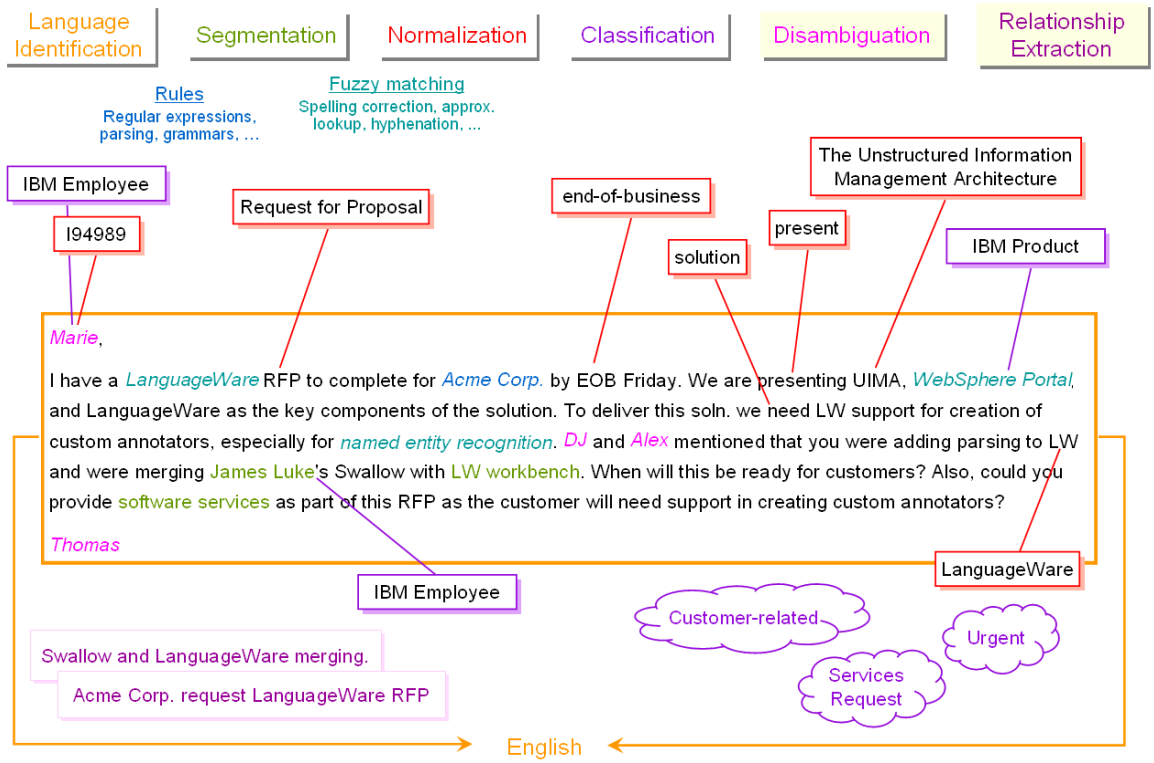
Introduction

In 2001, IBM launched an innovation project out of the Dublin (Ireland) Software Lab, to create IBM's next generation natural language processing (NLP) technology incorporating hundreds of person-years of existing NLP research across IBM. This project was a response to the increasing demand for solutions to help in how we organize, store, find, relate and ultimately action information – information which is frequently residing within ever growing volumes of unstructured text. Due to the requirements for NLP processing capabilities across the entire IBM software stack – the need for ubiquitous linguistic understanding – the technology needed to be a robust, flexible, and customizable component technology which:

- Could be integrated into any software application.
- Could be deployed in simple-to-use integration packages that satisfy any type of application development environment - from desktop to mainframe, through Java or C APIs, UIMA, Eclipse, SOA, across many platforms and languages.
- Would combine strong engineering principles with latest research techniques, leveraging ideas, algorithms and research assets from the various IBM Research labs and external research organizations.
- Would be open, extensible, scalable, customizable, and extremely fast (processing millions of words per second).

As a component technology, LanguageWare can support the broadest possible range of applications with a diverse set of linguistic requirements.



Today, LanguageWare is embedded in Lotus, WebSphere, DB2, Tivoli and Rational products, is deployed as part of GBS engagements, is used within IBM Research solutions, and is being licensed directly to IBM customers and partners.

Technology

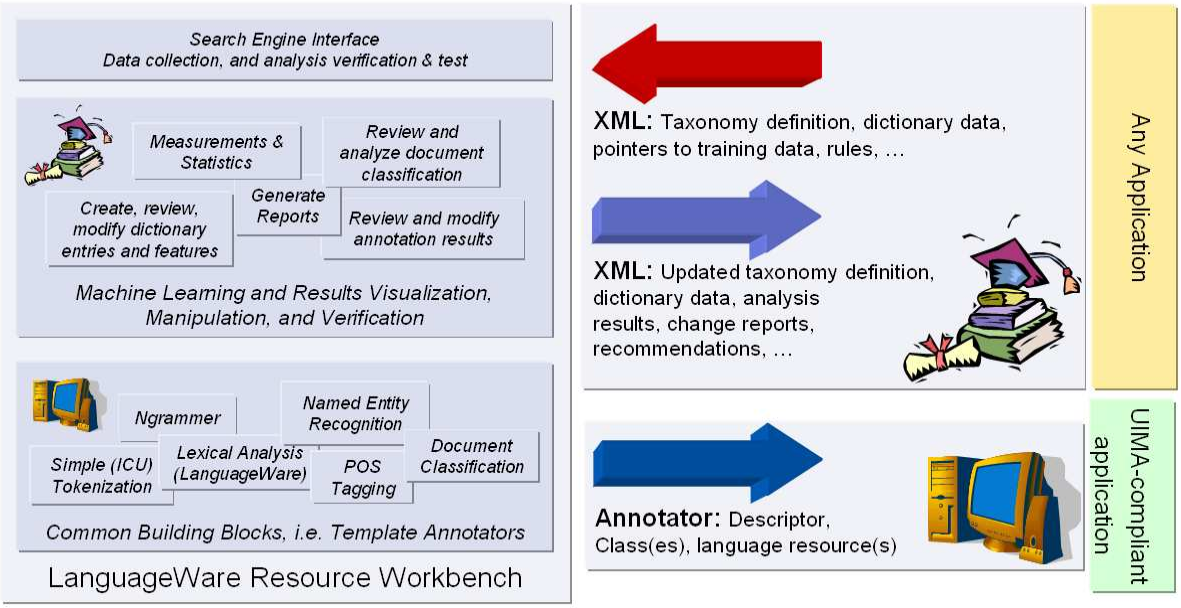
The efficiency and diversity of processing stems from LanguageWare's cross-linguistic architecture. Through identification of key phenomena that might span several languages and utilization of appropriate formal models (such as state machines, formal rule systems, logic and statistical tools) LanguageWare removes the overhead of traditional layered language engineering approaches.

Lexical data is represented in finite state devices which use a patented polymorphic node format. This enables LanguageWare to use a uniform and compact representation of lexical resources for all languages. This is achieved through investigation of graph metrics and statistical properties of morphological state transition networks from the point of view of random networks theory.

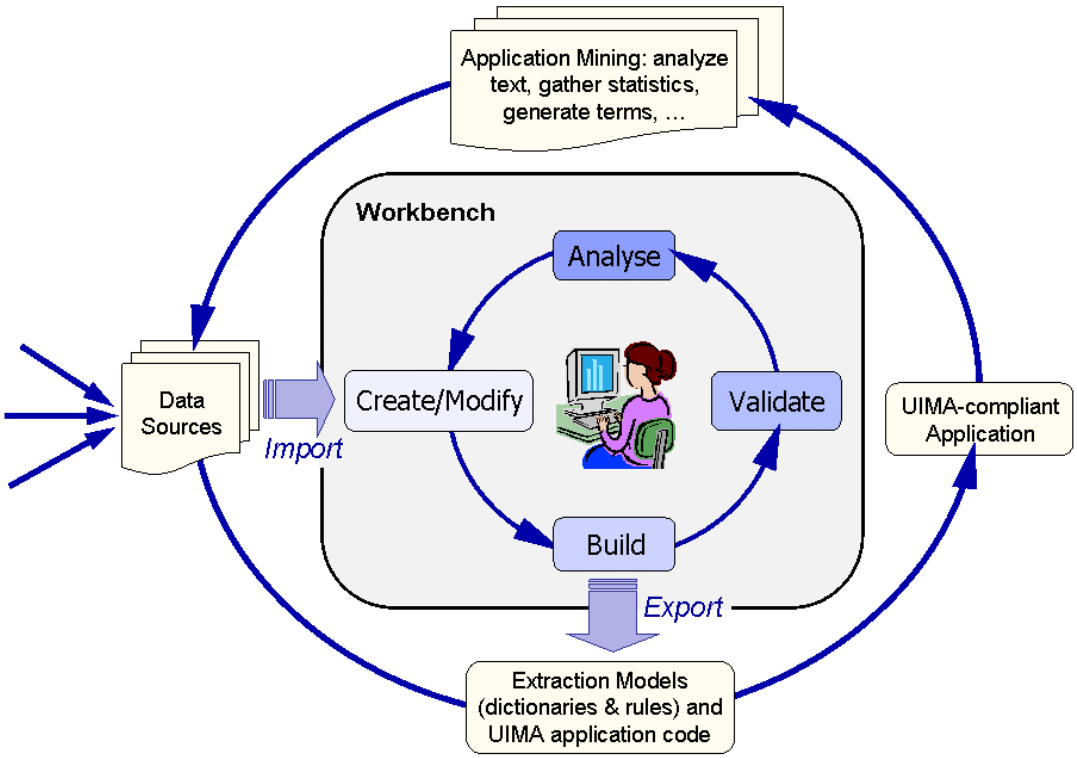
As a result of this cross-linguistic architecture, IBM LanguageWare facilitates the creation of multi-lingual text analytics by transparently handling the challenges associated with many different languages. For example:

- In Arabic the semantic component of words may be masked by prefixes and postfixes.
- For highly inflected languages, like Russian, lemmatization is particularly important to achieve high recall in information retrieval.
- For non-white spaced languages, such as Chinese or Japanese, text segmentation into words is crucial.
- For compounding languages, such as German, segmentation of words into their constituents is particularly important for information retrieval.
- For English part-of-speech ambiguity and inflectional irregularities need to be resolved.

One key design characteristic is that LanguageWare uses data-driven approach, where the behavior of the system can be significantly modified through the customization of the LanguageWare resources. This essentially allows the customer to inject any type of domain-specific knowledge into the LanguageWare analysis process. The information that may be stored in these resources and leveraged during the analysis process is not limited to the traditional lexical type of data. These resources allow you to encode/import morphology, morphotactics, synonyms, ontologies, taxonomies, relationships, constraints, parsing rules, etc.



LanguageWare also provides a comprehensive development environment – The LanguageWare Resource Workbench – which is an Eclipse-based application that simplifies this domain customization process through providing tooling that enables LanguageWare resources to be compiled from any domain data. The benefit of the Workbench is that it vastly simplifies the process of creating, updating and managing language resources and building them into your analysis process.



The Workbench also allows, through a simple drag-and-drop examples-based interface, the creation of rules and grammars to be applied during the analysis process. The Workbench is designed as to allow complete customization of our analysis process without the customer having to write one line of code. It aims to provide an entire development environment in which advanced analytics can be easily developed by domain specialists.

Today LanguageWare supports a wide range of linguistic functions, all of which are provided as component capabilities that can be easily integrated into any application environment.

Below is a quick list to the main features:

- **Dictionary Lookup:** Lookup functions, such as synonym expansion, hyphenation, and approximate string matching.
- **Language Identification:** Recognizes the language of a piece of text.
- **Lexical Analysis:** Main process through which the language analysis is performed.
- **Morphological Analysis:** Lemmatization, generate inflected form (GIF), inflect from model, etc.
- **Morphological Guesser:** Handles (guesses morphology) for words not in the dictionary.
- **Multiword Units:** Handling of multi-word terms – inflections, incompletes, ordering, etc.
- **Parsing:** Build rules, regular expressions, or shallow grammars for identifying entities in text.
- **Part of Speech (POS) Tagging:** Identifying the POS of words or phrases, with disambiguation.

- **Semantic Analysis & Disambiguation:** Allows for concepts (as opposed to terms) to be spotted in texts and disambiguated with respect to other concepts present, leveraging semantic graphs that connect concepts.
- **Text Correction:** Error checking and spelling suggestion.
- **Text Segmentation:** Tokenization & sentence and paragraph detection.
- **Tooling:** Eclipse-based tooling, LanguageWare Resource Workbench, for customizing LanguageWare – build dictionaries, rules, UIMA pipelines, compare analysis results, get statistics, performance benchmarking, etc.

Services

LanguageWare delivers Domain Customization Service that helps organizations integrate their own domain knowledge into the overall analysis process. The LanguageWare team comprises specialists in linguistics, computational linguistics, computer science, mathematics, terminology, and translation, at PhD, Masters, and Bachelors degree levels, with specialists in over 20 languages.

LanguageWare consultants can analyze our customers existing domain data – vocabularies, ontologies, terminologies – and recommend how these can be exploited during the text analysis process. They can provide tools to help extract new domain data from existing information repositories, linguistically enhance it (even with high quality domain vocabularies, they are frequently under-specified in terms of describing the linguistic characteristics that effect how the terms may be constructed in real text), and then compile it into the LanguageWare resources. They can advise on the terminology standards – both in construction and ongoing management of terminologies. They can develop any additional code required to allow the customer to exploit these new analysis capabilities into their solutions. They can provide ongoing support of the resulting domain-specific application layer, or alternatively provide the customer with the tooling so that they can update the application with the new domain vocabularies as/when they evolve within their organization. With LanguageWare you not only get the best NLP technology, but also a development and services team dedicated to ensuring that your organization gets the greatest return on investment from LanguageWare and your own domain data.