OVERVIEW

In the area of Oral Medicine, large amounts of clinical data are gathered every day from conducted oral examinations. Recording this data in the traditional paper-based manner presents problems when the data is to be processed or analyzed. The MedView project aims at providing ways to formalize, store, and subsequently computationally analyze the clinical data, making it more attractive for educational and research purposes. One use of the formalized data is to have the system generate patient medical records and summaries, using a generation context created by the user himself. The MedView project has produced a large knowledge base of clinical data within the area of Oral Medicine. This data, together with user-customizable templates and translators, is input to a text generator engine that in turn produces a medical document. By using various combinations of templates and translators, along with different graphical template contexts, documents with varying style and intended audience can be generated. Examples of such documents are medical records, discharge summaries, and personalized patient information.

SUMMARY/CREATOR

The generation components are created in an application called SummaryCreator. Here the user can insert terms into a template, where each term is translated at generation time by a translator. A template basically consists of a number of sections, where each section contains text with slots in which translated examination data is inserted at generation time. The translator acts as an intermediary between the data in an examination record and what is inserted into a slot. Different template and translator combinations can be used to produce a wide variety of documents in different languages based on the same data. Since the users create the components themselves, each clinician can customize the components to fit his or her own preferred style. Furthermore, the user is presented with the option to preview the currently worked-on template and translator pair, providing instant feedback on how the two components interact.

A term can have one or several values, where each value can be translated according to the translation string entered for the value in the translator. The translation string may contain macros, an example of such is a macro indicating that a certain value should cause the line containing the term to be removed. Macros can be mixed with the translation text in arbitrarily complex compositions.

TEXT GENERATION

To have the system generate a document, the user selects which template and translator combo to use, along with the examination records of interest and the sections to include. The generation process basically consists of removing the non-chosen sections, cloning the remaining text a certain number of times (depending on the number of examinations), building a so-called parse tree for each examination and finally processing each tree. The first three stages of this process are illustrated below. To the right is an example of an application utilizing the text generation subsystem.