Motivation

• Most available annotation tools are designed for one or few annotation types/levels
• Some widely-used tools are not actively developed anymore
• Some research questions require data annotated on more than one or a restricted set of levels (e.g., scope relations, typological questions, information structure)

→ Requirements for multi-level corpus annotation tools:

1) **Extensibility:** A pluggable architecture extensible with new tooling for new annotation types
2) **Genericity:** A data model able to accommodate potentially all types of annotations
3) **Accessibility** for users with different levels of experience/from different fields of research
4) **Compatibility:** A high level of compatibility with other tools and formats

• Atomic’s architecture has been designed with the aim to fulfil these requirements

Genericty

• Atomic uses Salt (Zipser and Romary, 2010) as its data model, a graph-based metamodel for linguistic data
• In Salt, potentially all kinds of annotations can be modeled, which enhances Atomic’s extensibility even further
• Salt is low in semantics, hence independent of specific linguistic analyses, tagsets, annotation schemes, theories, media types

Accessibility

• Atomic provides graphical editors to create and manipulate corpora, e.g., an annotation graph editor for Salt graphs, and a coreference editor as an example for a special purpose editor
• Atomic provides an expert mode – a command-line console for the native annotation language AtomicAL, which allows for rapid annotation

Extensibility

• Developed on top of the Eclipse Rich Client Platform (RCP, McAffer et al., 2010), a platform-independent Java application platform with a mature plugin framework
• Atomic from an architectural perspective is a set of plugins added to the RCP’s own plugins
• Further editors and tooling can be implemented as independent plugins
• Atomic profits from the large number of plugins already available in the Eclipse ecosystem, e.g., plugins for version control or real-time collaboration can easily be added

Compatibility

• Atomic includes Pepper (Zipser et al., 2011), a conversion framework for linguistic formats
• Pepper provides Atomic with compatibility to formats such as EXMARAlda, TigerXML, tiger2, PAULA, MMAX2, TCF, the ANNIS format, and many more, from which and into which Atomic can import and export

References