Getting Started with ROSANA$_{Deutsch}$

Roland Stuckardt
D-60433 Frankfurt am Main, Germany
e-mail: roland@stuckardt.de

23rd August 2005

1 Synopsis

This document briefly describes how to install and use the ROSANA$_{Deutsch}$ System (Version 5.1.1, August 2005) and the sample data (texts, parses, keys) under a Linux environment. ROSANA$_{Deutsch}$ should work under other operating systems as well, provided that Common Lisp is available.

2 Licensing conditions

ROSANA$_{Deutsch}$ is made available on condition that the user unconditionally accepts the "**License Agreement for ROSANA and ROSANA$_{Deutsch}$**" as stated in appendix A of the document at hand. In using ROSANA$_{Deutsch}$ or any component that comes along with the ROSANA$_{Deutsch}$ distribution, the user acknowledges that he/she has read through and understood the License Agreement for ROSANA$_{Deutsch}$ and unconditionally accepts it.

3 Prerequisites

- Common Lisp (e.g., Xanalys Lispworks 4.2.0 for Linux, or Allegro Common Lisp of Franz, Inc.)
- a reasonably equipped PC (500 MHz, 128 MB RAM should do)
- basic knowledge of Lisp
- in-depth knowledge of anaphor resolution and the respective formal evaluation issues
- the willingness as well as the ability to cope with an experimental software system for which virtually no documentation is available

4 Installation

1. Unpack the gzipped distribution file in a suitably chosen installation directory (subsequently referred to as `$ROSANAPATH`):

   ```
   > tar xvzf ROSANA511_DEUTSCH.tgz
   ```

   ROSANA$_{Deutsch}$ 5.1.1 unpacks to two subdirectories:

   - `$ROSANAPATH/ROSANA511_DEUTSCH/System`, which contains the Common Lisp sources, and
   - `$ROSANAPATH/ROSANA511_DEUTSCH/Data`, which contains the sample data.

2. Change to the source code directory:

   ```
   > cd ROSANA511_DEUTSCH/System
   ```

3. Adaptation of the installation path: in file *schnittstellen-definitionen.lisp*, change the constant ROSANAHOME according to the context of the local installation.
   E.g. if `$ROSANAPATH = ~/CommonLisp` then modify the constant definition to

   ```
   (defconstant ROSANAHOME "~/CommonLisp/ROSANA511_DEUTSCH")
   ```

4. Starting your Common Lisp environment: in case you are using Xanalys Lispworks 4.2.0 for Linux, type

   ```
   > lispworks-4200 &
   ```

5. Compiling the Common Lisp sources of ROSANA$_{Deutsch}$: load the file *compile.lisp* in a Lisp listener; this should compile the sources and generate the respective binaries (ufsl, fasl, depending on the Common Lisp environment you use):

   ```
   CL-USER 1 > (load "compile")
   ```

5 Using ROSANA$_{Deutsch}$ under Xanalys Lispworks

If you are using Xanalys Lispworks 4.2.0 for Linux, a simple graphical interface is provided.

1. *Starting the graphical version of ROSANA$_{Deutsch}$:* load the file *capi-rosana.lisp*:

   ```
   CL-USER 2 > (load "capi-rosana")
   ```

   During first-time use, a copyright and licensing message will be displayed. Please read carefully through the "**License Agreement for ROSANA and ROSANA$_{Deutsch}$**" that comes along with this distribution (see appendix A). Upon acceptance of the licensing conditions, ROSANA$_{Deutsch}$ will be loaded and initialized, and a frame titled ROSANA$_{Deutsch}$ will be opened which displays the ROSANA$_{Deutsch}$ listener. The ROSANA$_{Deutsch}$ frame offers some basic means to process the sample data and to modify the processing and trace settings of ROSANA$_{Deutsch}$.

2. *Processing sample data:* in the `Resolve` menu, select one of the sample data suites, e.g. `Mozart [ ".cod", scoring + ]`. This will paste a function call into the listener buffer of ROSANA$_{Deutsch}$. Execute this call in the ROSANA$_{Deutsch}$ listener:

   ```
   SEM 1 > (Mozart-sco)
   ```

   This invokes the anaphor resolution process. The results are written to the respective directory of the selected sample data. In the above case, the name of the directory is `$ROSANAPATH/ROSANA511_DEUTSCH/Data/Mozart_deutsch` .
   Alternatively, one may use the anaphor resolution interface functions *cod-anares* or *cod-po∗-anares*, which make available a more comprehensive set of options (see section 9).

3. *Browsing the generated output files:*

   (a) Switch to the directory of the processed sample input suite:

   ```
   > cd $ROSANAPATH/ROSANA511_DEUTSCH/Data/Mozart_deutsch/
   ```

   (b) Examine the various output files which, as far as the respective output options are selected (see section 11), have been generated by ROSANA$_{Deutsch}$:

   - *dg.ref*: depending on the selected options, this file displays the anaphor resolution output and/or the traces;
   - *dg.ana*: contains the coreference classes as identified by ROSANA$_{Deutsch}$ (used for scoring purposes);
   - *dg.sco*: depending on the selected scoring options, this file displays the evaluation results of ROSANA$_{Deutsch}$ in various evaluation disciplines (occurrences, coreference classes, immediate antecedents, lexical anchors).

   See section 8 for a comprehensive list of the sample data file types.

   (c) Compare the output with the *SHOULD_BE* reference files that come with the distribution:

   ```
   > diff dg.ref dg.ref_SHOULD_BE
   > diff dg.ana dg.ana_SHOULD_BE
   > diff dg.sco dg.sco_SHOULD_BE
   ```

   If the ROSANA$_{Deutsch}$ installation works properly, no differences will be found.

4. *Experimenting with the output settings:* you may experiment with the different trace options offered in the `AR Output` menu. These settings determine the comprehensiveness of the generated output (antecedents, candidates, salience computation, syntactic disjoint reference verification etc), as written to the *.ref* file. (See section 10 for a description of the most important options.)

5. *Experimenting with the processing settings:* further options are offered in the `Processing Traces` and `Processing Settings` menus. Of particular use might be the following `Processing Traces` options:

- whether the scoring results are written to a *.sco* file (default) or to the listener buffer,
- whether the anaphor resolution output is written to a *.ref* file (default) or to the listener buffer,
- whether a more detailed output regarding the scoring of
  - coreference classes
  - occurrences
  - lexical anchors

  will be generated.

See sections 11 and 12 for further details.

6 ROSANA$_{Deutsch}$ under other Lisp environments

ROSANA$_{Deutsch}$ should work under any Common Lisp Environment. In fact, it was originally developed under Allegro Common Lisp for Linux of Franz, Inc, and virtually no modifications were necessary to make it run under Xanalys Lispworks. Under environments other than Xanalys Lispworks, no graphical user interface will be available. The above described functionality, however, is still fully available. It may be accessed through function calls issued directly in the Lisp listener.

1. *Starting the listener version of ROSANA$_{Deutsch}$:* load the file *rosana.lisp*:

   ```
   CL-USER 2 > (load "rosana")
   ```

   During first-time use, a copyright and licensing message will be displayed. Please read carefully through the "**License Agreement for ROSANA and ROSANA$_{Deutsch}$**" that comes along with this distribution (see appendix A). Upon acceptance of the licensing conditions, the binaries will be loaded, and ROSANA$_{Deutsch}$ will be initialized to some default settings regarding the anaphor resolution output, the processing traces, and the processing options.

2. *Change to Package* `SEM`*:*

   ```
   CL-USER 3 > (in-package "SEM")
   ```

3. *Processing sample data:* anaphor resolution may be performed by issuing one of the sample calls (as output by the help function *(help-ar)*). E.g. the predefined call

   ```
   SEM 4 > (Mozart-sco)
   ```

   starts ROSANA$_{Deutsch}$ on a document *dg* (= Don Giovanni) of a corpus of Mozart Opera plot descriptions. (At current, only this single sample document is included.) Alternatively, one may use the interface functions *cod-anares* or *cod-po∗-anares*, which make available a more comprehensive set of options (see section 9).

4. *Browsing the generated output files:* as described in section 5.

5. *Experimenting with the output, trace, and processing settings:* these settings may be changed by directly typing in the function calls `(configure-ar-output)`, `(configure-processing-traces)`, and `(configure-processing-settings)`. E.g.

   ```
   SEM 5 > (configure-ar-output)
   ```

   allows the user to interactively select among the set of available anaphor resolution output options.

6. *Getting help:* the available sample calls and configuration options might be redisplayed by typing

   ```
   SEM 6 > (help-ar)
   ```

7 About the sample data

ROSANA$_{Deutsch}$ comes with sample data for some documents of a corpus of Mozart Opera plot descriptions, comprising ASCII versions of the texts, the respective parses (as determined by the Connexor parser for the German Language), the respective key data (coreference annotations), and some supplementary information. At current, only one document (*dg*, = Don Giovanni) is included, which comprises 1,506 tokens.

8 About the file types and the workflow

The different kinds of knowledge are distinguished through filename suffixes. Files used as input to anaphor resolution and evaluation/scoring are:

- *filename.txt:* the numbered collection of source documents;
- *filename.cod:* parsing results as determined by the Connexor parser for German (as provided by the Connexor web demo (*http://www.connexor.com/demo/syntax/*) in July/August 2005);
- *filename.par:* parsing results as processed by the predecessor system ROSANA51 for English, i, e. as determined by the Dependency Parser for English ([1], a predecessor parser of the Connexor parser for German) - still of relevance as ROSANA$_{Deutsch}$ employs token-numbered versions of this format as the base for performing anaphor resolution (using preprocessor module *cod-2-par-or-pot.lisp* (see included documentation) that maps *filename.cod* to either *filename.pot* or *filename.par/.pos*);
- *filename.pos:* token-numbered parsing results, including surface position numbers that specify the positions of the tokens in the source document; this information is generated automatically from the *.par* file when employing the separately available Emacs-Lisp-based tool *dpe-annot.el* for coreference annotation;

- *filename.pot:* (generated by the *filename.cod* preprocessor *cod-2-par-or-pot.lisp*) as *filename.pos*, but without token positions;
- *filename.ngp:* supplementary (general and corpus specific) lexical information;
- *filename.key:* intellectually gathered key data (coreference classes, referred to during evaluation);
- *filename.tag:* intellectually gathered key data (coreference classes, immediately annotated in *.pos* file, synchronized with (i.e. referentially equivalent to) *filename.key*).

Files generated during anaphor resolution and scoring are:

- *filename.ana:* the coreference classes as identified by ROSANA$_{Deutsch}$ (referred to during scoring);
- *filename.ref:* anaphor resolution output as determined by the settings in the `AR Output` menu (only available if the processing traces option *Write AR Output to .ref File* has been chosen);
- *filename.sco:* depending on the selected scoring options, this file displays the evaluation results of ROSANA$_{Deutsch}$ in various evaluation disciplines (as far as selected: occurrences, coreference classes, immediate antecedents, lexical anchors) (only available if the processing traces option *Write Scoring Results to .sco File* has been chosen).

Figure 1 illustrates the role of the different file types in the workflow of ROSANA$_{Deutsch}$. Two use cases are distinguished:

1. employing ROSANA$_{Deutsch}$ directly on the Connexor output, thus taking the *cod* file as the input and, by calling interface function *cod-anares*, applying the horizontal cascade of modules shown in figure 1, viz.: producing the intermediate *.pot* representation[1] and then performing anaphor resolution on it;
2. going the indirect way via the annotation tool *dpe-annot.el* for producing coreference annotations to be used during formal evaluation; here, the workflow comprises two steps involving ROSANA$_{Deutsch}$: (2a) calling interface function *cod-2-par*, which produces the *.par* file on which the annotation tool *dpe-annot.el* works (followed by actually employing *dpe-annot.el* for corpus annotation, which happens external to ROSANA$_{Deutsch}$); (2b) calling interface function *cod-po*-anares* in order to perform anaphor resolution on the *.pos* file produced by *dpe-annot.el*.

Regarding the predefined sample function calls, `(Mozart)` and `(Mozart-sco)` correspond to use case (1), whereas `(Mozart-pos)` and `(Mozart-sco-pos)` correspond to step (b) of use case (2). In these predefined calls, it is further distinguished between whether or whether not the *.ana* coreference output of ROSANA$_{Deutsch}$ is formally evaluated against the *.key* data accompanying the distribution. However, it is most natural to employ `(Mozart-sco-pos)`

[1] The *.pot* representation plays a further role for relating the output of ROSANA$_{Deutsch}$ ({*.ana,.ref,.sco*} files) to the input as the occurrence numbers assigned during anaphor resolution correspond to the document-unique token numbers contained in the *.pot* representation.

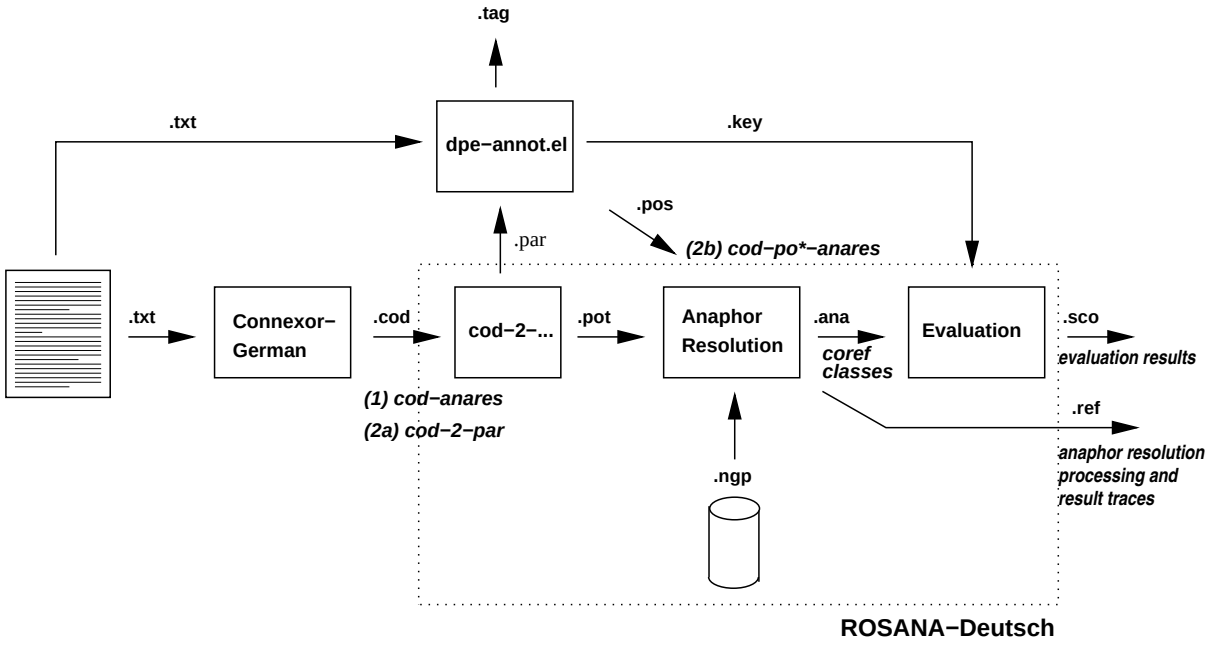

Figure 1: File types and workflow

(step (2b) only, in general) when evaluation is desired, as the existence of the required *.key* file implies that the respective *.pos* file is already available, thus making it obsolete to again convert the *.cod* parse into the required *po*{*s,t*} representation.[2]

9 Calling the Interface Functions of ROSANA$_{Deutsch}$

Instead of using one of the predefined shortcuts, anaphor resolution might instead be invoked by issuing interface function calls in the Lisp listener, thus having available a more comprehensive set of options. A typical call of the interface function *cod-anares*, which implements the above use case (1), looks as follows:

```
SEM 7 > (cod-anares
         "Mozart_deutsch/dg.cod"    ;; (1) document parses
         nil                        ;; (2) external morph. data (obsolete)
         "Mozart_deutsch/dg.key")   ;; (3) key data to be used for scoring
```

The most important parameters of this call, which is equivalent to the predefined call `(Mozart-sco)`, are: parameter *(1)*, which specifies the document collection parses to be processed; parameter *(3)*, which specifies the key data file to be used for evaluation (if set to *nil*, the output won't be scored). At current, it is assumed that the *cod* file contains the Connexor parses of a *single* text.

[2]Moreover, this ensures the (essential) consistence of the *.po*{*s,t*} representation to be processed with the employed *.key* data, which might no longer hold if one proceeded according to use case (1) and, e. g., there were changes to the Connexor parser.

Step (b) of use case (2) can be invoked by calling the second interface function, which works on the intermediate (token-numbered) representations of the Connexor parses (in particular, on the *pos* files, as generated by *dpe-annot.el*, see above):[3]

```
SEM 8 > (cod-po*-anares
         "Mozart_deutsch/dg.pos"    ;; (1) token-numbered doc. coll. parses
         nil                        ;; (2) external morph. data (obsolete)
         "Mozart_deutsch/dg.key"    ;; (3) key data to be used for scoring
         (dnum-nach-dname '(1 1))   ;; (4) specificaton of a document sub-
                                    ;;     set to be processed (optional)
```

In this case, which is equivalent to the predefined call `(Mozart-pos-sco)`, the file containing the parses might contain multiple documents separated by tags *"% DOKUMENT i"* ($1 \leq i \leq n$). Correspondingly, there is an additional parameter *(4)*, which selects for resolution a subset of documents of the collection (if not provided or set to *nil*, *all* documents of the collection will be processed). Parameter *(4)* refers to the document number tags that are provided in the *.pos* or *.pot* file (e.g. *"% DOKUMENT 1"*); the function *dnum-nach-dname* serves as a tool for generating respective tag reference lists for the documents to be selected, e.g.

- by specifying *(dnum-nach-dname '(32 66))*, documents with numbers 32 to 66 (i.e. an interval) will be processed;
- by specifying *(dnum-nach-dname nil '(1 7 18))*, exactly the documents with numbers 1, 7, and 18 will be processed.

This is of particular use for distinguishing between development/training and evaluation subsets of larger document collections.
Regarding both interface functions, there is a further optional parameter (not shown in the above examples) that allows to explicitly specify the data directory prefix under which it is looked for the *.cod*, *.pos*, *.pot* and *.key* files, and in which the result data are written. If ROSANAHOME has been set to `$ROSANAPATH/ROSANA511_DEUTSCH` (see section 4), this directory prefix defaults to `$ROSANAPATH/ROSANA511_DEUTSCH/Data`.
Finally, step (a) of use case (2) - generation of a respective *.par* file from a *.cod* document collection parse - can be accomplished by the call

```
SEM 9 > (cod-2-par
         "Mozart_deutsch/dg.cod"    ;; (1) document collection parses
         "Mozart_deutsch/dg.par")   ;; (2) the output file name
```

As above, there is a further optional parameter (not shown in the example) that allows to explicitly specify the data directory prefix.

[3] The function processes the *pot* files as well, which are generated as intermediate representations in use case (1); hence its name *cod-po*-anares*.

10 Anaphor resolution output options

In the graphical user interface of the Xanalys Lispworks version of ROSANA$_{Deutsch}$, the lower part of the `AR Output` menu provides a number of options through which the anaphor resolution output might be configured according to the specific needs. The most important options are:

- *Candidates:* displaying the list of ranked candidates among which the antecedent occurrence is selected;
- *Salience:* displaying information about the computation and assignment of salience / preference weights;
- *Antecedents:* displaying the selected antecedent occurrence;
- *Syntactic Disjoint Reference:* displaying details regarding the robust verification of the syntactic disjoint reference conditions (binding principles A, B, C);
- *Parse Tree Results:* displaying the (possibly fragmentary) surface-syntactic information that constitutes, in particular, the base for the verification of the syntactic disjoint reference conditions;
- *Compact Output:* if selected, occurrences (anaphors, candidates, antecedents) will be displayed in a compact format.

The upper part of the menu provides shortcuts to several useful combinations of the above atomic options.
The *configure-ar-output* function of the non-grphical interface makes available an even more comprehensive set of combined, non-exclusive (i.e. combinable) output offers.

11 Processing trace options

Some options that might be of particular use are:

- *Progress of Processing:* if selected, progress information will be displayed in the ROSANA$_{Deutsch}$ listener buffer;
- *Verbose Coreference Class Scoring:* if selected, the scoring output will provide detailed information regarding the coreference classes scoring (illustrating the model-theoretic coreference scoring scheme of Vilain et al. ([4]) and its extension by Stuckardt ([3]));
- *Verbose Occurrence Scoring:* if selected, the scoring output will provide detailed information regarding the scoring of the identified occurrencs;
- *Verbose Lexical Anchor Scoring:* if selected, the scoring output will provide detailed information regarding the scoring of the lexical anchors that have been determined for pronominal anaphors;

- *Scoring of Immediate Antecedents:* if selected, the performance with respect to the identification of immediate (possibly pronominal) antecedents will be scored as well;
- *Write Scoring Results to .sco File:* if selected, scoring results will be written to a *.sco* file (otherwise they will be output to the ROSANA$_{Deutsch}$ listener buffer);
- *Write AR Output to .ref File:* if selected, the anaphor resolution output will be written to a *.ref* file (otherwise it will be output to the ROSANA$_{Deutsch}$ listener buffer).

12 General processing settings

Some options that perhaps may be of use are:

- *Apposition as Coreference Trigger:* (recommended) if selected, appositive relations will be interpreted as syntactic coreference triggers;
- *Coreference Class Scoring on Correct Occurrences only:* (strongly recommended) if selected, model-thoretic coreference class scoring will be confined to the subset of correctly identified occurrences, i.e. to occurrences in the intersection set of system output and key; through this, the two evaluation disciplines are mutually decoupled in order to avoid certain cases of scoring anomaly (see [3, 2]);
- *Determining Lexical Anchors for Pronouns:* if selected, ROSANA$_{Deutsch}$ tries to determine lexical anchors (i.e. occurrences providing conceptual content, e.g. common nouns or names) for pronominal anaphors; this discipline might be paraphrased as anaphor resolution proper.

References

[1] Timo Järvinen and Pasi Tapanainen. *A Dependency Parser for English.* Technical Report TR-1, Department of General Linguistics, University of Helsinki, 1997.

[2] Roland Stuckardt. *Qualitative Inhaltsanalyse durch Computer - ein uneinlösbarer Anspruch? Untersuchungen zur algorithmischen Textinhaltserschließung am Beispiel der referentiellen Interpretation*, Ph.D. dissertation, Department of Social Sciences, Johann Wolfgang Goethe University Frankfurt am Main. Also: Tenea, Berlin, 2000.

[3] Roland Stuckardt. *Design and Enhanced Evaluation of a Robust Anaphor Resolution Algorithm*, In: *Computational Linguistics* 27(4), 2001, 479-506.

[4] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman, *A Model-Theoretic Coreference Scoring Scheme.* In: *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, Morgan Kaufmann, San Francisco, 1996, 45-52.

A License Agreement for ROSANA and ROSANA$_{Deutsch}$

This agreement is made and entered into as of _____ (dd/mm/yy) by and between the parties:

a)
Dr. Roland Stuckardt
Im Mellsig 25
D-60433 Frankfurt am Main
Germany

referred to as: "**LICENSER**"

b)

[The user who gained access to ROSANA and ROSANA$_{Deutsch}$ immediately through LICENSER]

referred to as: "**USER**"

A.1 Product

ROSANA and ROSANA$_{Deutsch}$ are Common Lisp implementations of a robust syntactic salience-based anaphor resolution algorithm.

ROSANA and ROSANA$_{Deutsch}$ come with copy-righted(!) sample document collections, comprising ASCII versions of the texts and, depending upon distribution, the respective parses, the respective key data (coreference annotations), and supplementary information. An evaluation module is included which scores the output of ROSANA and ROSANA$_{Deutsch}$ according to various evaluation disciplines.

A.2 Copyright

Copyright of ROSANA and ROSANA$_{Deutsch}$ is to Dr. Roland Stuckardt.

A.3 License

LICENSER grants USER a non-exclusive license to use ROSANA and ROSANA$_{Deutsch}$, versions as distributed. USER agrees to use ROSANA and ROSANA$_{Deutsch}$ only for non-commercial, non-profit research purposes; to report changes that USER makes to the programs to LICENSER; and to acknowledge the use of ROSANA and ROSANA$_{Deutsch}$ in all

publications reporting on results produced with the help of ROSANA or ROSANA$_{Deutsch}$, including publications on webpages.
Use of ROSANA, ROSANA$_{Deutsch}$, or products derived from ROSANA or ROSANA$_{Deutsch}$ for any commercial purposes requires explicit written agreement of LICENSER.

A.4 Non-Disclosure

ROSANA and ROSANA$_{Deutsch}$ will be held in confidence by USER and will not be disclosed by USER to third parties.
USER shall and will employ all necessary precautions to ensure that no persons or institutions other than persons as are in the employ of USER or in the same research project as USER will get access to ROSANA or ROSANA$_{Deutsch}$ or parts thereof. Other persons or institutions desiring access to ROSANA or ROSANA$_{Deutsch}$ should be directed to LICENSER to obtain separate license agreements.

A.5 Fee

This license agreement is granted by LICENSER to USER free of charge.

A.6 Termination

This license agreement will terminate upon thirty (30) days' written notice by either party to the other party.

A.7 Disclaimer

ROSANA, ROSANA$_{Deutsch}$, and its documentation are provided on an "as is" basis, with no guarantee of its veracity or accuracy. No liability is accepted for any damage caused by its use.

Frankfurt/Main,	_____ (dd/mm/yy)	_____ (place)	_____ (dd/mm/yy)
LICENSER		USER	
Signature	_____	Signature	_____
Name	Dr. Roland Stuckardt	Name	