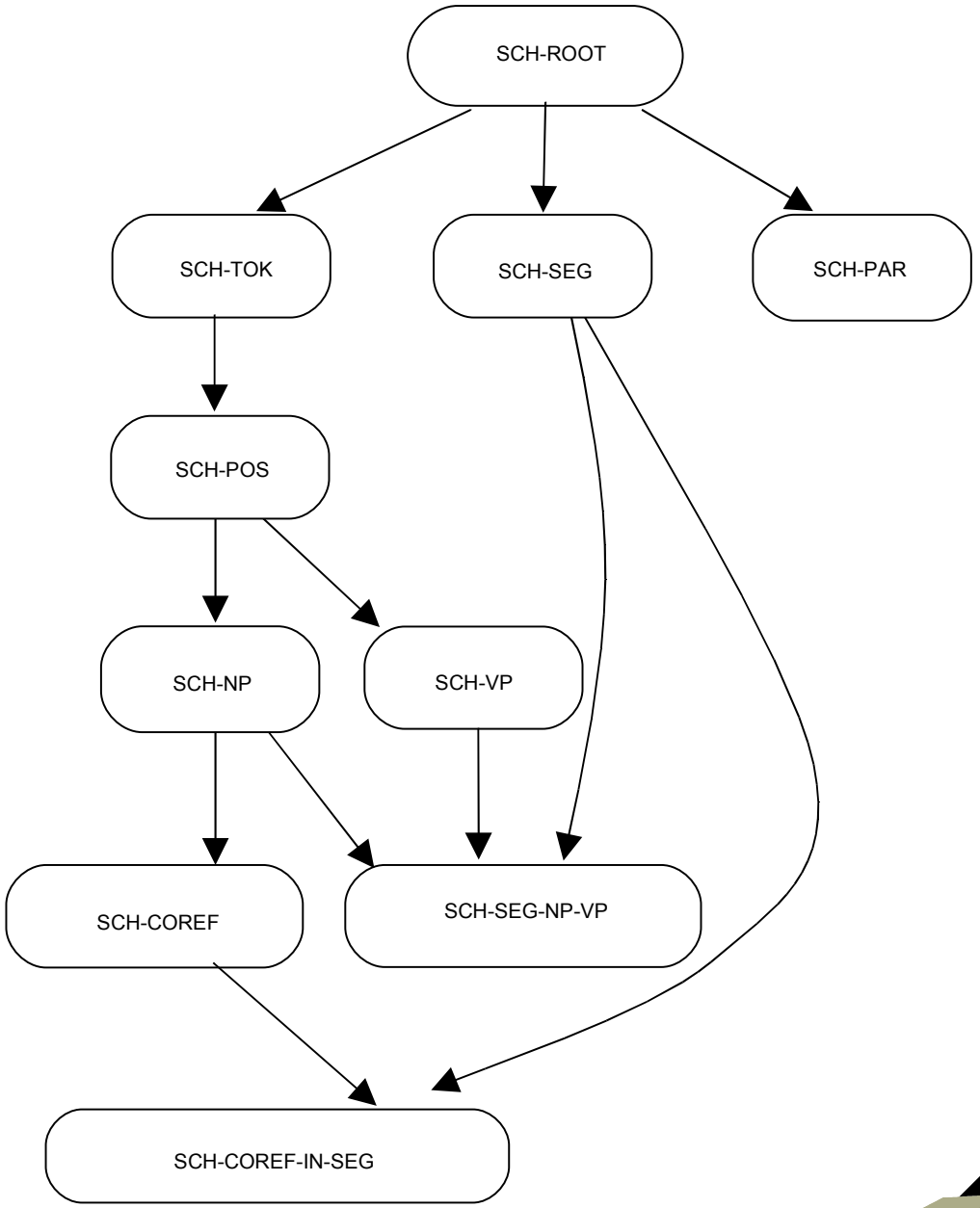
- Access to a consistent list of LP modules
- The user is able to add new modules
- The user is able to configure processing flows
- The user is able to run processing flows on any number of documents
- The user is able to edit and/or configure output format (UIMA)

What they don't offer?

- The user has to determine the format and language of a document in order to select the appropriate modules and resources
- The user can't include new formats in a processing flow
- The user has to make a documented decision wrt what modules to include in a flow
- The user has to manually build the flow
- The time and effort saved is minimal: the user still has to read the module documentations and the meta-systems own documentation (GATE 292 pages, UIMA 364 pages just for the User's Guide).

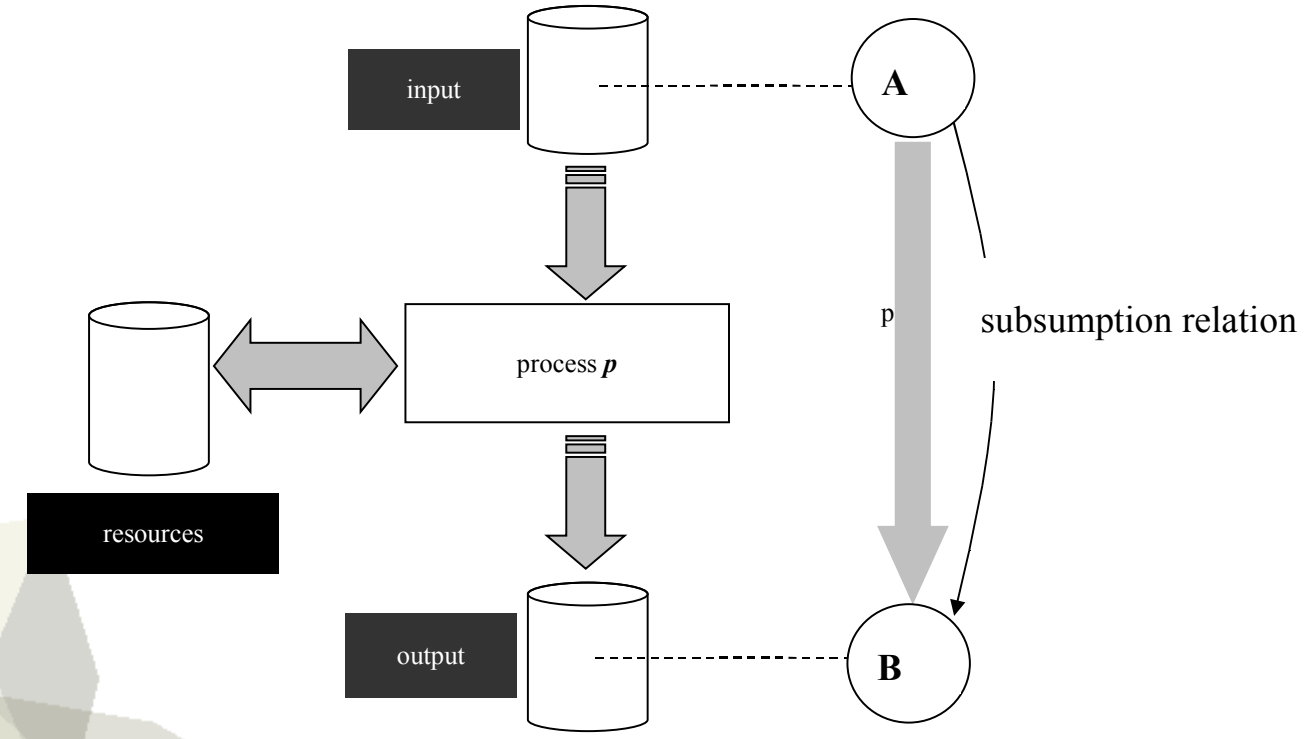
Hierarchy of annotations schemas

- Dan Cristea and Cristina Butnariu. 2004. Hierarchical XML representation for heavily annotated corpora. In *Proceedings of the LREC 2004 Workshop on XML-Based Richly Annotated Corpora*, Lisbon, Portugal.
- Graph representation of XML annotation schemas
- Directed acyclic graph:
  - nodes: annotation schemas
  - edges: hierarchical relation (subsuming)
  - the root: the void annotation



The augmented hierarchy

- If node A subsumes node B we can attach a processing module to the edge between B and A. A module can be attached to the edge between B and A if running that module on a file conforming to the annotation schema corresponding to B obtains an output conforming to A.



ALPE hierarchies

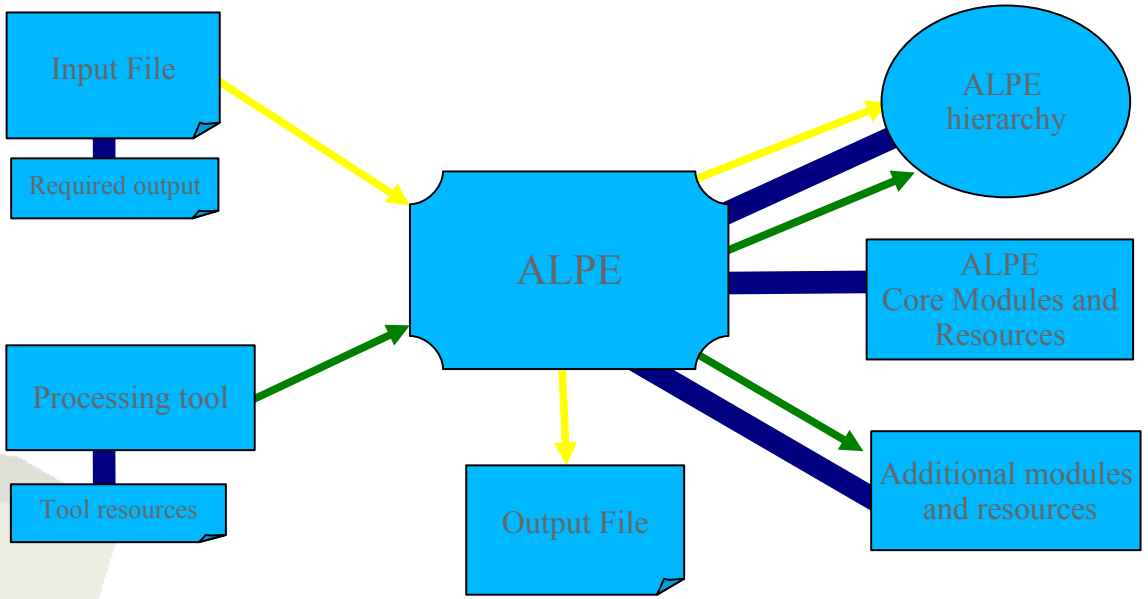
- On an edge can be attached multiple processing modules
- The edges that have no explicit module attached are considered to have attached a void module
- Each processing module can also have attached to it various processing resources. These processing resources form a further extension of the augmented hierarchy, called the REX ALH (Resource Expanded ALPE Hierarchy). The nodes in the REX hierarchy are resources and the edges are constraints wrt their usage (language, speed, performance).
- Three hierarchy building operations: **initialize-hierarchy**, **classify-document** and **integrate-process**.

ALPE processing flows

- On the augmented graph we define the following operations:
  - *Pipeline (m1,m2,...,mk)*: adds the modules m1,...,mk to a processing pipeline
  - *Simplify (sd1,sd2)*: simplifies document from the standard sd1 to the subsumed standard sd2
- We define a **direct processing flow** as:
  - FØ is a direct processing flow that when applied leaves the input unchanged
  - Any combination of applications of *pipeline* and *simplify* on any number of modules and FØ is a direct processing flow
- Another operation:
  - *Merge (flow1, flow2, ..., flowk)* : produces a flow that merges the output of flow1, flow2,..., flowk
- The processing flows that include merge operations are called **merging processing flows**

ALPE (Automated Linguistic Processing Environment)

- Implements the augmented hierarchy model
- Allows the development of ALH, the computation and execution of flows
- Main functionality: the user provides an input document (corpora) and selects a desired output format from those available in the hierarchy, ALPE computes available flows and executes one (or all) on the input document, producing a document conforming to the required output format



ALPE Core modules

- Language identification module
- Format identification and classification for documents
- Simplification of a document to a node (standard) in the hierarchy
- Merging of annotated version of the same base text
- Creation and development of ALH
- Integration of a new processing module in a existing ALH
- Possibly others...

Integration of a new LP tool

•This module allows new tools to be added to an existing hierarchy using the previously described algorithm.

•Any linguistic processing tool can be added to any existing hierarchy, but some additional information has to be provided to this module:

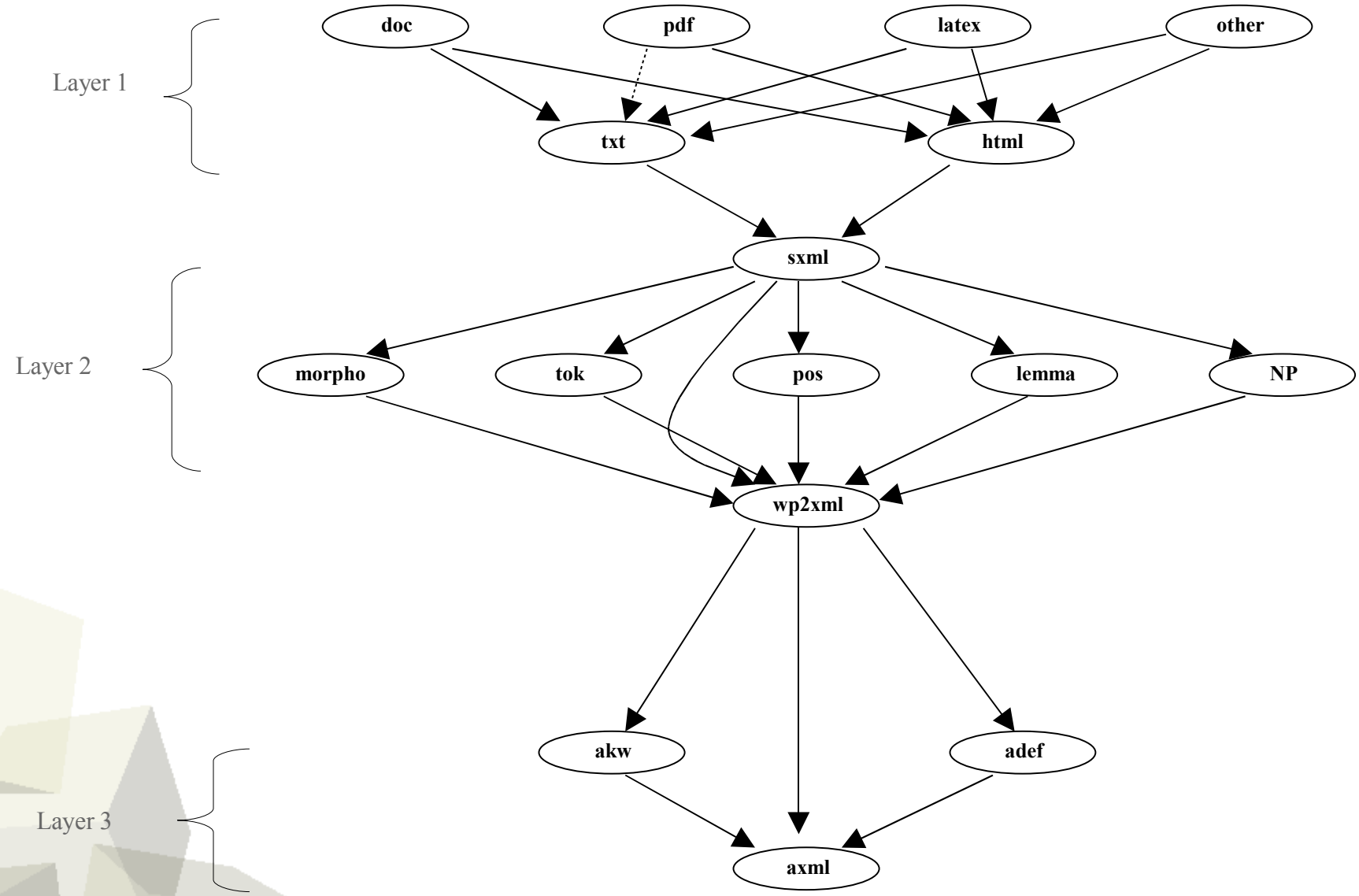
- the software requirement for running the tool (if the actual tool is provided)
- the actual executable tool, or the address and configuration information, if the tool is a webservice
- the run parameters (input/output files, resources required, etc.)
- additional resources required (language models, lexicons, etc.)

•After the tool is integrated it will be automatically used in future processing chains, if required.

Flow computation and execution

- If a module requires some language resources which are not available, that module will be invisible in the flow computation
- **Alternative flows :** If more than one flow is available for the same task, the user can select one or more to be executed:
  - The user can select between flows provided with the following parameters for each flow: number of modules involved (length of flow), number of intermediate formats (weight of flow), estimated time of execution, accuracy of result, financial cost (all based on module specification)
  - ALPE allow the user to specify a default option, which will be chosen automatically
- **Execution time :** Some flows can have an execution time of several hours or more.
- What can be done:
  - Distributed computing (GRID network using several computers and a central hub). Already used with very good results (execution time dropped from 8 hours to 1,5 minutes using a 5 computer GRID).
  - Partial execution and the ability to resume paused/stopped flows.

The LT4eL annotation formats hierarchy



Use case scenarios

- User uploads a new document (in text format) to the LT4eL server and marks the new document as a LO. The server detects the language of the document, automatically produces the significant formats (sxml, wp2xml, axml) and notifies the user when completed.
- User uploads a new tool that performs tokenisation. The tool is automatically added in the hierarchy and made available as a possible module in processing flows. The new tool can be used in alternative processing flows, so the user can compare the results of two or more modules performing the same task.
- The user requires just a tokenised version of a LO. He can use any version of that LO as input and select the tokenised format from the hierarchy as output.

LT4eL processing chain requirements

- Able to support multilinguality (language resources)
- Able to support modules running on different platforms (perl, java, C++, (?) webservices)
- Linear processing chain, with a non-annotated input (although different formats: txt, html, doc, (?)pdf) and a single desired output format
- Real – time (probably) unrealistic:
  - Server processing a queue of request
  - Server running a GRID network
  - Stand-alone off-line free software made available : input txt (or any format), output wp2

What did we find out? (10-12 july, Tuebingen)

- Q: For what languages are the necessary modules (freely) available? A: at least rom, eng, cze
- Q: What are the languages for the initial development? A: rom, eng, cze
- Q: What system should be used to create/execute the processing chains? A: ALPE
- Q: Is real-time on-line processing possible? A: probably not – this will require significant hardware upgrades
- Q: What is the deadline? A: for the initial development : february 2008, for the other languages : if the hierarchy is already integrated and working, adding new tools is easy, so march-april 2008 (providing that the tools are available)

Thank you!
Questions? Suggestions? Offers to help?