# Decision support in the routing of translation projects

*Report on the Regensburg prototype of TransRouter*

Rainer Hammwöhner

Department of Information Science
University of Regensburg

# Table of contents

# 1. Introduction

Quality and efficiency of translation services depend on the appropriate routing of translation projects. The choice of a translation route requires knowledge about technology at hand, the resources available and relevant project parameters. For several reasons this information is difficult to obtain for translation managers.

- Translation technology is undergoing a swift development process. As a consequence new approaches to translation support or new systems will not be considered in decision making.
- The need of an organisational memory covering the resources (e.g. translation memories, term banks) at hand is not recognised by a majority of translation agencies.
- Relevant project parameters may not be estimated easily. Is the text repetitive and to which degree? Is special terminology employed? Is the text to complex to use a machine translation system?

The objective of the TransRouter project is the design of a decision support system providing the translation manager with all the relevant information. Since no widely accepted notion of decision support can be built on in this application field, the general approach of TransRouter is based on the development of a sequence of prototypes which are presented to the public. These prototypes share the following features:

- Several profiles contain relevant features of agents (translators or translation tools) and resources (translation memories or term banks)
- The features of translation projects are covered by a different type of profiles.
- A set of tools is developed which allow the (semi)-automatic acquisition of project data (e.g. text size, terminology, complexity, repetitiveness).

Nevertheless there are two kinds of prototype being developed. Two prototypes developed by the LRC emphasise the integration of tools and the estimation of time and cost factors. These prototypes are implemented in C++, thus aiming more directly at a commercial market. An additional prototype, which is developed at the University of Regensburg, is conceived of as a more experimental platform that allows the testing of user interfaces and decision models. To support this flexible and experimental approach this prototype is implemented in Smalltalk (VisualWorks 3.0). To minimise double work between project partners, the emphasis in design and development of the Regensburg prototype is put not on the integration of tools and not on the content but on the form of decision making within TransRouter. As a consequence the structure of this prototype is that of a shell system allowing the efficient and user friendly redefinition of most of the content rules contained in the system by simply using the graphical user interface. This focus of prototype development does not mean that tool integration would not be possible within this prototype or that the content rules of the decision process were not reasonable. It simply means that these questions are neither stressed in the development of the prototype nor in this presentation.

The profiles implemented in the prototype were derived from the definitions in the TransRouter report on component tools and profiles (D3.2). Some few profiles were slightly simplified, some others were enhanced to fit the needs of the calculation process. The process of agent and resource selection, of combining them with routes and of calculating overall route features will be described in some detail.

The major objective of the TransRouter system as has been pointed out already is to give support in deciding on an appropriate route for a translation project. But this task consists of a number of steps, which need specific support. Additionally there will be services provided by TransRouter, which are of interest not only for the translation manager but for others as well. Thus, the first part of this report will give an overview of the Regensburg prototype focussing

on the workflow being supported by the system. There will be some remarks on user classes and access permissions as well.
As a consequence of this workflow centred approach it will be shown that there are lots of functions relevant to a system like TransRouter besides mere route assessment and route selection.
Before a detailed discussion can be started, the nature of the intended support should be pointed out. The goal of TransRouter is not to find *the* optimal route for a translation automatically by mapping project data directly to a route suggestion. This approach would need formalised rules of decision making in translation projects, which are not at hand. Furthermore it seems to be questionable whether translation managers would accept a system which would seem to take over the responsibility of decision making. Thus, TransRouter will not make decisions but support the manager in decision making by pointing out alternatives in agent and resource selection and the resulting route choice. TransRouter will give support in the acquisition of the relevant project data, the resource assessment and the assessment of routes with respect to time, costs and quality.
If in the following *TransRouter* is mentioned as the name of a system with specific capabilities this is meant as an abbreviation for the *Regensburg prototype of TransRouter*.

## 2. Workflow

This section will provide a short overview of the general workflow in TransRouter. More details of the workflow will be introduced in the following sections.

- Data acquisition: The translation manager will expect agent and resource data to be available in the system when he starts to use it. Nevertheless there will be a need to update the system at regular intervals. Furthermore there should be some means to enter project data conveniently.
- Information retrieval: The user might want to extract data from the system without using the inference mechanism.
- Agent and resource selection: The first step of decision making is the selection of those agents and resources, which may be used to process a given project. This step may be performed automatically since it is based on formally defined criteria. Some manual editing may follow the automatic selection step.
- Route construction: The selected agents and resources are used to construct possible routes, which are based on a built-in route model. The system will not generate all possible routes or the optimal ones but a representative set of routes covering all possible route types. The user can refine these routes afterwards.
- Route assessment: The routes, which are created by the system, have to be assessed by the user. He will choose one of several evaluation functions, a cost function, time or quality function or a combination of those. The system will then sort the routes according to those criteria. The manager will pick one or more routes which seem to be promising. He then may modify the set of tools and resources assigned to these routes.
- Route selection and route processing: when route assessment and refinement are completed, a route can be selected for the further processing of the translation project.
- Data storage: The main goal of TransRouter will require the persistent storage of agent and resources profiles. Project data do not necessarily have to be entered into the archives. It will be shown, however, that a couple of functions which can be of vital interest to a TransRouter user can be based on data of projects which have been processed in the past.

The steps as mentioned above give an outline of a macroscopic workflow in the use of TransRouter. At least some agent profiles must be defined before the system can be used for decision support. The refinement of existing profiles and the addition of new ones will occur at certain intervals. When a request for a new project is to be answered, the translation manager will probably try to get a rough overview of the agents and resources which could be useful for this project. Only if this request is successfully answered, will he make the effort to elaborate a complete project profile as the starting point for further route elaboration. This will then lead to the steps of agent and resource selection, route elaboration and so on.
If the agent profiles are kept up to date – only in this case will TransRouter be a success – access to this information may be of interest to others and not only to the translation managers. Thus, it could be a reasonable approach to open segments of the TransRouter data to the entire organisation via an intranet or to a wider public via the Internet. The software structure of the prototype contains a gateway to the WWW. This interface, however, could not be optimised within the scope of this project.

# 3. Browsing of data and data acquisition

A need for data acquisition will arise in the following cases.

- Acquisition of agent data
- Acquisition of resource data
- Entering of project parameters
- Setting of general system parameters

TransRouter will provide a set of tools allowing for form-based interaction with project, agent and resource profiles. The structure and content of these forms depend on the type of object, which is displayed. The general interaction mechanism is the same for all object types. Object attributes can be selected from a list or filled into text fields. Most widgets allow the selection of features from context sensitive pop-up menus and the modification of objects via drag and drop.
These tools can be used for data presentation and acquisition. The tools will offer only those functions to the user which are allowed by his user profile.

## *3.1. User profiles and access permissions*

TransRouter allows the identification of individual users. A user profile describes the kind of data a user is permitted to access or modify. Read or write protections apply – with some exceptions mentioned below - to types but not to individual objects. If a user is allowed to modify agent profiles he may do this with any agent profile in the system. When TransRouter is started, it will log-in to a *guest* user mode automatically. A user may then log-in to some other profile.

- **Accessing agent profiles:** Any user may read agent profiles. Product data of translation technology are available to everybody. Providing convenient access to these data may be thought of as a service to the public. But there is one exception. Data about human translators or service providers require specific access permission.
- **Modification of agent profiles:** The need to modify agent profiles will arise only when new products arrive on the market or when there are new contracts with translators or freelancers. The permission to modify these profiles can therefore be restricted to a few people.
- **Accessing resource profiles:** The content of resource profiles gives some information on the internal processes of a translation agency. Thus, they may not be accessible to the public. Within the organisation everybody should be able to access these data.
- **Modification of resource profiles:** Resources are subject to modification within almost every translation project. A project manager should have the permission to modify the profiles of the resources being used in the scope of his projects. TransRouter can even do some updates to resource profiles automatically.
- **Accessing project profiles:** Within the organisation everybody may access project profiles.
- **Modification of project profiles:** The project manager should be able to modify the profiles of his projects. Modification of other projects needs specific permission.
- **Modification of the system parameters:** Only a system administrator will be allowed to modify the system parameters.

### 3.2. Modification of agent profiles

For the time being TransRouter can offer no more than the opportunity to enter agent profiles using a specific tool. There is no support in obtaining these data. In the long run it would be desirable to integrate tools into TransRouter which will perform benchmarks with translation tools. The result of these benchmarks could then be fed into the profiles automatically.

### 3.3. Modification of resource profiles – assessment of resources

TransRouter offers tools for the editing of resource profiles as well. These profiles have only a few attributes – e.g. language pair –, which are significant for the selection of resources themselves. The validation of resource quality is most important in this context.
The selection of resources is based not only on their inherent features but on an additional assessment, which takes the context of the project – especially the text – into account. Coverage is an indicator of whether a translation memory is useful. The number of unknown terms indicates the quality of a term bank. TransRouter provides tools, which will provide the translation manager with these data. The use of these tools on the other hand – especially in the case of large projects – is time consuming. The number of resources which have to be assessed, should therefore be as small as possible.
A further indicator for the value of resources for a project could be the use of these resources in a prior version of the project. This means that project profiles would have to be kept within the system at least for some time.

### 3.4. Building of project profiles

The definition and updating of a project profile is up to the project manager. Organisational details have to be entered manually. Some features of the text – e.g. average sentence length, text complexity or repetitiveness – will be provided by TransRouter tools, whereas others – like text style or domain – have to be entered manually. Information about text representation – location or format of files – is provided by the project manager or by TransRouter tools – number of files – in a similar way.
If a new project is created as a new version of a project that is known to TransRouter, a significant amount of data – client data, text type, subject area etc. – can be inherited.

### 3.5. Modification of system parameters

As has already been mentioned the system will offer popup menus at several places to allow for a simple definition of project, resource or agent profiles. Thus, the system has to know about appropriate identifiers for text-formats, languages, domains, styles and so on. This information is not coded into the system but may be adapted to the special needs of a translation agency. There are several tools within TransRouter to define these data sets, which can be used by a system administrator.

# 4. Information Retrieval

The whole process of decision support may be conceived as some kind of information retrieval. In this chapter only those search processes will be considered that occur in the system in addition to agent and resource selection or route construction which will be described later on in this text. There are two distinct approaches to information retrieval within TransRouter. Matching oriented retrieval is based on the formulation of a query. This query will be processed by the system and result in the presentation of relevant objects fitting the request. Browsing uses implicitly or explicitly given references between objects, which can be accessed via the TransRouter interface like other object features.

## *4.1. Matching oriented information retrieval*

TransRouter's approach to matching oriented retrieval is based on the construction of sample objects. A user, who wants to find a machine translation system by its specific features, will have to construct an agent profile describing such a system. He will select the agent type, create a sample object, fill in the relevant profile data and then start the retrieval process. All those objects with matching type and features will be found. Truncation symbols ('#' and '*') substituting an arbitrary character or sub-string can be used. If attributes are numeric, a match requires the difference of the respective numbers not to exceed a certain threshold. Dates match if they refer to the same year. Of course more subtle matching algorithms could be introduced. But the overall approach would not be changed significantly.

Matching oriented retrieval may be used for several reasons. A translation manager may want to check the existence of relevant agents and resources before taking the trouble of defining a project profile. He may, for example, simply want to retrieve the profile of a service provider whose name he has forgotten just using the name of town as a search criterion. A guest user may want to get an overview over all machine translation systems, which are capable of handling a certain language pair.

## *4.2. Browsing oriented information retrieval*

Browsing oriented retrieval employs relationships between objects, which are made explicit by the interface. A profile of a translation memory may provide not only the data storage format of this resource but also list translation memory systems which can process this data format. These object references are not necessarily represented in the profile explicitly but can be made explicit by the interface.
One of the most promising applications of browsing in TransRouter seems to be the use of organisational experience. This would require TransRouter to be not only a decision support system but also a project documentation system. The translation manager could then see whether a machine translation system has already been used in successful projects. Did these projects have anything in common with the project currently in progress? Were there any projects at all that were similar to the current one? What can be learned from their performance?
This leads to the question of which notion of similarity should be employed. Should the similarity of projects be based on basic project attributes only or should chosen routes be considered too? Since the similarity measure should be applicable to all projects – even if

newly defined – there must be at least one notion of similarity which does not take routes into account.

The similarity measure should allow a ranking of projects. It would be reasonable to have similarity values between 0 (no match) and 1 (full match). The numeric attributes of a project can be used easily to compute such a measure. Even for symbolic attributes some distance measure could be defined. TransRouter in its current implementation uses a simpler approach.

- If the language pair is not the same, the projects are considered to be not similar at all. This seems to be reasonable since nothing or little can be learned from such projects about the use of tools and resources.
- The same applies to text style and domain.
- Formats are not used for similarity computation since there is only a minor effect on translation routing.

Another application of browsing oriented retrieval is the exploration of tools able to process a given resource – e.g. a translation memory or a term-bank – or to find available resources that can be used when operating a given tool – e.g. a dictionaries used by a machine translation system.

# 5. Agent and resource selection

The first step of route construction is the selection of agents and resources, which are relevant with respect to the project profile. TransRouter is capable of handling several kinds of selection rules, which will be described in this section. A common feature of these rules is that they distinguish only between relevant and irrelevant objects.

## *5.1. Type specific selection rules*

Type specific rules do not apply to individual objects but to object classes. Such a rule may express the fact that machine translation systems in general are of no use for projects with certain features. This kind of selection rules prevents the system from checking general rules with any individual object again and again.

Currently the following type specific rules are built into the system:

- If there is no previous version of a project and no further version is to be expected and if the repetitiveness of the text is below a threshold, then no translation memory should be built.
- A machine translation system should not be used if the complexity of the text exceeds a certain threshold.

## *5.2. Selection rules applying to individual objects*

Most of the selection rules implemented within TransRouter apply to individual objects.

- Agents or resources must support the language pair required by the project.
- Human agents may act in different roles (translator, reviser etc) which represent individual translation services. They must know the required language pair (target language in case of revising) and be able to provide the service asked for at the required quality level.
- Dictionaries and term banks must cover the same content domain as the project's text.
- A translation memory must be derived from a prior version of the same or a similar project. This will prevent the system from doing costly assessments on memories, which probably will not have a sufficient coverage.
- Machine translation systems must be able to provide the required quality.

## *5.3. Weak selection rules applying to individual objects*

These rules cover phenomena which are mere obstacles in the use of a resource or system. Examples are licences being outdated or format not matching. These obstacles will lead to the exclusion of a system or resource if and only if an alternative is at hand. Otherwise the rule will be suppressed in order to get some operational routes. A comment on the problems with these systems / resources is provided.

- Licence must be up to date. If no other tool is available a licence can be updated easily.

- Tools must be able to process the storage format of the project's text. In most cases it should be possible to convert formats with some reasonable effort.
- Tools must be able to produce the desired destination storage format (see above).
- Resources must be approved by a translation manager.
- Human translators must be knowledgeable in the content domain of the text and know about the relevant text styles. If nobody is at hand who has this knowledge, somebody knowing the languages should be able to do at least a low quality translation. A good reviser can perhaps sort out quality problems in the last step of the route.

### *5.4. Check tools and resources*

When the selection rules which were described above, have been applied by the system, a set of agents and resources will be left which can be considered as relevant to the project. Nevertheless one further selection step has to be performed. If there is a resource selected but there is no tool which can be used to process this resource, it has to be removed.
If a tool is selected which requires some kind of resource which is not available, the system will check whether an empty resource should be created. This may be the case especially for translation memories. If no resource can be created, the tool will be removed from the list.

### *5.5. Grouping of objects*

As a result of the selection process a set of relevant agents and resources is extracted from the TransRouter data repository. If some objects of the same type are contained within this set they may be used either alternatively or – in some cases – in cooperation. Some translators may form a team. Several term banks, dictionaries or translation memories can be integrated to form a new bigger one. The resulting object will have new features, which can only be inferred to a certain degree. If such a grouping of objects is feasible TransRouter will construct a group object.

- Teams are groups of translators. The translation performance of a team is the sum of the performance values of its members reduced by some organisational overhead. This overhead depends on team size and to a lesser extent on project duration. The optimal team size therefore will increase with the size of the project. The existence of a team profile is restricted to the course of a single project.
- Groups of terminology banks, dictionaries or translation memories will be built if there is a tool available which can process multiple resources. TransRouter will group only those resources, which can be processed by the same agent. Thus, several groups of a similar type may be built containing resources with various data formats. Since there will probably be some overlap in content, the number of entries in a group of dictionaries or the coverage of a set of translation memories can't be estimated simply from the features of a group's elements. The resulting new objects need their own assessment with respect to the project.

### *5.6. Manual manipulation of the relevance set*

After the completion of the automatic selection process the translation manager may want to reduce the relevance set further. He may know that some translator is occupied by other

projects or that he for some reasons does not want to use a specific tool etc. Removing agents or resources at this stage of the decision process will simplify the task of route construction.

# 6. Route construction

The explanation of the route construction process first needs some introduction to the route model of TransRouter. Then it can be shown how a route will be furnished with agents and resources.

## *6.1. The route model of TransRouter*

The route model of TransRouter is comparatively simple. A route basically consists of three processing steps. Each step is performed by one main agent using a set of tools operating on a set of resources associated with this step. The **pre-processing step** covers all activities, which are necessary to prepare text and or resources – initial proof-reading, enhancing dictionaries. Within the **translation step** the translation process is performed whereas the **post-processing step** deals with all activities following the translation until the end of the project – e.g. proof-reading, formatting. Pre- and post-processing steps are performed by humans. The main agent of the translation step may be a tool as well (e.g. a machine translation system). The type of a translation step or a route will be defined according to the nature of the main agent. Because each type of main agent has its own requirements regarding pre- and post-processing, there are special subtypes for these steps too. The route type will have consequences on the time, cost and quality estimations as well.

TransRouter supports the following route types:

- Translation by a translator who is employed by the agency
- Translation by a service provider
- Translation by a machine translation system
- Translation by a translation memory system (automatic mode)

Human translators can be assigned to various roles in the translation process. They can be the main agents of the translation step, revisers, pre- or post-editors. Each of these activities requires its own sub-profile to be filled out within the agent's profile.

## *6.2. Generation of routes, assigning agents to routes*

The system will use all agents from the relevance set which may serve as a route's main agent. This is the case for human translators, service providers and machine translation systems. The system will create a route for each of these agents, who will then be the main agent of the translation step.
The route will then control the assignment of main agents to the other processing steps. The rules as formulated here are a first approach and need some refinement.

- Routes involving service providers have no need of pre- and post-processing steps, since the whole translation process is under control of the service provider.
- Routes involving human translators assume that there is one main agent responsible for all translation steps. Nevertheless, if the human translator has no sub-profile for text revision some other translator will be chosen for the post-processing step.
- If several human agents – translators or post-editors – are available for a specific route involving machine translation the system will choose according to the best performance.

### *6.3. Furnishing routes with tools and resources*

TransRouter has some basic understanding of which kinds of agents and resources can be combined and which kinds of route steps they may be assigned to. The system will not try to generate and assess all possible combinations of main agents, tools and resources but to find some reasonable equipment for each step. This process starts with the translation step. The system will select resources first, because the content of a term bank or a translation memory is assumed to be prior to the effects of handling software. In a following step the system will find the optimal tool for each of the selected resources. Finally those tools which do not need any resources (e.g. an alignment tool) will be assigned to a translation step. This approach has two implications. The system must provide the means of finding a ranking of resources and tools in order to find the best fit. The solution found may not be the optimal one, because a slightly less optimal resource may be processed by some more user friendly or efficient tool, which could not be used for the resource selected.
The equipment of the auxiliary steps will follow almost the same procedure with the only exception that, if possible, the same resources and tools will be used as assigned to the main step.

#### 6.3.1. Consistency rules for translation steps

Not every combination of translation steps and agents is possible. The following rules apply:

- The agents of pre- and post-processing steps are human translators.
- The step type defines the agent type of the main step.
- If the main agent of the translation step is a tool, the main agents of the pre- and post-processing steps must know how to use the tool.
- The profile of a human translator must indicate that he may take the appropriate role – translator, reviser, pre- or post-editor – in the route step. This means that a processing performance greater than 0 must be assigned to this specific activity.
- A reviser is a senior translator checking the output of a human translator. He must know the source and target language. The target language should be his native language.
- A post-editor improves the output of a machine translation system. He must know the system and the required language pair.
- A pre-editor will prepare a text to be translated by a machine translation system. He will do spell checking and enhance the dictionaries if necessary. He must know the language pair and the system.

#### 6.3.2. Sorting of resources

The most straightforward approach to the sorting of resources implies the use of resource assessments.

- Translation memories would be sorted according to the coverage of the project's text.
- The sorting of term banks would make use of the information on the number of unknown terms within the text.

Unfortunately the assessment of resources is time consuming. Therefore it can't be assumed that all resources which are of some relevance to the project are assessed. Thus, TransRouter has to employ two sorting strategies. If all resources of some type are assessed, TransRouter will use the assessments for sorting. If this is not the case, TransRouter has to use an alternative strategy using basic resource features for sorting. All resources then are sorted

according to their validated quality. The following type specific sorting criteria apply additionally.

- Translation memories will be sorted according to their position in the version hierarchy. The translation memory which is most recent with respect to the ongoing project, will be preferred. It is most likely that this memory will have the best coverage.
- A good indicator to estimate the quality of a term bank is its size. This largest term bank probably will have the best terminology coverage.

#### 6.3.3. Sorting of agents

The sorting of agents imposes fewer problems than the sorting of resources. Basic features of the respective profiles may be used.

- Translating (translators, machine translation systems) agents are sorted according to translation quality and performance.
- The relevant features of service providers are quality and costs.
- Translation memory systems judged according to their performance (average access and storage time).

Nevertheless, some of these data can be fully estimated only with knowledge of the complete route data. The performance of a translator for instance depends on the tools at hand. The quality of a machine translation is related to the quality of the resource being used. TransRouter will feed as much information into this sorting and ranking process as is available within the current state of decision making. In an early step only the agent profile will be available, in a next one a project profile will be added. Finally all data of the route and route steps currently being elaborated are available and can be used for agent assessment.

### *6.4. Dependencies between agents*

The sorting process as described above does not take into account any dependencies between agents. Nevertheless it seems to be quite obvious that a terminology management system, which is an integral part of some other tool being used (machine translation system, translation memory system), is to be preferred to others which are not. The same applies to alignment tools or even translation memory systems. TransRouter distinguishes three levels of integration (built in, add on, compatible output). Human agents or service providers on the other hand are more experienced in the use of some tools compared to others. These dependencies are represented in the agent's profiles and will be used in the construction of routes as follows:

- If the main agent is human, TransRouter will prefer tools that are familiar to the translator. Furthermore it will prefer tools which are able to mutually cooperate. The level of integration will be considered only if there is no severe lack of performance compared to some other tools.
- If the main agent is a tool – e.g. a machine translation system – it is requested that all tools assigned to the main translation step allow some integration with the main agent.

### *6.5. Manual modification of routes*

The system, as already has been mentioned, will not necessarily find the optimal route. But even an optimal route could be of little use, if the agents of the route were occupied with other projects. In this case there is a need for the translation manager to modify routes suggested by

TransRouter manually. He may delete entire routes or copy routes to try out different versions of the same general approach.
Possible modifications of a route include the replacement of the main agent of a step, the removal of tools or resources from a step or the assignment of additional or alternative ones. This process is governed by a set of simple rules.

- The main agent may only be replaced by an agent of the same type. The route type will be unaffected. Changing the main agent will trigger a consistency check on tools and resources assigned to that step. It is checked whether the new agent may use them. If this is not the case the resource or tool will be replaced as well.
- If an agent is removed from a step, the corresponding resource will be removed too and vice versa. This will prevent the user from constructing inconsistent translation steps containing resources lacking an agent or useless tools.
- If a new resource is added to a step, TransRouter will remove an equivalent resource (same type) from the route if present. If the agent corresponding to the replaced resource is not able to process the new resource it will be replaced too. The optimal tool, which can process the needed data format, will be chosen automatically. An equivalent process will take place if an agent is replaced. Since all steps of a route should have the same equipment if possible, these exchange processes are performed on all steps simultaneously if the new agents or resources are valid for all of them. Otherwise the manipulation is restricted to the explicit manipulation of a single step.

### *6.6. Consistency check of routes*

Perhaps not all routes generated by the system are consistent. There may be essential resources lacking. Especially there might be a lack of translators who are able to use a specific tool. Thus, no agent may be found for a pre- or post-processing step. No further elaboration of an inconsistent route is possible. The system will offer the following options to deal with inconsistent routes.

- Marking of inconsistent routes.
- Providing information on the cause of inconsistency.
- Removing all inconsistent routes from the route set.

# 7. Route assessment

The step of route construction is followed by the assessment of routes. This step will be performed automatically. The user can guide this process only by adjusting the criteria which have to be used. Since the overall process of route assessment can not be understood without a deeper understanding of the criteria involved – time, cost and quality - these will be introduced at the beginning of this chapter. Then the effects of route features will be looked at in more detail.

Generally the assessment of routes can serve different purposes:

- Ranking of routes will help to find the best route with respect to a set of criteria.
- Estimation of time, cost or quality numbers will be helpful for the final planning processes. Setting a frame for time, cost and quality is an important task at the very start of a project.

In the course of the workflow supported by TransRouter the ranking of routes would be the first step. An exact estimation could be restricted to the best routes one should consider implementing. Even from the viewpoint of the designer of a decision support system this sequence seems to be reasonable. While the ranking of routes is a rather feasible task, the exact estimation of route features imposes severe methodological problems:

- Some of the relevant criteria – this is true especially for quality – are not well defined.
- The nature of translation processes is up to now not well understood. The effects of the environment – features of projects, agents and resources – on the translation process with respect to time, costs and quality can – in many cases – be quantified only by very rough approximations.
- There are aspects of the handling of a translation project, which are idiosyncratic to any translation agency.
- Some cost relevant issues can be discussed only on a larger scale than a single project. What is the benefit of a high quality translation memory? Which share of a software licence has to be charged for?
- A really exact estimation of cost, time and quality would require rather exact data about projects, agents and resources. It is questionable whether the result would justify the effort of data acquisition.

A reasonable approach to these issues in a project like TransRouter with its limited resources is to look for a compromise. The consequences for the Regensburg prototype of TransRouter are outlined in the following. There will be measures allowing the ranking of projects with respect to time and quality. Processing time can be computed in a way that is near to an exact estimate from an implementation point of view. The tuning of such a measure would need experience in the practical issues of translation projects, which is not present in a university environment. The same would apply to costs. But for the reason of resources within the TransRouter project the issue of costing will not be dealt within Regensburg in more detail than providing a general framework for cost computations. There is a qualitative measure for translation quality allowing no more than the ranking of routes and giving an indication of possible pitfalls in the course of the translation project. The extension of the TransRouter software to more exact measures would require less new implementation than more domain knowledge about translation. TransRouter will provide some documentation facilities allowing the user to gather such kind of knowledge while using TransRouter.

## *7.1. Translation time*

TransRouter makes use of several notions of translation time:

- Working time is used for cost computations. It measures the number of hours which have been spent on the project (2 people working 3 hours are delivering 6 hours work time).
- Raw elapsed time measures the time passing when the project team is working. This is an easy to compute measure for the efficiency of a route.
- True elapsed time recognises the fact that people work no longer than 8 hours a day and 5 days a week. An improved time estimator should take vacations or the risk of getting ill into account. Obviously the computation of true elapsed time differs with respect to the type of translation agent since the restrictions mentioned above do not apply to machine translation systems. True elapsed time is a measure to estimate the probable end date of a project.

### 7.1.1. Basic notion of translation time

The most basic understanding of translation time can be defined in a few sentences.

- The time needed to process a translation route is the sum of the processing time of all of its steps.
- The translation time of a route step is computed from the number of words of the text times the agent's (translator, reviser etc) translation speed as contained in his profile (measured in words per hour).

Obviously this formula is a crude abstraction because there is no single translation speed of a translator. Performance in translation is dependent on a number of factors the most important of which will be named here:

- The first important factor seems to be the language pair. A translator may be competent in several languages but the translation performance will vary.
- High translation speed will probably have a negative impact on quality. Thus translation speed will decrease in projects with higher quality requirements.
- Subject domain and text style will also be influential. The knowledge of a specialised vocabulary or of conventional rules of text structure and formulation might be necessary.
- If the text contains a great deal of new terminology, this will affect translation performance adversely.
- Certain text types, for example legal text containing quotations which must be quoted rather than translated, will have an adverse effect on translation performance (because of the time required to search out the quotations):mitigated of course, if an appropriate translation memory is available.
- Translation performance probably will depend on the readability or complexity of the text. TransRouter provides a tool for the estimation of text complexity. Since the notion of text complexity is not well understood up to now, this estimate can only be heuristic.

The influence of these factors seems to differ between individual translators. Thus, an exact translation performance measure would require the empirical acquisition of a huge matrix of interdependencies. Since this is not feasible a sufficient approximation must be found.

- The decision at which depth performance of translators or tools will be measured is left to the administrator of TransRouter (see next section).
- Any of the roles a human may have in a route (translator, reviser etc) will have its own performance profile.
- Performance will differ with respect to the required quality. Three quality levels are defined for translating and revising (browsing quality, information dissemination quality

and publication quality; more details will be provided in a chapter about quality issues). For machine translation, pre- and post-editing only two quality-related levels are distinguished: rapid-editing-translation and high-quality-editing translation.

### 7.1.2. Mapping of project features to translation performance

The mutual dependencies between translation performance and project features are of major importance for the time estimates of TransRouter. TransRouter uses an associative access method based on keys of variable length. This mechanism will be described on the basis of performance mapping as an example. The same mechanism will be used for other complex features – translation quality and translation costs – as well.

**Definition of absolute translation speed**

Each profile of a translator or machine translation system contains mappings from project features to translation performance values (quality, cost). The current implementation uses all relevant project features (source and target language, subject domain, text style and complexity, formats etc.) for keyed performance access. Additionally information about the translation route may be used (tools being used, features of resources being used). To avoid data acquisition overhead, a partial definition of access keys is possible. A fully unspecified key will retrieve a default value from the system. An access key may contain the following 'wildcards' instead of true project or resource data:

- '*': Matches any value. This is useful especially if a default value is to be defined that is valid for any project constellation.
- 'some': Matches any value other than the empty object. This is useful for instance if a default value for pre-editing for machine translation is to be defined. In this case at least some machine translation system must be present within the route.
- 'none': Matches if only the empty object is present. This is useful if the use of a specific system or agent type should be excluded.

Using this specification method the following phenomena can be expressed easily:

- A machine translation system can handle the following six language pairs at an average performance with given quality. The language pairs will be defined in the profile. No specific performance keys will be used. System performance will be defined as default.
- The system will translate English to German at a higher speed. A specific key (*source:* English, *target:* German) concerning this language pair will be entered.
- If an English text about agriculture is translated into German and if it has a specific format, the quality will be extremely good. An even more specific key (*source:* English *target:* German *domain:* agriculture *style:* annual-report) has to be defined.
- A translator is responsible only for some very specific cases (e.g. scientific reports about biology). A specific key covers the respective translation performance. The default translation performance will be set to zero, thus prohibiting the assignment of other projects.

To access performance data for some project or route the following steps will be performed:

- Derive an access key from the project's (route's) features.
- Sort the access keys of the profile according to the number of features specified in descending order.
- Select the keys, which subsume the access-key for the chosen project or route. Each feature of these keys has either the same value as the project's (route's) access key or no entry at all.
- Map the selected keys to their performance values.

- Compute a single value from the selected ones. In the case of translation performance and translation quality this means using the smallest value. Performance values thus define upper bounds, which are valid for specific situations.

**Definition of factors affecting translation performance, attached procedures**

Obviously this matching algorithm, which prefers the most specific access key, does not allow the use of general rules. More general effects are not described by absolute values but by numeric factors or even attached procedures. TransRouter will use a specific key structure (subject domain, text structure, text complexity) to access these data using the following algorithm:

- Derive an access key from the project's features.
- Find all keys matching the search key.
- Compute the product of all factors, which are associated to these access keys.
- Compute the product of the resulting cumulative factor and the performance value that was already found using a specific key.
- The code fragments, which were found, are sorted according to an inherent precedence value and then arranged as a pipe. The performance value found so far is used as input to this pipe. The output of the pipe is the final performance value, which will be further used for time and quality computations. The attached procedures accept three parameters: 1st is the translation step currently being elaborated (or nil), 2nd is the project profile and 3rd is the translation performance value that has been computed so far.

The expressive power of TransRouter now is extended to phenomena like the following:

- A translator translates scientific texts by an excess of 30% of average translation time. Assign a factor of 1.3 to scientific texts.
- The use of a specific tool increases productivity by 15%. Assign a factor of 1.15 for the use of this tool.
- Do not even consider using a specific machine translation system to translate texts of legislation. Assign a performance factor of 0 to a general key (*domain:* legislation).
- The translation performance will not exceed a certain threshold if the text is very complex. In this case a code fragment will check the threshold.
- The performance of teams is computed by some algorithm, which is defined as default. As a consequence this algorithm can be modified easily according to the specific needs of an organisation.

**Defaults**

A final extension of the data model allows the definition of default parameters for agent classes. A default performance profile for each type of agent concerned with translation (translators, service providers, and machine translation systems) and the related activities (revising, pre- and post-editing) is available. In a commercial environment a system like TransRouter would probably be delivered with agent profiles (except translators) and defaults being set.

Default profiles have the same structure as those of individual agents. Thus, the same phenomena can be expressed. Nevertheless, default profiles will contain only a few absolute values but most of the general factors and attached procedures of the system. Individual profiles on the other hand will contain absolute values, which will be modified by factors or procedures derived from the default profiles.

Default profile and individual profile will be merged on access time. Each key and value pair of the default will be moved to the individual profile if and only if there is not a similar key

already in existence in the individual profile. Thus, definitions in individual profiles take precedence over those of default profiles.

**Preinstalled defaults**
Some defaults concerning attached procedures will be defined automatically by the system at installation time. They may be modified later on according to the specific need of an organisation or user.

- Team performance: If a translator is member of a team, his translation performance will be reduced by a certain amount to cover organisational overhead.
- The performance of a post-editor depends on the difference between the quality value of the main translation step and the degree of quality expected from the project.

**Stepwise refinement of access keys**

The flexible size of access keys does not only allow the choice of an adequate level of detail in the definition of data but also the stepwise refinement of access within the decision process. In the beginning only project data are available. Later on additional information about possible routes and their resource assignment is at hand. Especially knowledge about the tools being used within a translation step can influence the translation performance and therefore will be included into the key structure. As a consequence TransRouter's estimate of translation performance (and quality) will be improved when the user enters additional information – especially about routes.

### 7.1.3. Time effects of resource use

The use of resources and their associated tools will affect the translation time and quality. In this section only the effects on the translation time will be considered.

- Use of a translation memory will reduce the text to be translated by an amount which can be derived from the coverage values provided by the resource assessment (obtained from the translation memory coverage checker). The access to a translation memory via a special tool on the other hand will cost some small amount of time, which can be computed from the tool's performance (contained in the profile) and the coverage values.
- The effects of unknown terminology can be disastrous especially in the case of machine translation. The coding of unknown words then is crucial for the success of a project. The time effect is computed from the efficiency of the terminology management system (or machine translation system) being used and the number of unknown terms provided by the resource assessment (originating from the unknown word detector).

The efficiency of resource use primarily depends on the efficiency of the tools used to access these resources. The experience and convenience a translator has in using these tools have a major impact on the performance too. Thus, the translator profile offers the opportunity to specify preferences in tool use and their effect on performance (if quantification is possible).

## *7.2. Translation quality*

Quality is one of the most problematic concepts within translation evaluation. There is neither a clear definition of the concept of text or translation quality nor a sufficient understanding of the interrelation between the translation process and its outcomes.

Within a conventional production process quality is described as the probability of a product having the required features. This requires a clear definition of these features which is usually given in a product description (design, modes of operation etc.). Additionally a sufficient number of similar objects must be produced in order to be able to compute probabilities.
Neither of these conditions is fulfilled in the case of text translation. Every translation is a very individual product presumably not allowing the estimation of fault probabilities. Most of the quality criteria that can be agreed on can not be formalised in a way that a decision support system would require.

- **Orthography:** Proper spelling is one of the most basic quality features of texts. Spelling errors can be found easily. Since a text contains long sequences of words even some kind of probability estimation should be possible. Unfortunately proper spelling is not one of the major quality problems even in machine translation.
- **Grammar:** Even a text of low quality usually can be expected to be grammatically sound. Text resulting from machine translation nevertheless may be ungrammatical especially if complex grammatical structures are concerned. Some grammatical phenomena causing problems for all or some individual machine translation systems may be exactly specified. If these structures could be identified within texts, their effect on translation quality could be foreseen. This diagnostic feature would require the grammar model of TransRouter to be more comprehensive than those used by the machine translation systems. Obviously it would be more reasonable to spend this implementation effort on machine translation than on a decision support system. As a consequence TransRouter employs a rather simple and heuristic model of text complexity that is sufficient only to give very general hints on quality effects.
- **Style:** The translator needs knowledge to meet style requirements imposed on some specific text type. Machine translation systems can to a certain extent be adjusted to such requirements too. Since style is a rather fuzzy criterion that is subject to individual taste TransRouter has to stick once more to rather general remarks.
- **Semantic fidelity:** The reader of a translation expects it to have the same meaning and to contain the same facts than the original text. Once again domain knowledge is a prerequisite of correct translation. The effects of a lack of domain knowledge nevertheless can not be quantified.
- **Readability:** The translation of a text should not impose additional obstacles to the reader. This criterion comprises most of the criteria mentioned above since bad spelling, grammar or style will negatively effect the readability of a text. Readability does not allow for a specific diagnosis of quality effects but may be measured by user tests.

The lack of an exact quality measure is a common problem for service providers. One solution of this problem is to define quality not primarily as a feature of a product but as a feature of the process of product construction or service delivery. From this point of view a tool like TransRouter is a major means of quality management since TransRouter will show possible translation routes and name the quality effects that can be expected. This is a major step forward even if a general quantitative model of quality can not be provided.
Nevertheless a quantification of quality within the decision support process is possible. But the resulting quality value can not be understood as some kind of error probability or a prognosis of good text style or text readability. If TransRouter assigns a high quality value to a specific route, this only means that the processes, which are part of this route, are designed to achieve this high degree of quality. Routes following this design will in general have a good quality. An individual instantiation of this route type may nevertheless produce poor quality as a result of effects, which are not under the control of TransRouter.

TransRouter will use the closed interval from 0 to 1 for the representation of quality values. The results are communicated to the user by four symbolic values, which are mapped to intervals of fixed size within the interval 0-1:

- No use: The translation will probably be in such a bad shape that it can not be used at all.
- Browsing quality: The reader will be able to identify what the text is about.
- Information dissemination quality: The reader will identify the arguments and major facts of the text.
- Publication quality: The translation fully meets the quality standards of the original version with respect to content and form.

### 7.2.1. General quality effects of resource use

The resources being used have a significant effect on the translation quality that can be achieved within a translation step. Negative effects however can to a certain extent be compensated for by the translator or a reviser / post-editor.

- Resources need an approval by a translation manager. For high quality projects this is a cut off criterion. In all other cases a warning will be given.
- The quality of a resource has a major impact on the quality of a translation. Since fragments taken from a translation memory or terms from a term-bank may be scattered all over the translated text, the resulting quality value may not exceed that of the resource. If a translation memory has comparatively many full matches an average of the qualities of new and stored translations may be computed.
- If the dictionaries and term-banks used by a machine translation system do not fully cover the vocabulary of a text, this will have severe consequences on the resulting translation, which depend on the translation competence of the system.

### 7.2.2. Quality effects of translation agents

The quality of a translation will primarily result from the competence of the translator or translation system. Thus, TransRouter allows the definition of a detailed quality profile for translation agents. A quality profile generally is a mapping from project data (languages, text complexity, subject domain, text style etc) to quality values. The access method is the same as that introduced in the context of the keyed access to translation performance data. Similarly quality effects can be described using absolute values, factors or algorithms. All relevant project and route features – language pair, domain, text style, TM coverage, unknown terminology etc. - are covered.

As has already been mentioned quality values assigned to agents do not allow for the prognosis of any features of the resulting text. They are means of agent selection in the context of organisational rules of quality management. These procedures however may be learned from experiences resulting from translation tests or real translation projects.

### 7.2.3. A notion of quality within the translation process

Not only resource and agent features have influence on the quality of a translation but also the specific translation procedures which are chosen and represented as route descriptions. In the following the effects of the implementation of the translation steps in a concrete route will be described:

- **Pre-edit step:** Pre-editing is of major consequence only for machine translation. Pre-editing comprises the acquisition of new terminology and the marking of proper names and other passages that shall not be translated. If the effort of terminology acquisition is

not spent, a major lack of quality will result. The consequence is a reduced quality that can be expected from the main translation step. The degree of quality loss is influenced by the percentage of text affected by the existence of unknown terminology and by the quality profile of the machine translation system.

- **Translation step:** The quality of the main translation step primarily depends on the desired quality level as defined in the project profile, the quality profile of its main agent and the quality values of the resources used. As already has been mentioned there may be quality restrictions resulting from the pre-edit step.
- **Post-edit step:** The quality of this step also defines the overall quality of the translation route. The quality will not exceed the quality value from the agent's quality profile and that one found in the project profile. Some shortcomings of the main translation step can be compensated for in the post-edit step, but the increase of quality may not exceed one level (from browsing to information dissemination or from information dissemination to publication quality). The quality gap that has to be bridged by the post-edit step also has an influence on the performance, which can be expected from the step's agent (see above).

### *7.3. Translation costs*

A true cost calculation is out of the scope of TransRouter as a tool for decision support in translation routing. Most of the fixed costs are not relevant in this case since they apply to any of the routes in question and thus do not influence the decision. Exceptions would be the decision about hiring a new translator, contracting a service provider or buying a license for some translation tool. But even in this case the costs of a purchase can not be attributed to a single project, since the acquired tool will be used in more than one project.
Therefore it is assumed that this global cost assessment is performed outside TransRouter. The results of this calculation are costing figures describing the costs of using some tool or employing a translator. These figures may be provided as costs per hour or as costs per word. They may be assigned individually or collectively (as defaults). In some cases not even default values may be available. In this case no cost calculation is possible.
To be able to assign specific cost tables to specific project constellations, the cost tables in TransRouter are organised in the same flexible way as the performance and quality tables.

# 8. Project documentation and learning from data

Since TransRouter is a tool for supporting a translation manager in the appropriate choice of a route for a translation project any issues concerning the project management and documentation seem to be outside the scope of this project. This is only true from the point of view of a single project. The decisions taken in a project and the outcome of these decisions however constitute valuable information for a new project if the project features are comparable in some way.
This is especially true as most of the data initially fed into TransRouter can only by approximations or even guesses. If the projects that are processed using TransRouter are documented, the translation manager can get a notion of the quality and value of the decisions taken by TransRouter.
Thus, TransRouter will ask the translation manager for the following information – if not already present - on every project and put it into the archive:

- The project profile
- The route that is finally chosen within the decision process, including the agent, tool and resource assignments and the calculated time, cost and quality values
- The route as it is implemented within the course of the translation project, including the agents, tools and resources that were finally assigned to the project and the time that was needed for each translation step and the quality which was achieved.

On the basis of this information TransRouter can assess its own performance and calculate the averages of the deviation between predicted and true values for all projects and – more decisive – for all projects similar to a new project which is to be tackled by TransRouter. Furthermore all tools or resources may be identified, which although chosen by TransRouter on the basis of their profiles are often skipped by the managers in the real implementation of the project. These simple but useful features are already implemented in the TransRouter prototype.
The next step in the evolution of TransRouter, which cannot be taken within the scope of this project, would be the learning of agent features from real world data. Every translation project represents a new case from which system parameters can be learned. The first step would be the acquisition of very specific access keys to performance and quality that represent the relevant project features. These keys will then be assigned to the agent profiles. If similar cases occur later on, the data may be adapted to get a best fit to all similar cases. Later on, when a fairly sized pool of cases is at hand, generalisation processes may be run on the profiles. They will isolate those project and route features, which contribute significantly to the project's outcome and skip those that don't. Thus, the general predictive quality of the system will be gradually enhanced.

# 9. The user interface of TransRouter

This chapter will provide a general model of the user interface and a more detailed description of the major TransRouter interface tools. The object model of TransRouter makes use of software reuse (inheritance and use of specifications and methods) as much as possible. As a consequence several TransRouter tools share layout specifications. The description of these tool features will be given only once and on the occasion of the first occurrence.

## *9.1. Design of the user interface*

The description of the TransRouter interface covers the following aspects:

- window layout
- form filling within TransRouter

### 9.1.1. The window layout

TransRouter is a complex window application employing multiple overlapping windows, each belonging to some TransRouter tool. Each window is divided into a set of tiled subviews according to a uniform layout (see figure 1).

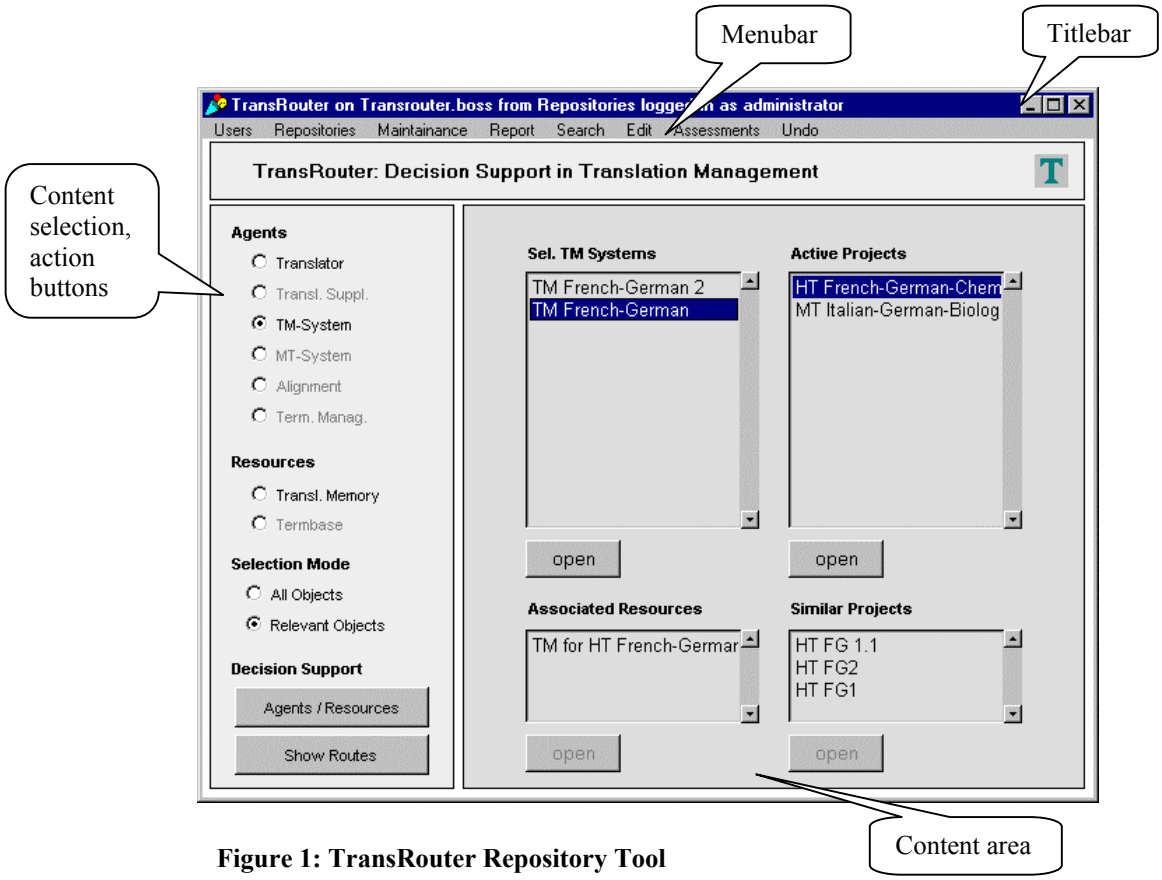



**Figure 1: TransRouter Repository Tool**

The TransRouter window as depicted in figure 1 contains the following sub-views:

- A title bar providing the name of the tool and additional meta-information
- A menu bar offering most of the tool‘s functions to the user.
- A content area, where tool data are presented to the user. Since most of the TransRouter objects are defined by a large number of complex features that can not be presented on a single page, the content area presents only a selected subset of data.
- The data to be presented in the content area are chosen from the content selection area. A couple of radio buttons allow the selection of specific pages to be presented in the content area. In addition to the page selectors some of the most important tool functions, which must be accessible on all pages, are presented in this sub-view as action buttons, which allow a more convenient access than the menu bar.

### 9.1.2. Form filling with TransRouter

The dominant interaction mode of TransRouter is form filling. Most of the relevant features of projects, agents and resources are defined by filling numeric or textual input fields or by selecting from menus or lists. Where predefined identifiers are available – languages, quality levels, currencies etc. – these may be selected from a pop up menu. If an appropriate identifier is not at hand (some new language to be translated), a new identifier may be entered manually.
An alternative interaction mode to menu selection is drag and drop. Data items may be selected from lists and be moved to any input field accepting the respective data type.
Since the data format required in some input fields is not obvious (e.g. the accepted date formats), a fly-out string will provide a short explanation, when an input field becomes active (touched by the mouse). Since this feature can be very annoying to experienced users, it can be turned off.

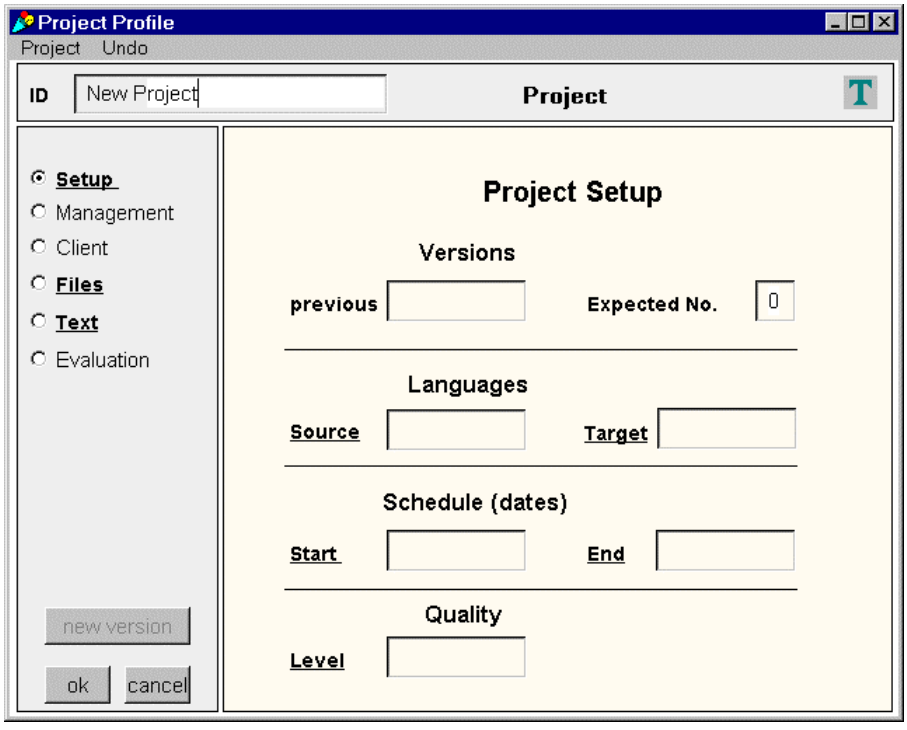



**Figure 2: Project tool – enhancing necessary data entries**

TransRouter asks for a significant amount of data from the user. Some of these data are not needed for the decision process but only for documentation purposes. Examples are the client’s address or name of the project manager. The user may decide to skip some of these

data. TransRouter shows page identifiers and field labels of necessary data, which are still missing, in an enhanced display mode, in order to help the user to make a difference between optional and necessary data entries.
TransRouter minimises the effects of erroneous data manipulations. Any sequence of user actions can be undone. Any tool can be terminated by a cancel operation, leaving the manipulated data unchanged.
A locking mechanism guarantees that no objects may be manipulated in parallel from several tools. This feature is a first step towards a net-based use of TransRouter.

### *9.2. Repository access*

The TransRouter Repository Tool is the entrance to TransRouter. It provides access to almost all objects that are contained within the repository of TransRouter.

When initially started the tool will be in guest mode. Only a restricted set of data will be accessible. After the user has made himself known to the system an enhanced data set will be displayed according to his access permissions.

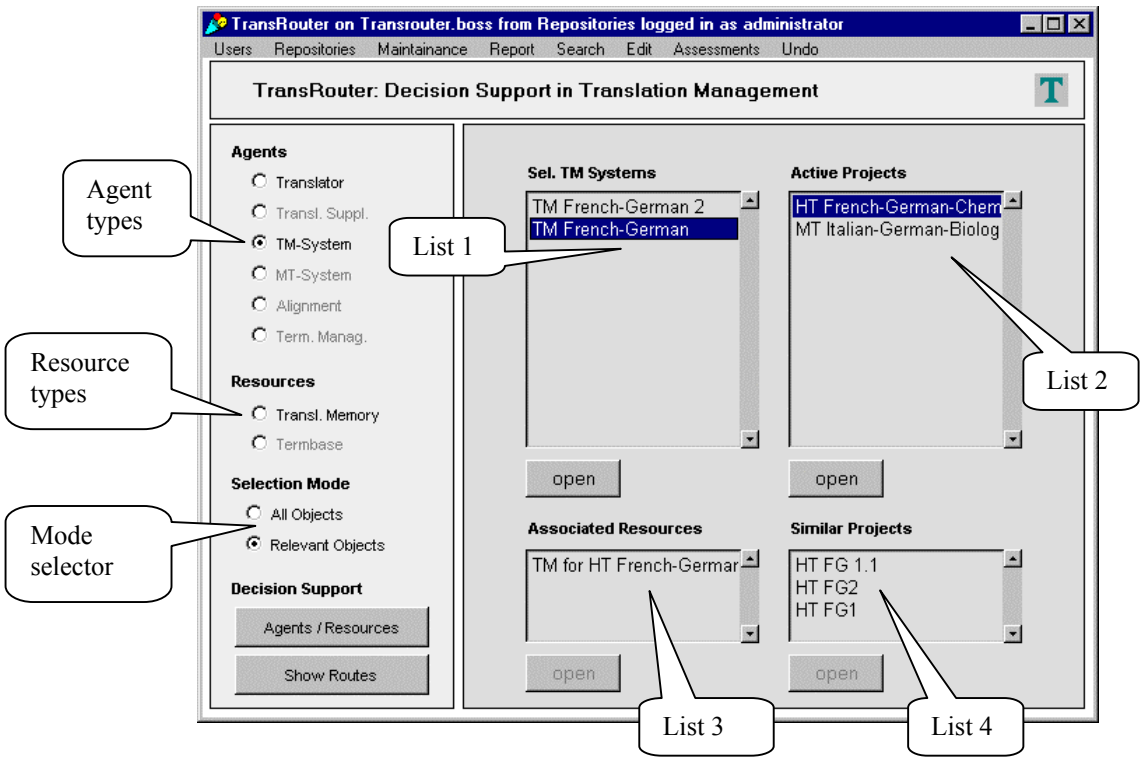



**Figure 3: TransRouter Repository Tool**

#### Accessing agents and resources

The user may select a type of agent or resource from the content selection area of the tool. All objects of this type will then be displayed within the upper left view of the content area (list 1, see figure 3). If, however, the mode selector is switched to 'Relevant Objects', only those agents or resources will be presented which are relevant for the currently selected project (list 2). If an agent is selected in list 1, list 3 will show all (relevant) resources that can be

processed by this agent. If a resource is selected in list 2, list 3 will show all agents that can process this resource.

Selecting an object from a list and then pushing the ‘open’ button will activate a tool for the presentation and manipulation of an agent and resource profile. Functions for the creation or deletion of agent and resource profiles can be selected from the ‘edit’ menu within the menubar. Searching for agent and resource profiles is possible by selecting an object type and then using a search function from the menubar.

If only the agents or resources that are relevant to a selected project are displayed, the creation and removal of objects will lead to a manipulation of the relevance set only. The selection of agents and resources that are relevant to a project is started by pushing the ‘agent / resource’ button.

**Accessing projects**

List 2 presents the identifiers of all active projects. Projects are considered as active until a true end date is defined and the project is put into the archive. If a project is selected a list of similar projects – active and completed ones – is shown in list 4. The user may look, how and with what outcome these projects where processed.
Manipulation, creation, removal of and search for projects are handled similarly to the approach taken for agents and resources.

Pushing the ‘Show Routes’ button leads to the computation and presentation of possible routes for the processing of a selected project.

### 9.3. *Project profile*

The project tool allows the definition and browsing of project profiles. Project information is presented on six distinct pages.

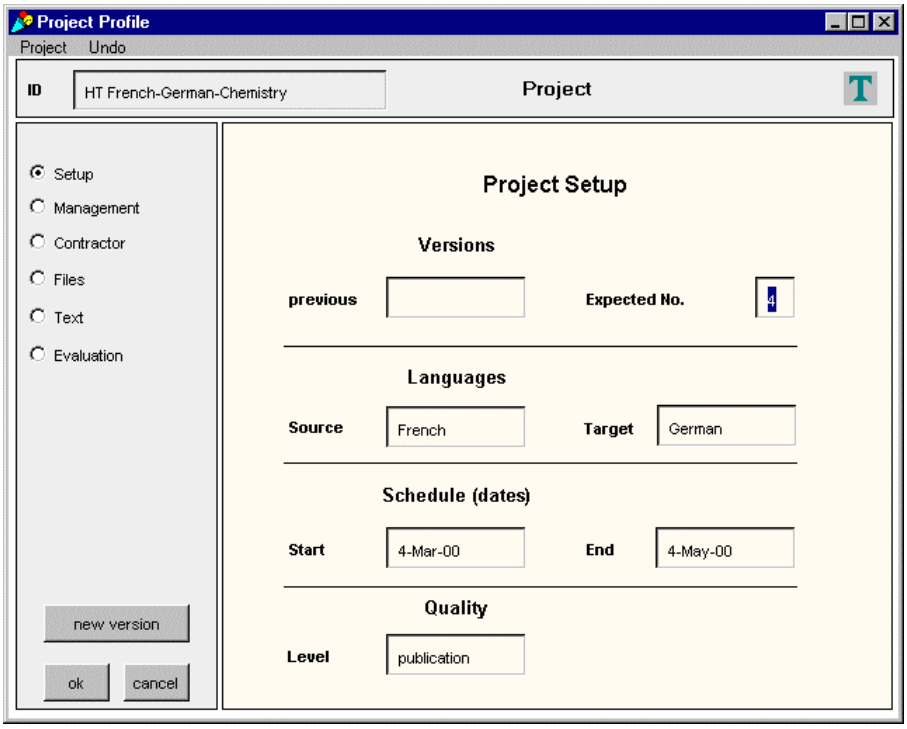



**Figure 4: Project tool – setup page**

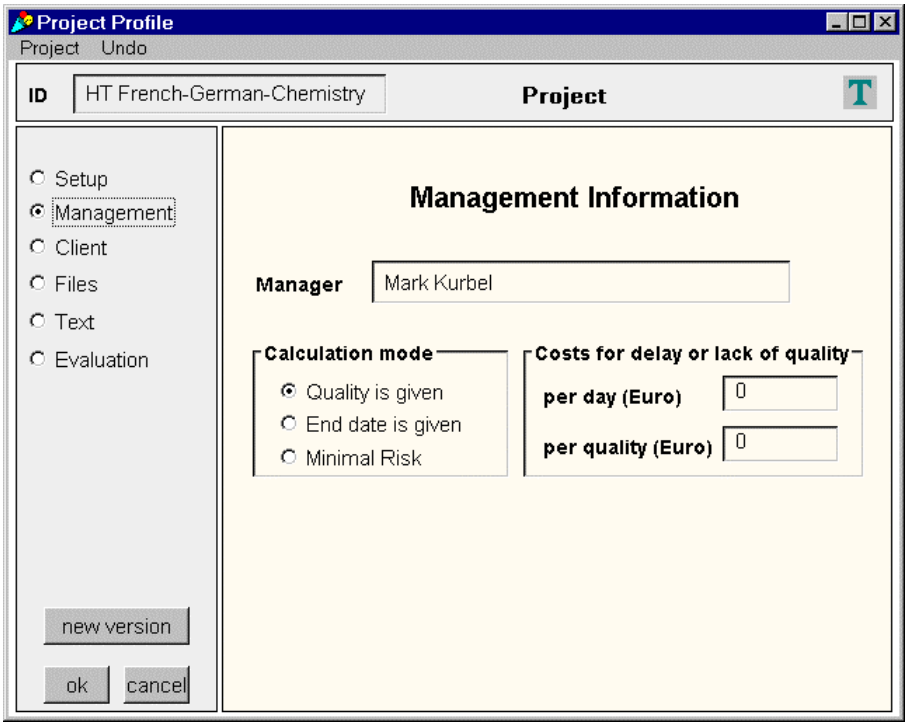



**Figure 5: Project tool – Management information**

The set-up page contains information about general aspects of the project: start and end date, source and target language, required quality. Predefined identifiers are available for the definition of languages and quality levels. If parts of the text have already been translated in previous versions of the project, the most recent project should be specified here by its identifier (or by dropping a proj??ect object). TransRouter will use this information for the purpose of resource lookup. The number of successive versions of the text will help to decide on the use of a translation memory.

The translation manager – named in the management page of the project profile – can choose between three approaches to route selection (see figure 5):

- Prefer routes resulting in a quality that is best fitting to the required quality as specified within the project profile.
- Prefer routes that meet the specified

end date of the translation project.

- Assess routes according to a risk function that will provide a unified view of the importance of quality and time.

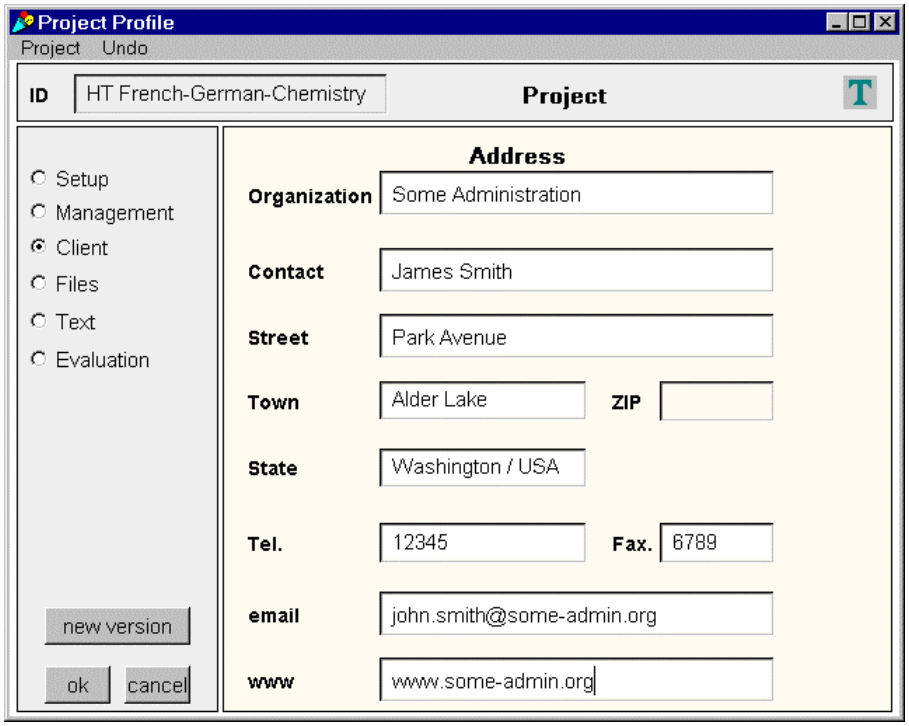



**Figure 6: Project tool – client address**

Most of the TransRouter profiles contain address information, which is accessible in a unified layout. The project profile contains the address of the client. Agent profiles will contain the addresses of a translator or the hotline of a software house etc (see figure 6). The address will cover all relevant contact information about a person or organisation.

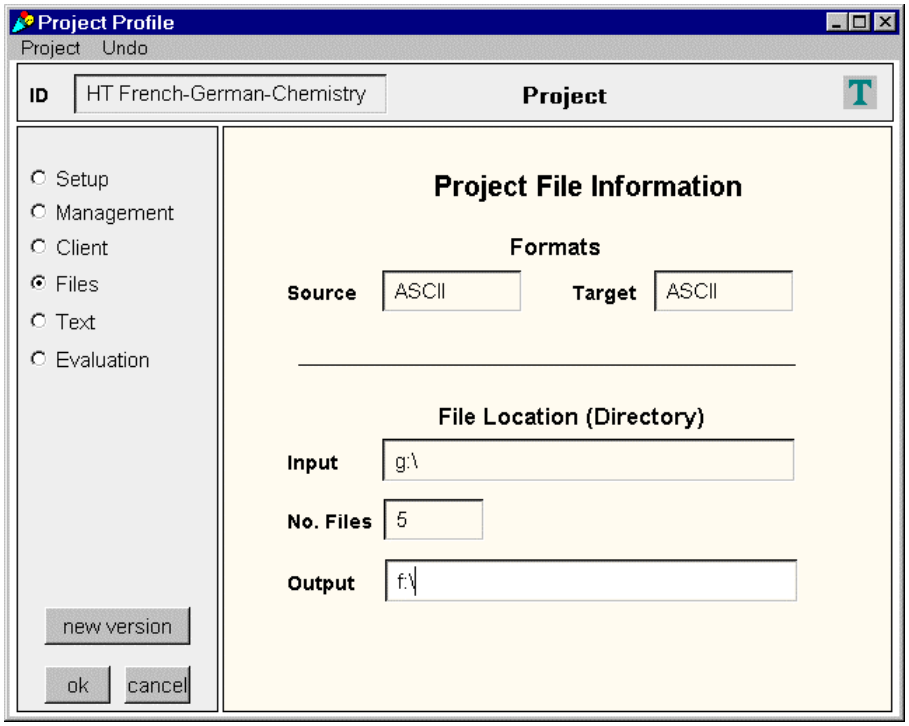



**Figure 7: Project tool – file information**

Information on the provided and required file formats will be used to decide on tools that can be used within the translation process (figure 7). The directory containing the text files of the translation project will be accessed by the TransRouter analysis tools in order to estimate text features and to assess resources.

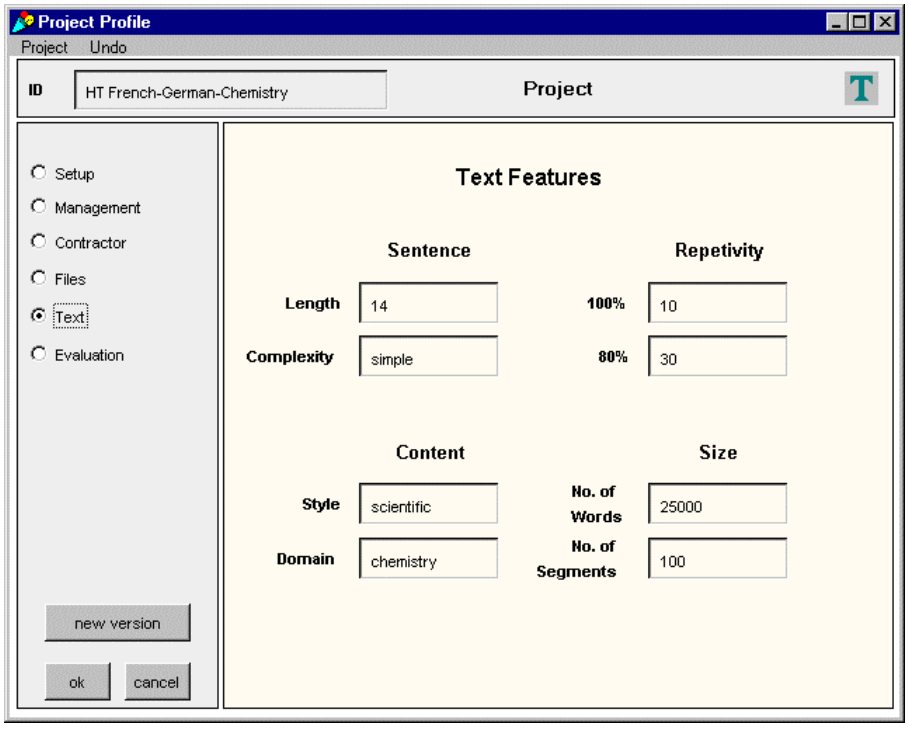



**Figure 8: Project tool – text features**

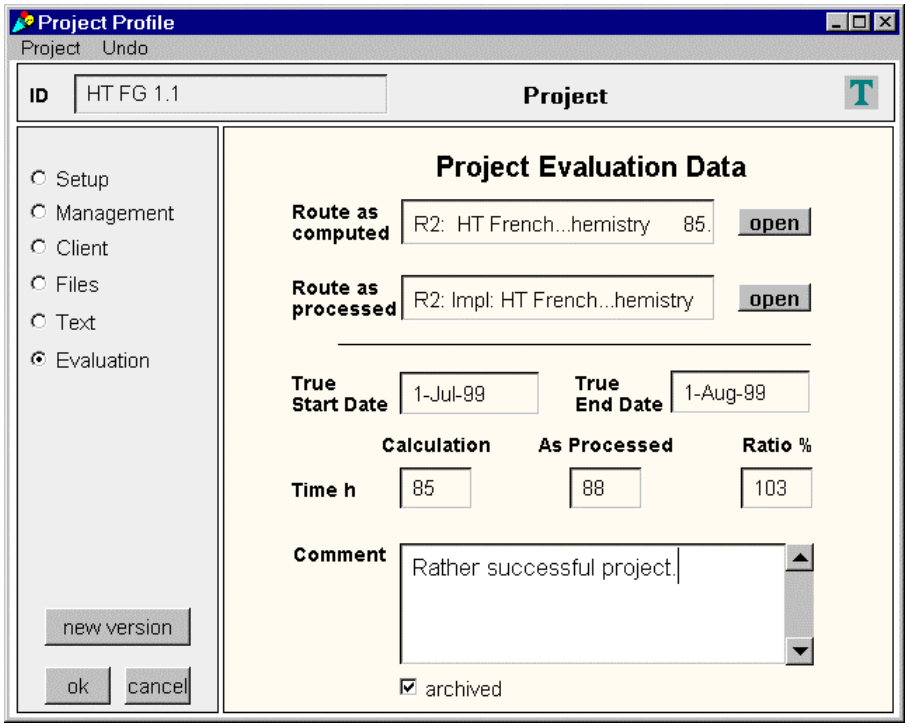



**Figure 9: Project tool - evaluation**

Text features (figure 8) play a major role in the selection of agents and resources. All of these features besides style and domain will (the LRC prototype already implements this feature) be determined automatically by TransRouter analysis tools but can also be entered manually.

Text features are of significant importance to decision making since they are part of the access key to translation performance and quality. The evaluation page of the project tool serves documentation purposes (figure 9). It allows access to the translation route as it was computed by TransRouter and selected by the translation manager. For comparison the route as it was finally processed in the course of the project is documented. Information about agents and resources employed and the time and cost values of any single translation step may then be entered. These data then will be used as a feedback to the system to allow the assessment of the prognostic quality of the system and – in some later version – to improve it.

## 9.4. Agent profiles

There are a couple of tools allowing the browsing and editing of agent profiles. They share several pages displaying common features. In the following, a complete description of the tool for machine translation systems will be provided first. All subsequent tool descriptions will contain only those pages that are specific for that individual tool.

### 9.4.1. Machine Translation System Tool

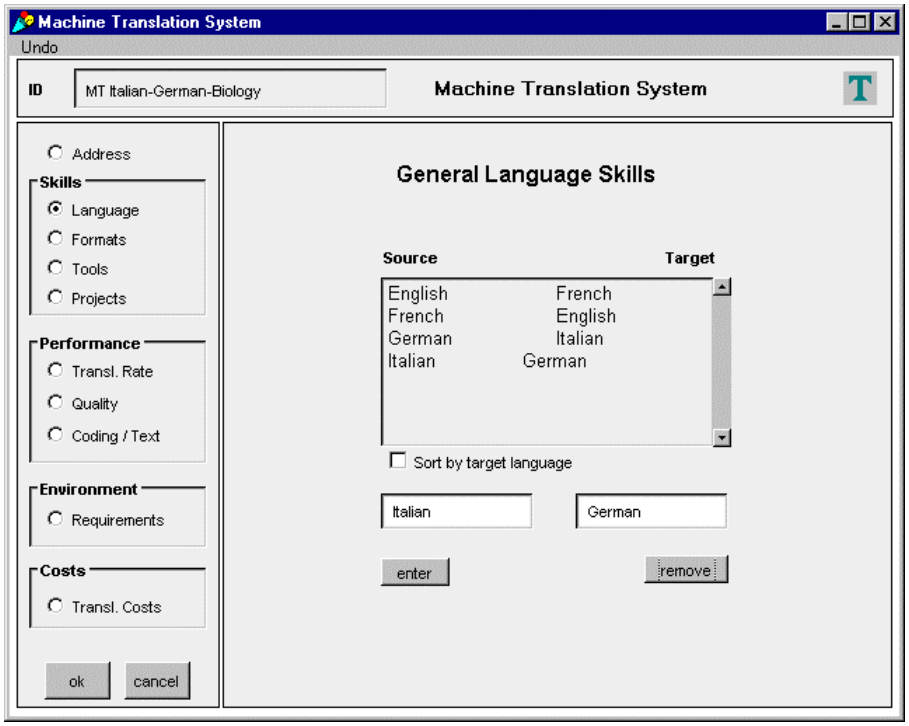



**Figure 10: Machine translation system tool – language skills**

Machine translation systems can translate a number of source languages into a number of target languages. Translation speed and quality may differ for each of these language pairs as a consequence of tuning activities or as a result of the quality of the resources at hand.

This page (see figure 10) shows general language skills. The agent is able to handle the named language pairs. For some tools – e.g. translation memory systems – no more information is necessary. For all agents performing translations – machine translation systems or human translators – additional information about performance and quality is needed.

All software tools require some minimal system resources or a specific environment. These data can be defined as shown in figure 11.

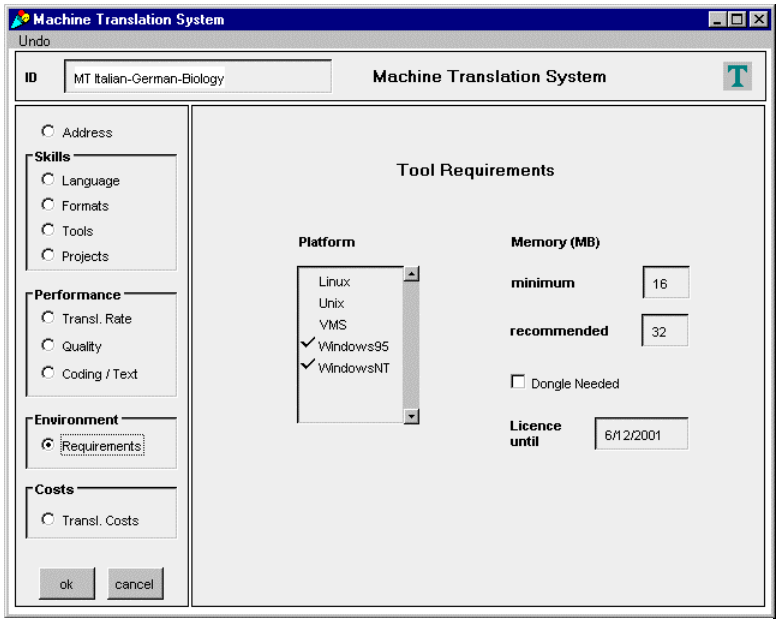
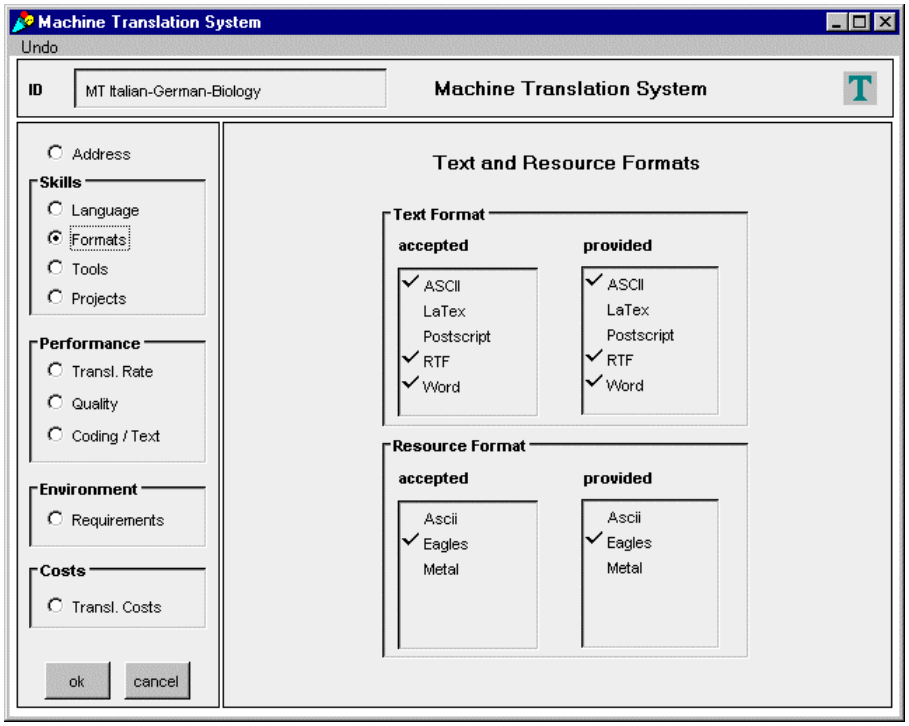
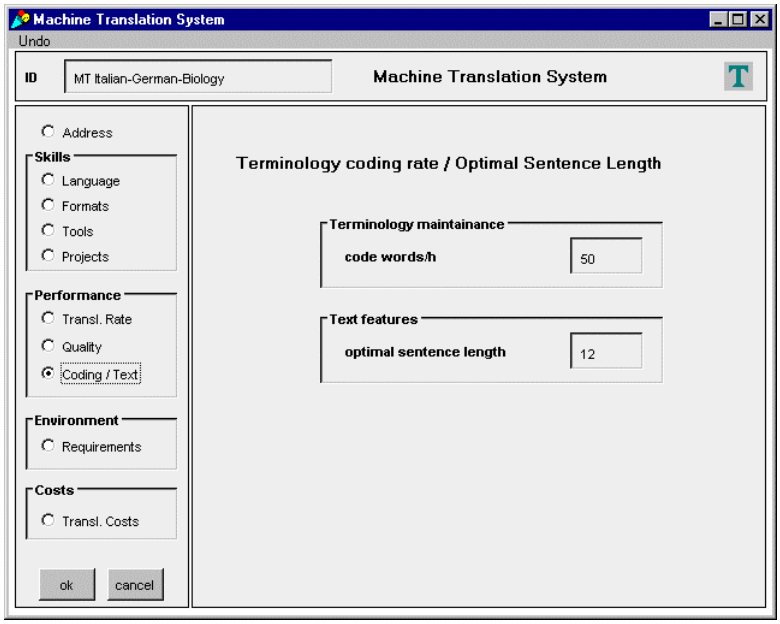



**Figure 11: Machine translation system tool - requirements**




All agents provide information about the text formats they are able to process and to produce (see figure 12). Some agent types depend on specific resources – translation memories or term banks etc. - that may be delivered in a special format.

**Figure 12: Machine translation system tool - formats**




Some more features are assembled on an additional page (see figure 13).

Machine translation systems offer the opportunity to enhance the terminology. The coding time is used within the overall time calculation.

Sentence length is rather influential on the translation quality. The optimal sentence length of a MT-system can be used as a means of route assessment.

**Figure 13: Mt system - miscellanea**

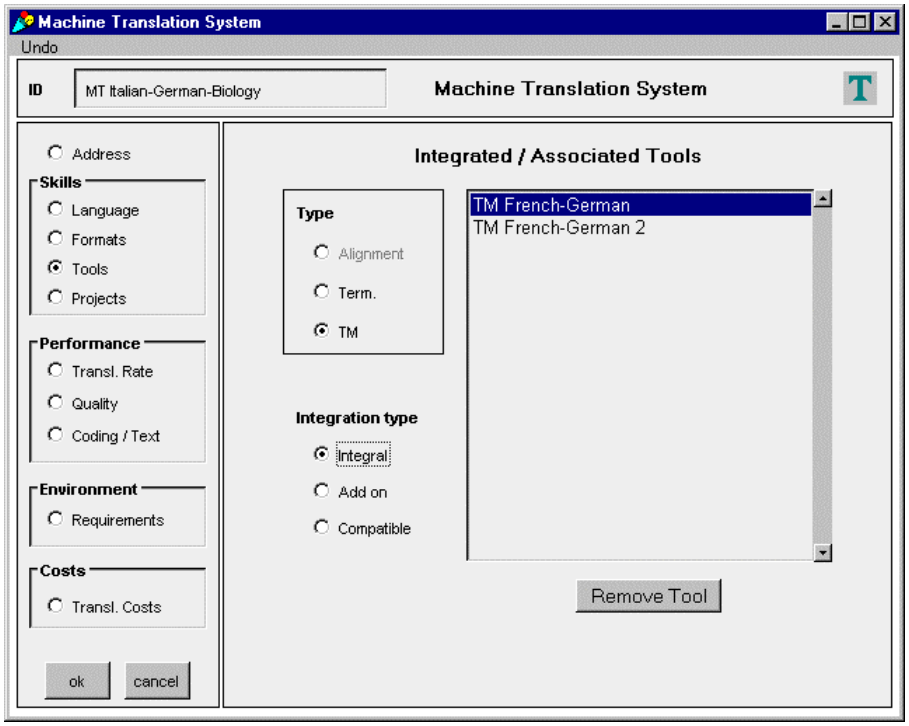



**Figure 14: Machine translation system tool – associated tools**

Some machine translation systems are able to integrate other translation tools like translation memories or terminology management system. The same feature applies to other system types as well (see figure 14). Similarly human translators are knowledgeable and experienced in the use of some systems or system types, while not knowing about others. The decision kernel will prefer combinations of translators and systems that are compatible.

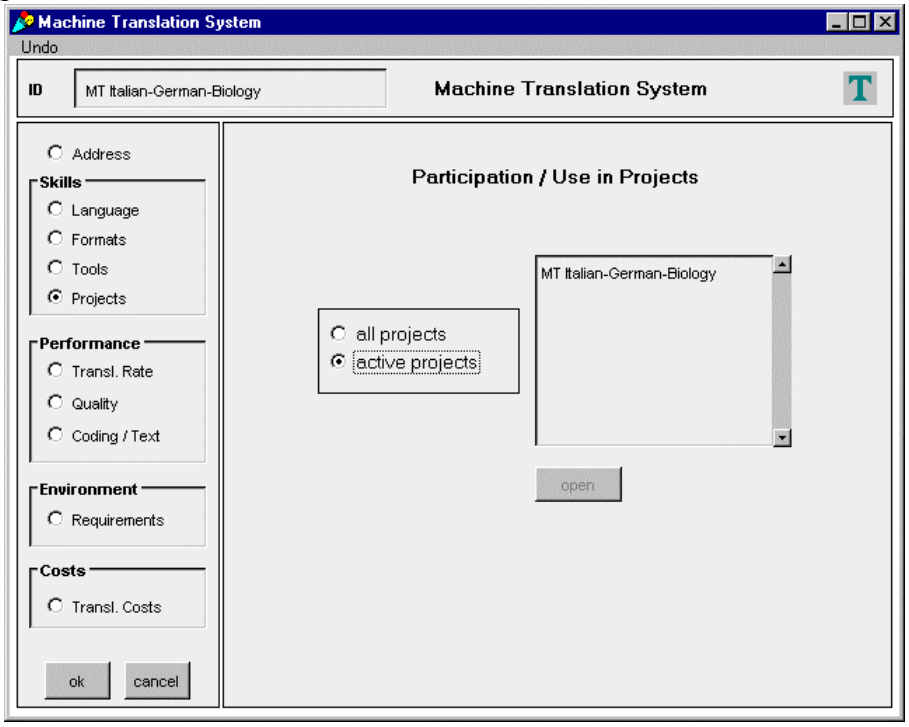



**Figure 15: Machine translation tool - projects**

All agents and resources carry information about the project they were used in (see figure 15). This information is extracted automatically from the descriptions of translation routes as they finally were processed. Accessing these projects and their documentation may help to estimate, whether the agent or resource participated successfully in projects similar to the project in question.

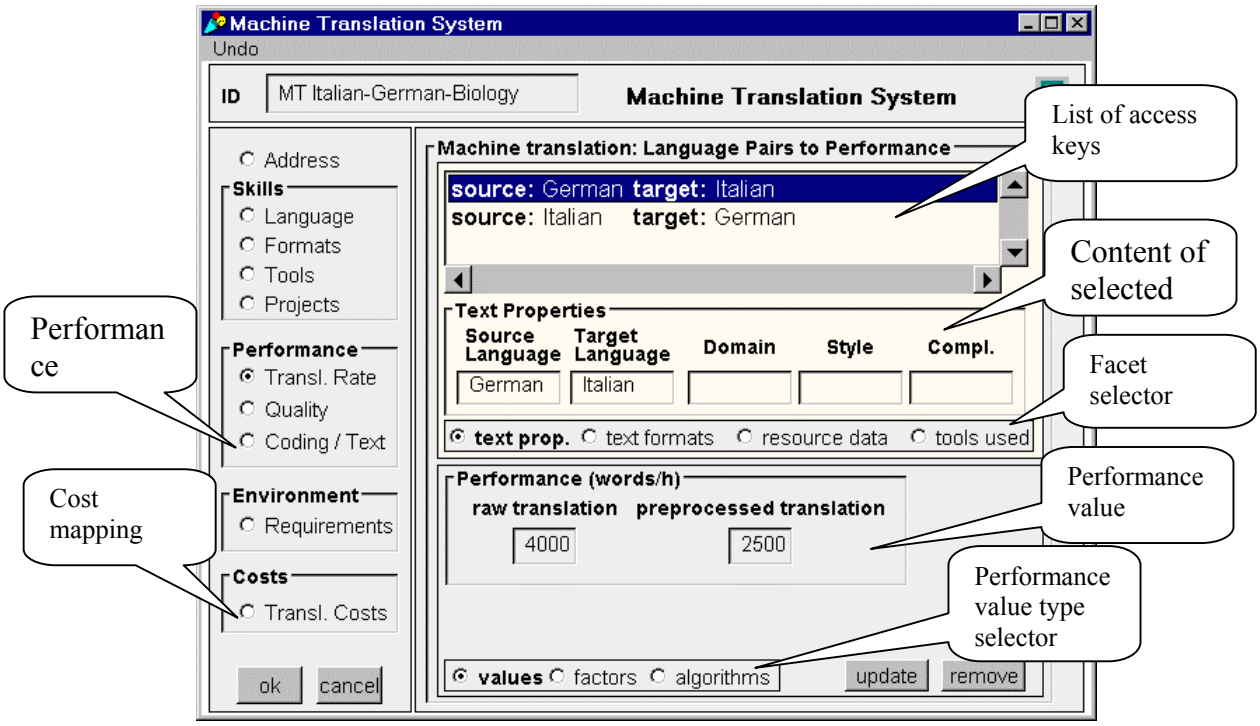



**Figure 16: Machine translation system tool – translation speed**

The performance of translation agents is defined by complex mappings from project data to performance values. This mechanism has been described earlier in this report. Figure 16 depicts the interface that allows the specification of the translation performance of machine translation systems. The only aspect that is specific for machine translation however is the mapped value. The overall mechanism is valid also for other features of machine translation and for other agent types as well.

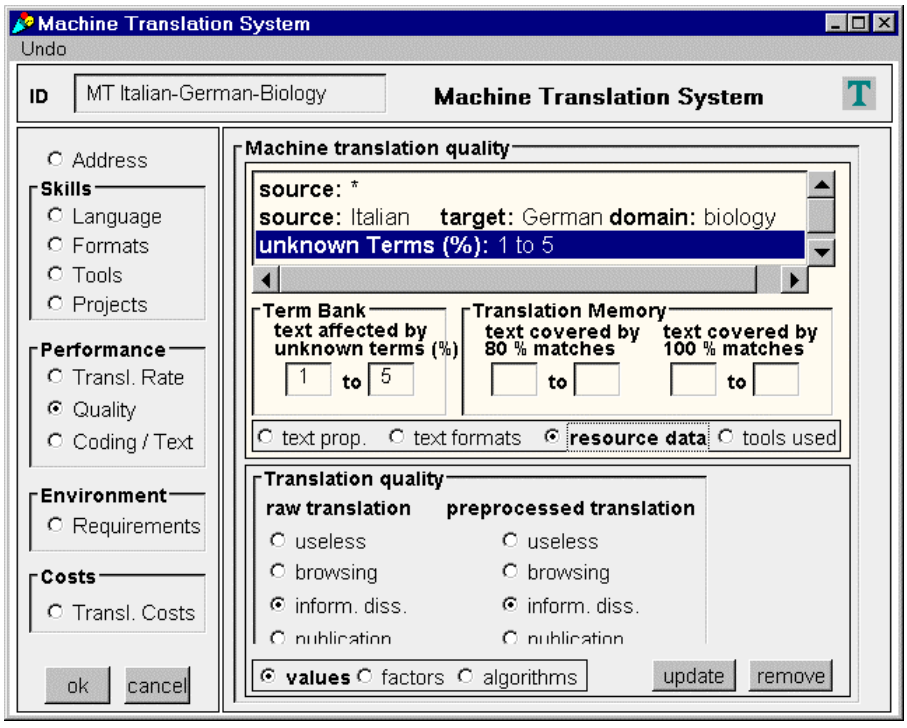



**Figure 17: machine translation system tool – translation quality**

- A list of access keys provides access to performance values. If a key is selected, a performance value will appear in the lower data field (*performance value*). Additionally

the facets and fields of the selected key become accessible in the upper data field (*content of selected key*).

- Each key is defined by a couple of data fields that are grouped to four facets (*text properties*, *text formats*, *resource data* and *tools used*). The facet that is to be displayed can be selected from a list of radio buttons (*facet selector*). A boldface character type indicates facets that contain data.
- There are three types of performance data values that may be assigned to an access key. Absolute values, factors and algorithms. The user can select the value type that is to be displayed using a list of radio buttons (*performance value type selector*). Bold face characters once more indicate the existence of instances of the respective type.

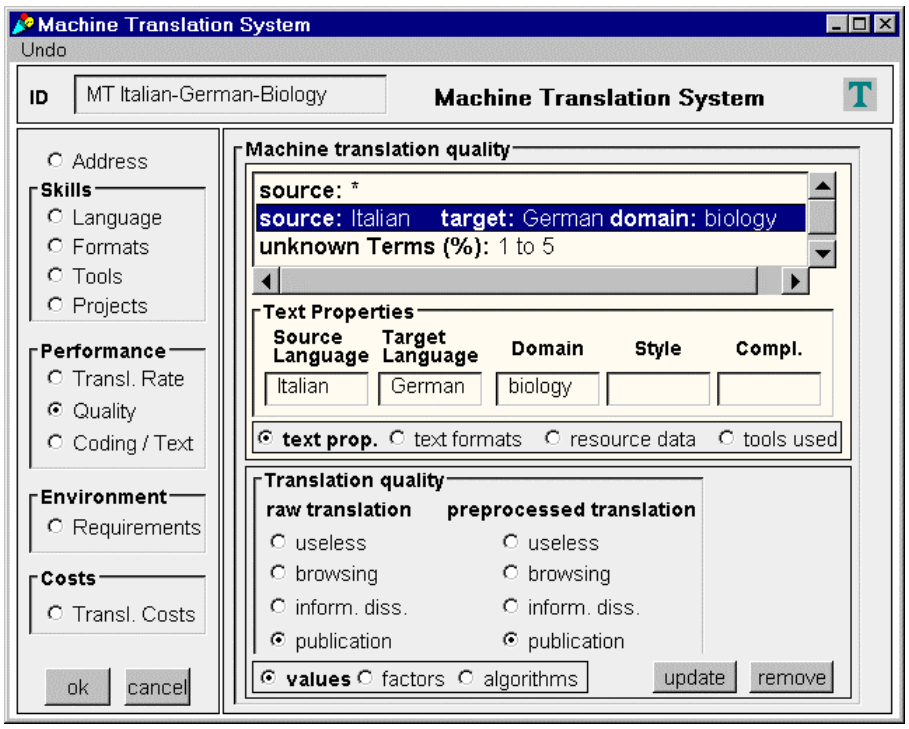



**Figure 18: Machine translation system tool – translation quality**

Figure 17 shows a performance mapping concerning not translation speed but translation quality. The keys contain data from an alternative facet (*resource data*). The performance data define quality values.
Figure 18 depicts an alternate quality value. Since the access key shown in the figures 17 and 18 stem from different facets, both values may be valid in a specific situation. In this case the minor value (*information dissemination*) will be chosen.

### 9.4.2. Translator Tool

Translators may be engaged in more activities than mere translation. Each of these activities in principle requires its own complete performance profile covering time, cost and quality. To avoid this modeling overhead some simplifications have to be accepted.

- An individual translator is assumed to guarantee the same quality level for all activities he is involved in.
- Costs of an activity are calculated by the time the activity consumes. Fees for software use may be added. There is no special costing for special activities.
- Some tasks are grouped into a complex activity.

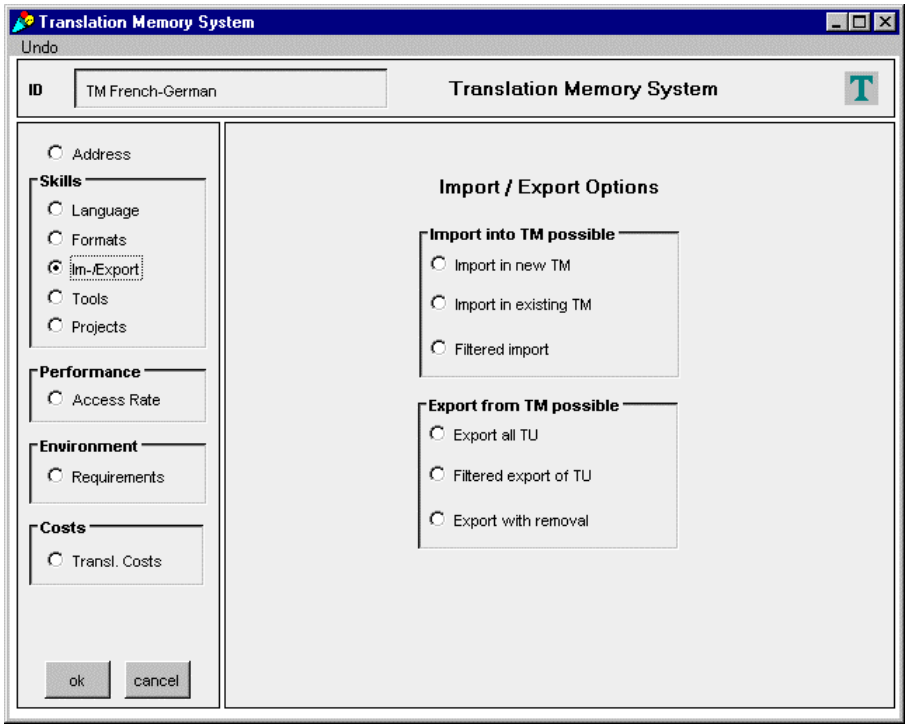
The following activities (besides translation) can be found in a translator‘s profile:

- Pre-edit: Prepare a text for translation by a translator or a machine translation system. This includes the acquisition of new terminology, spell checking and tagging. Since this task is

by far more time consuming for machine translation systems, a difference should be made within the tool facet of the performance access key.

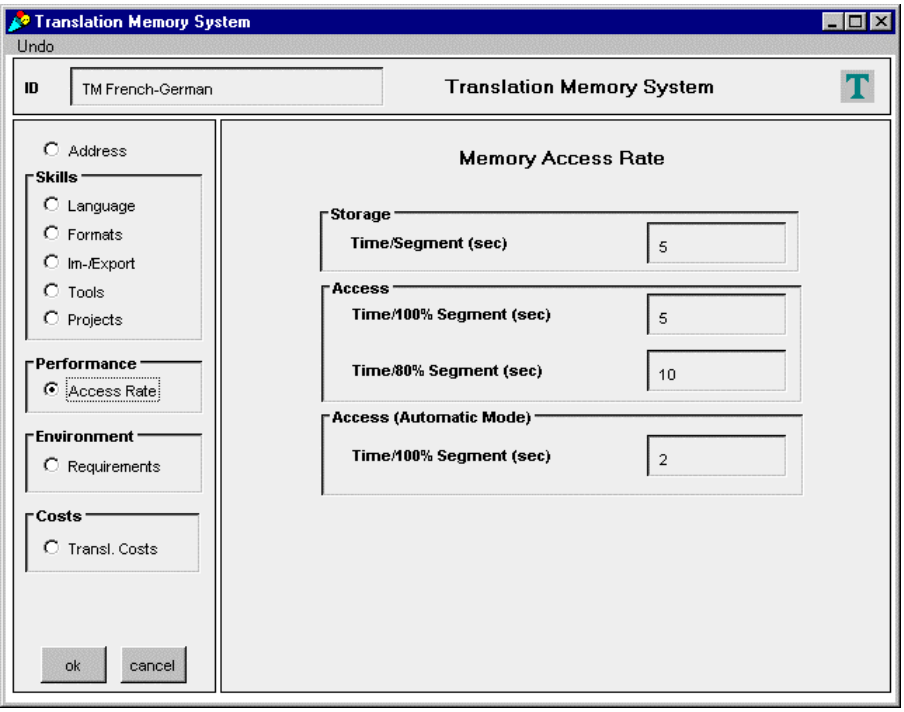
- Post-edit: Proofread the translated text resulting from machine translation.
- Revise: Revise the text produced by a human translator.

### 9.4.3. Translation Memory System Tool




Some translation memory systems allow the integration of a number of existing translation memories into a new one. If TransRouter finds more than one relevant translation memory such agents should be preferred.

**Figure 19: Translation memory system tool - options**




The efficiency of using a translation memory system is determined by the access rates of the tool as defined in the tool profile (see figure 20) and by the efficiency of the handling of the system as defined in the user's performance profile.

**Figure 20: Terminology management system tool**

### 9.4.4. Terminology Management System Tool

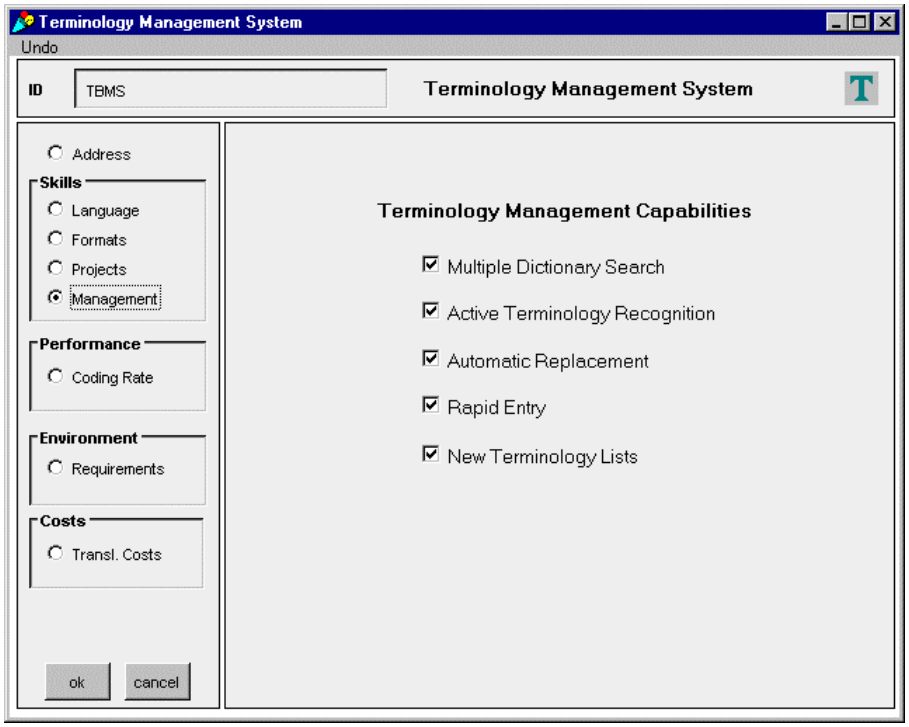



**Figure 21: Translation memory system tool – access rates**

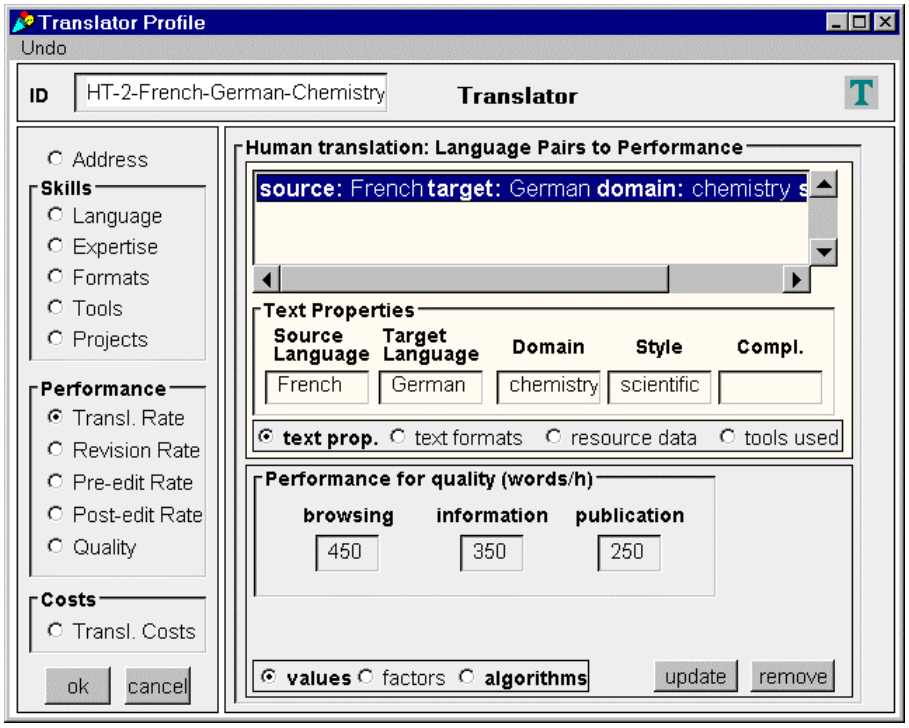


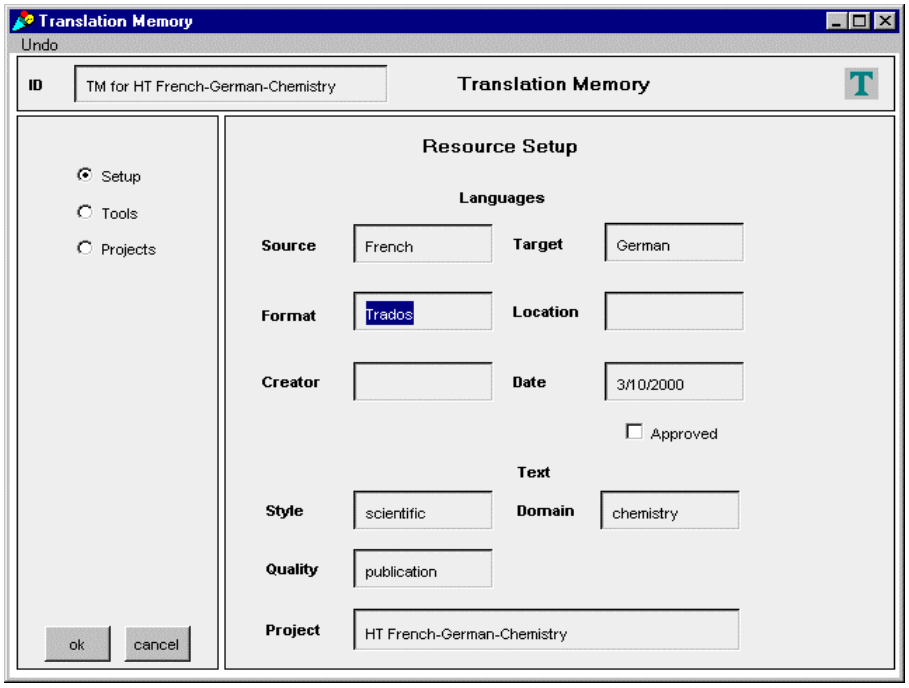

**Figure 22: Translator tool - performance**

The ease of use of a terminology management system will be influenced by features as listed in figure 21. Since these effects can-not easily be quantified, TransRouter uses only one feature actively for decision making. If more than one relevant dictionary is found, the management system should be able to do multiple dictionary search.

The major difference between the profiles of translators and translation tools is that translators can participate in a variety of tasks such as translation, text revision or pre- and post-editing (see figure 22). Each of these activities is described by its individual performance profile. The performance of a translator depends also on the quality level that is to be achieved.

## 9.5. *Resource profiles*

### 9.5.1. Translation Memory Tool

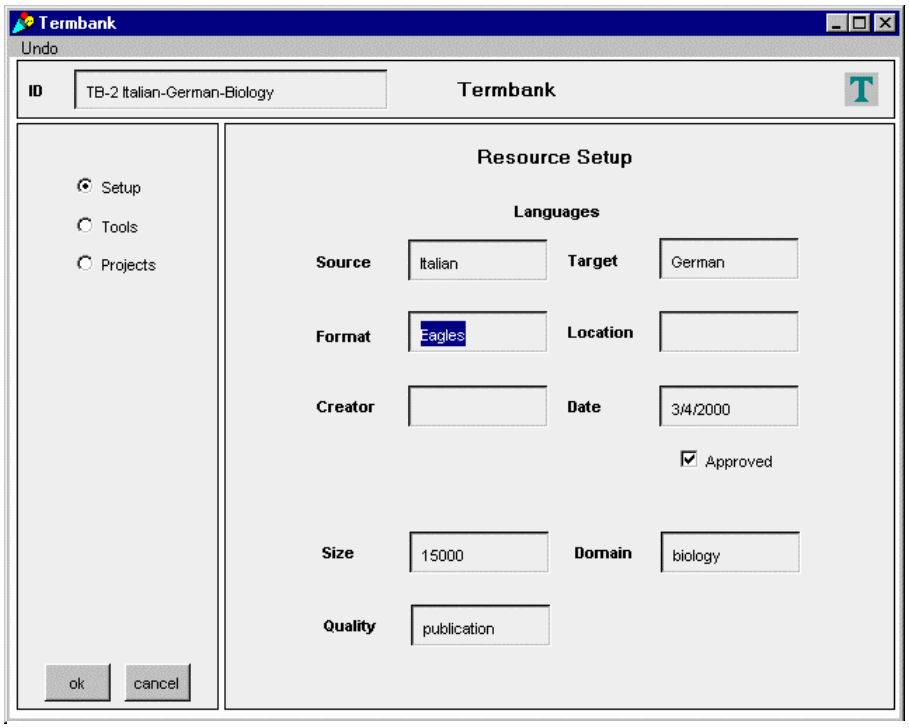



Most features are shared between the two resource profile types. There is a language pair, a resource format and a file location. All resources need approval to be considered for high quality translation.

Text style is the only feature that is assigned individually to translation memories.

**Figure 23: Translation memory tool – set- up**

### 9.5.2. Termbank tool




Term-banks are described primarily by the size of vocabulary. Other features are shared with translation memories.

**Figure 24: Termbank tool – set-up**

## 9.6. Group tool

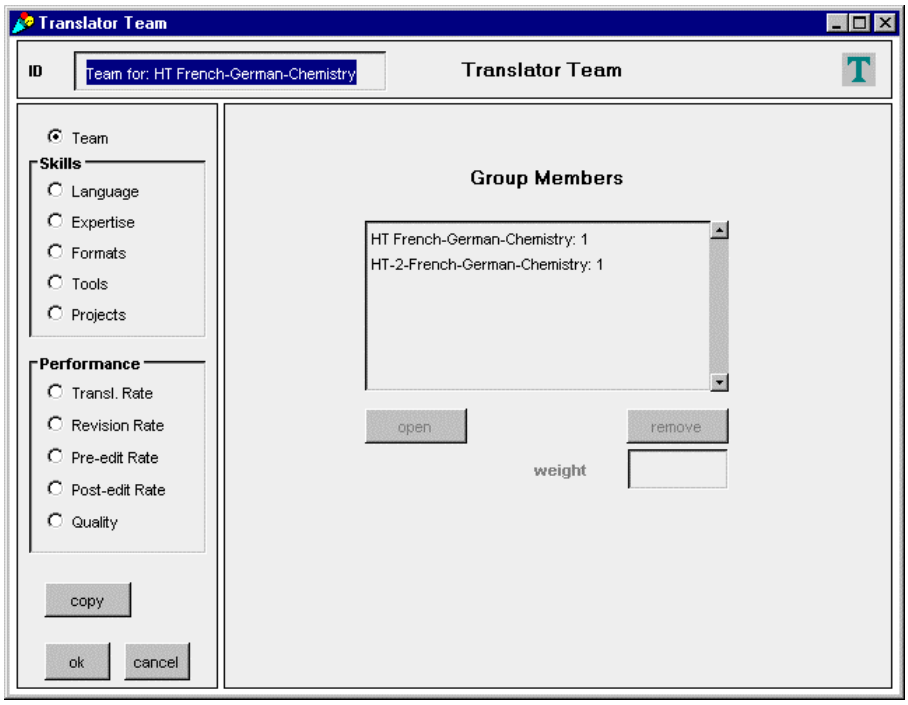



**Figure 25: Group tool**

All resources may be grouped into complex objects. A grouped object shares the features of its parts. Additionally members may be added or removed from the group.
The same applies to translators. Groups of translators may build a team. A person can be part of the team with full or partial workload. The performance of the team will be computed from performance of the individual group members.

## 9.7. Assessment Tools

The assessment establishes a relation between a project and a single resource. It quantifies the relevance of the respective resource. The relevance of a translation memory will be measured by the percentage of the project's text that matches text contained in the memory.

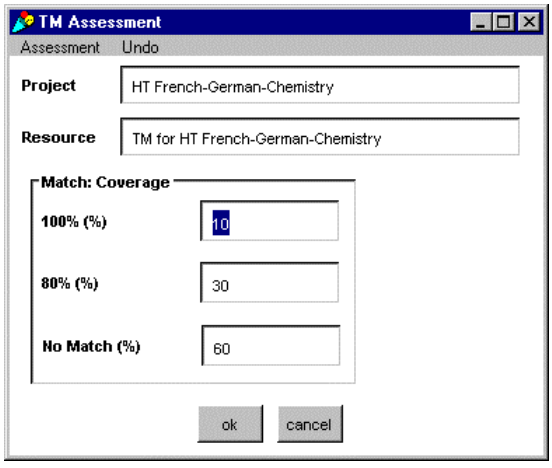



**Figure 26: Assessment tool – translation memory**

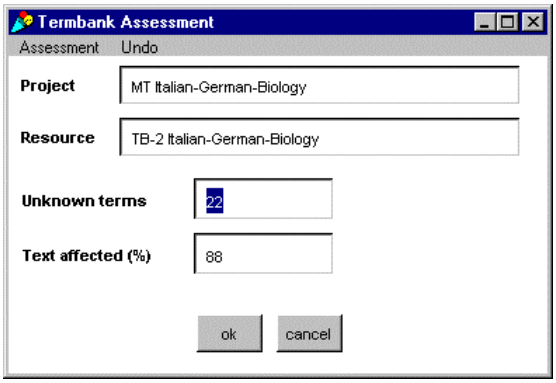



**Figure 27: Assessment tool – term-bank**

The relevance of term-banks is measured by the amount of unknown vocabulary and the percentage of text that is affected by missing terms.

## 9.8. Retrieving objects from the repository

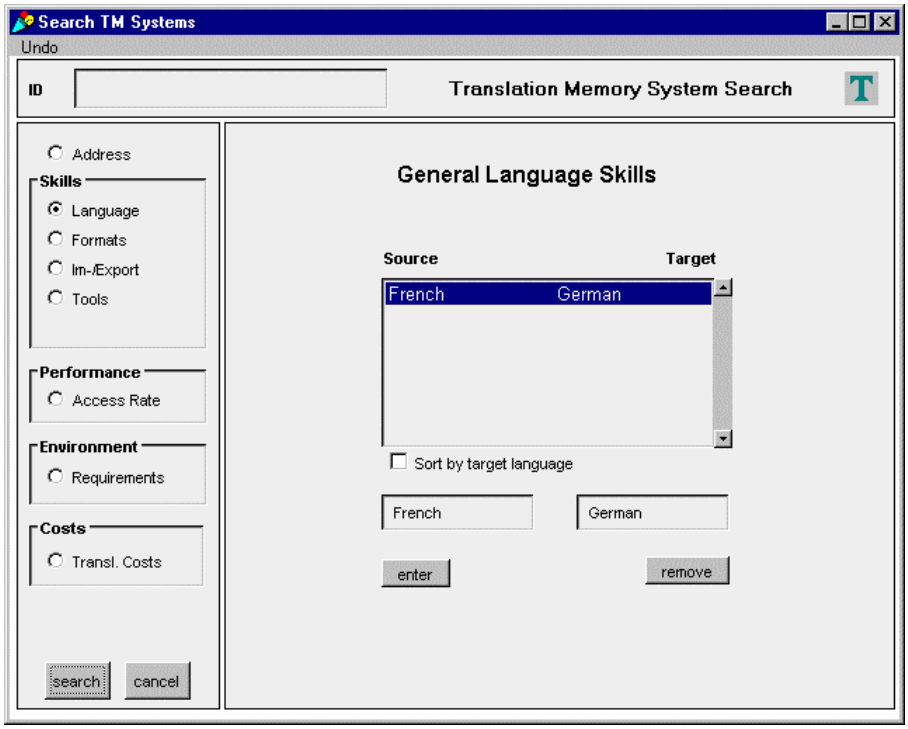



**Figure 28: Search for a translation memory**

Matching oriented retrieval of TransRouter objects, as has been mentioned earlier in this report, is performed by defining a sample object of the appropriate type using the same form filling approach used for the editing of TransRouter profiles. The snapshot on the left depicts a search profile for translation memory systems, where a language pair is specified. The only difference from conventional agent tools is the search button on the lower left side of the window. The result of the search is a list of relevant objects that can be explored using conventional TransRouter object viewing tools (see figure 29).


**Figure 29: Search results**

## 9.9. Browsing Routes

When the user pushes the 'Routes' button from the TransRouter repository tool, a list of possible routes for a translation project is presented. As has been described earlier, TransRouter will not present all possible routes but only one route for each relevant main agent.

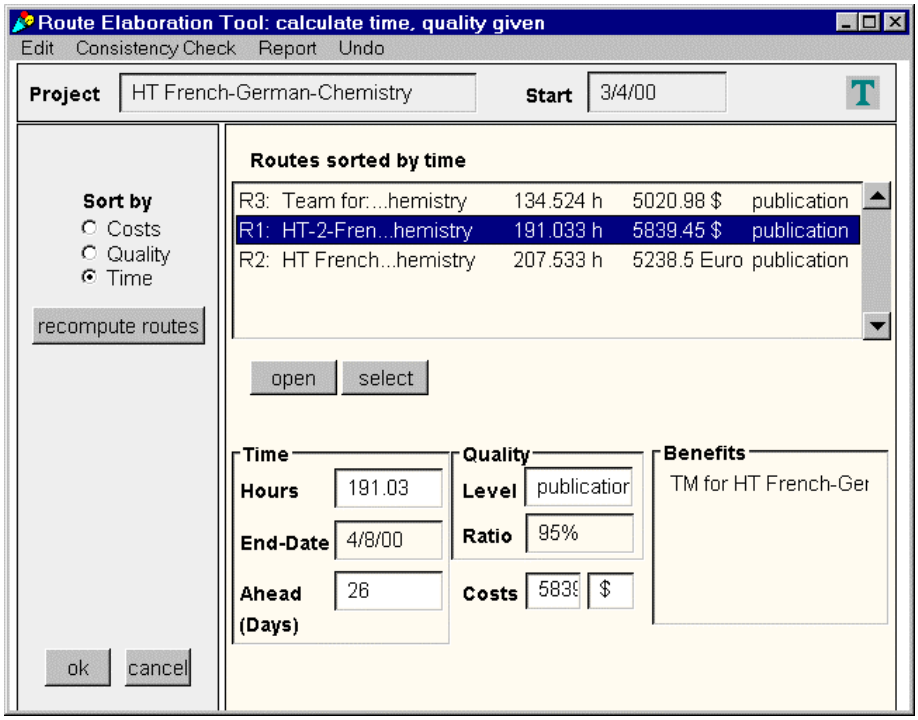



**Figure 30: Route elaboration tool – route list**

This tool shows a list of possible routes for a given translation project. The routes are sorted by time, cost or quality. Total values for these attributes are given within the list. If a route is selected more detailed information for this route is provided. Cost values may be converted to alternative currencies. Opening the route tool (open-button) will allow the modification of route features. Pushing the select-button will conclude the decision process.

This route is then chosen for further implementation.

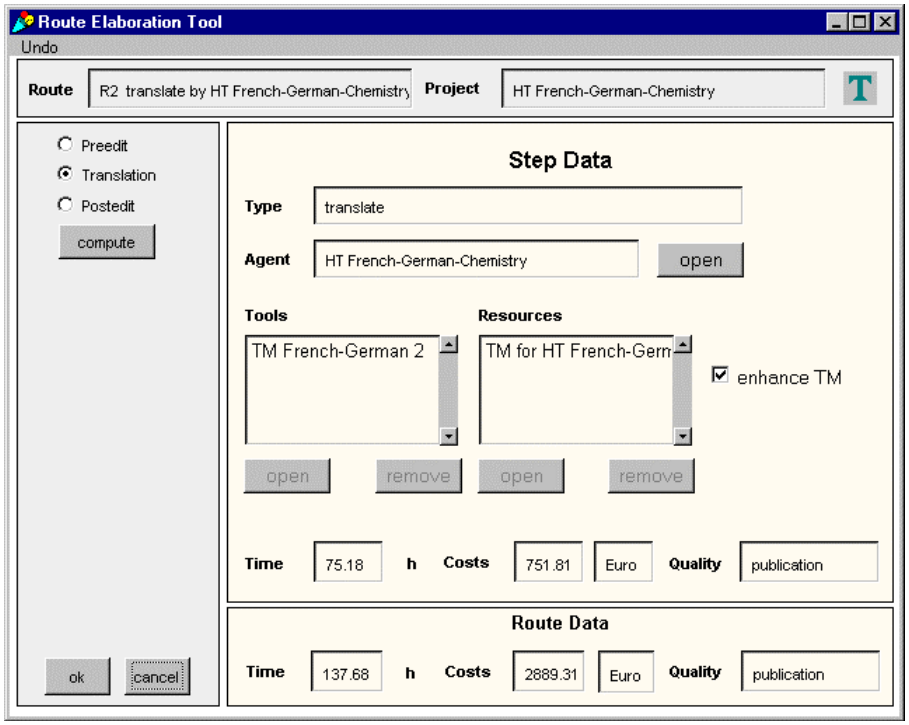



**Figure 31: Route elaboration tool – route view**

The route tool gives detailed information about a translation route. Agents, associated tools and resources are accessible for any single translation step. Agents, tools and resources may be removed from or added to a route using menus or drag and drop. Consistency between tools and resources is preserved automatically. Time, cost and quality values for each step are provided and shown in contrast to the overall values for the route.

In the decision making step these values will be computed by the system. They may be entered manually in order to document the true implementation process of the project in contrast to the computed values.

## 9.10 Utilities

### 9.5.3. Identifier definition

The first step in setting up a TransRouter system is the definition of the identifiers used in agent, resource and project definition. TransRouter provides an interface for identifier definition as depicted in figure 32.

The tool allows the selection of an identifier class – file formats, translation memory formats,

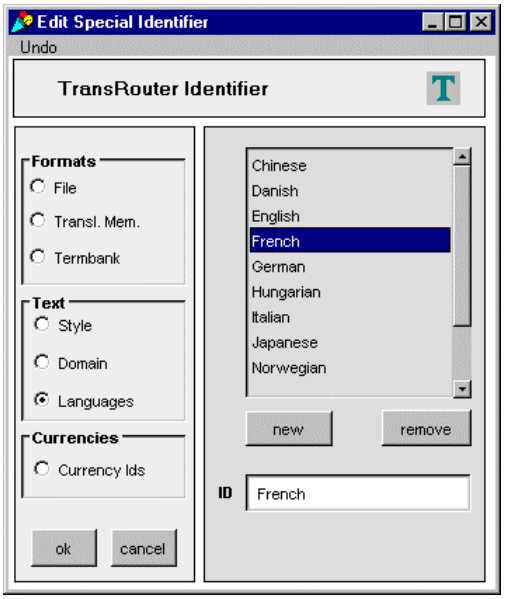



**Figure 32: Identifier definition**

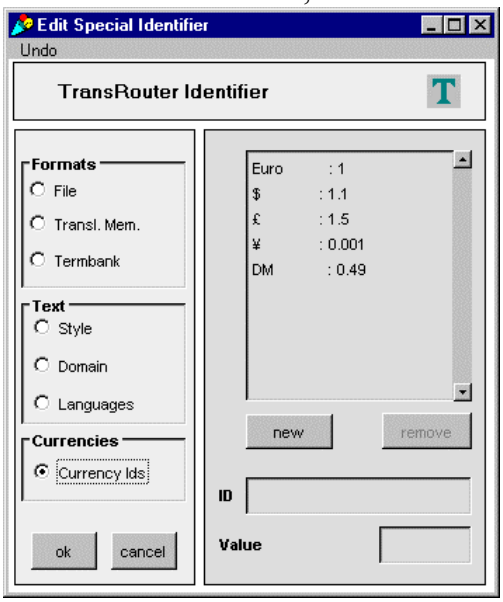



**Figure 33: Currency definition**

termbank formats, text styles, content domains and languages (shown in the snapshot). The definition of currencies requires additionally the assignment of exchange rates (see figure 33).

### 9.5.4. Default definition

Performance, quality and cost values as defined for individual agents, as described earlier in this report, can be enhanced by default values that apply to agent types. Performance defaults are defined by mappings similar to the performance mappings of individual agents. The only difference is that these mappings are assigned not to individuals but to object types. The tool for default manipulation therefore shows an interface (see figures 34,35) similar to the respective pages of the agent tools.

### 9.5.5. User management

Access to the TransRouter repository is only provided according to the access permissions that are assigned to a user. The definition of user ids and the assignment of access permissions is supported by a special tool (see figure 36).

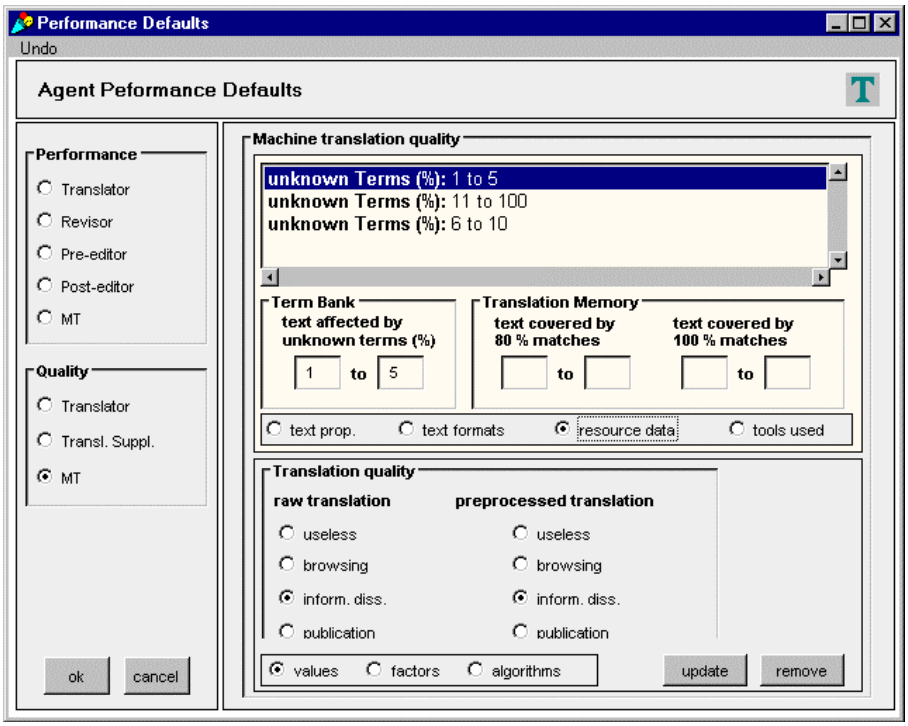



**Figure 34: Default definition – mt translation quality**

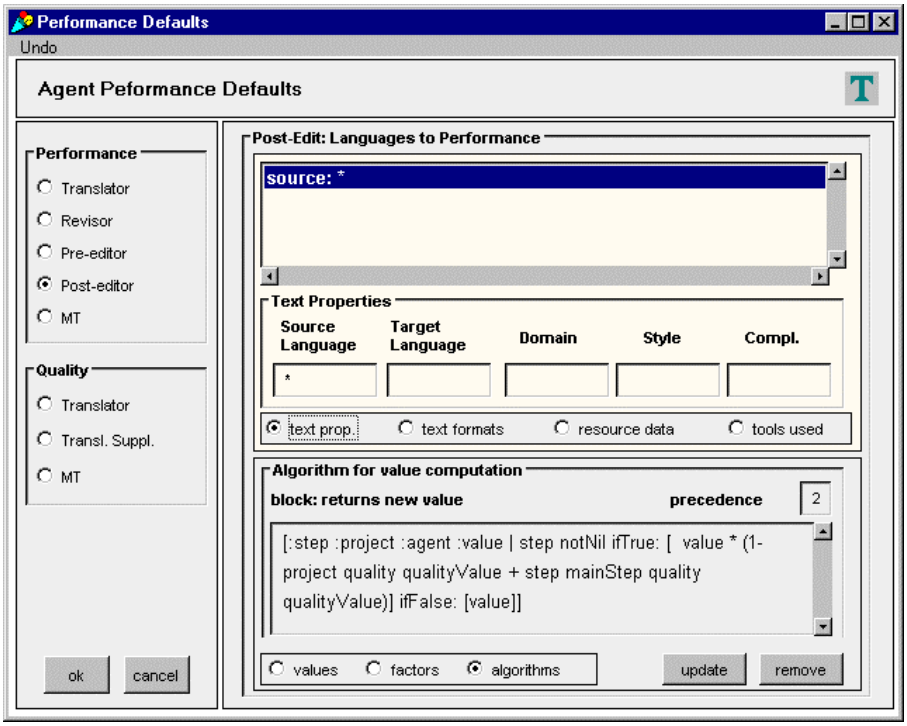



**Figure 35: Default definition: algorithm for quality computation**

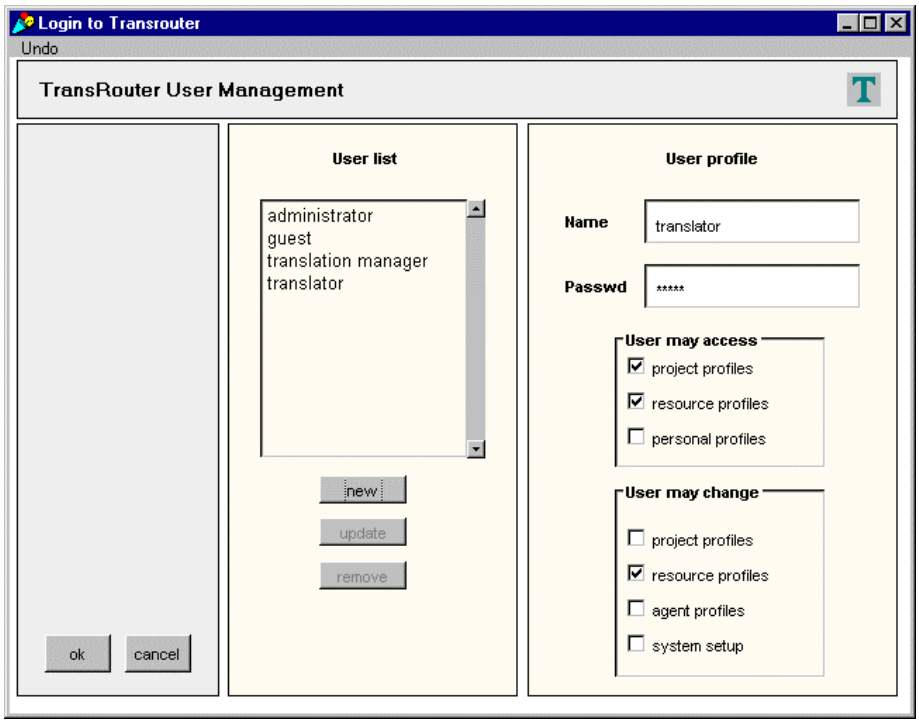



**Figure 36: User management**

# 10. The object model of TransRouter

The model of the TransRouter prototype is strictly object oriented. There are two hierarchies of classes. The first one covers the modeling of the content. It provides classes for agents, resources and projects. Complex attributes – addresses, performance mappings etc. – are described by their own classes. The class hierarchy of the content objects can be mapped almost directly to the class hierarchy of the interface objects, since most of the content objects have their own viewer, a tool providing access to this type of objects. The naming rules make this fact explicit. A *TRAgentViewer* is a viewer giving access to *TRAgents* and so on. The following two sections will give a short overview of the class structure of TransRouter accompanied by a very short description of every class.

## *10.1. Content objects*

- Object – The most abstract class of the entire Smalltalk system.
  - Transrouter – TransRouter repository containing all projects, agents and resources.
  - TRObject – the most abstract class of the content domain of TransRouter. Each object knows about the repository it belongs to.
    - TRAgent – most abstract class of all active objects – tools and translators – covering language pairs and data formats that can be processed. Furthermore mutual cooperation of agents is handled here.
      - TRTranslationSupplier – external provider of translation services.
      - TRTool – Translation software – covers licence and platform requirements.
        - TRAlignmentTool – tool for text alignment
        - TRToolWithResource – tool needing some data resource that has to comply with a given format.

        - TRMTSystem – machine translation system. Access to a term-bank is needed.
        - TRTerminologyManagementSystem – provides access to term-banks.
        - TRTranslationMemorySystem – provides access to translation memories.
    - TRTranslator – person translating, pre-editing and revising texts.
- TRGroup – group of agents or resources.
    - TRAgentGroup – group of agents. Performance values can be computed from the performance of the group members.
        - TRTranslatorTeam – team of translators.
    - TRResourceGroup – collection of resources. Most group features (e.g. vocabulary coverage) cannot be inferred from the features of the group members.
        - TRTBGroup – collection of term-banks.
        - TRTMGroup – collection of translation memories.
- TRProject – translation project. This class covers all features of the project setup and the further project documentation.
- TRResource – passive data resource used by translation agents.
    - TRTranslationMemory – translation memory.
    - TRTermBank – lexical resource, term-bank.
- TRAssessment – assessments represent the value of a resource for a given project
    - TRTMAssessment – the relevance of a translation memory for a project is given by the text coverage (100% matches, 80% matches)
    - TRTBAssessment – term-banks are assessed by the number of unknown words and the percentage of text affected by unknown terminology.
- TRAddress – address providing contact information for translators, software companies, translation clients etc.
- TRToolRequirements – covers all licence and platform requirements of software tools.
- TREvaluator – each project has its associated evaluator. The evaluator class implements algorithms of agent and resource selection. Evaluators keep hold of relevant agents and resources and elaborated route variants.
- TRGeneralSetup – there is one instance of this class associated to each TransRouter repository. It provides access to all identifiers and defaults that are part of the current TransRouter version.
- TRUser – each user of TransRouter is represented by an instance of this class. The main task of user objects is the control of access permissions.
- TRPerformanceKey – all keys of performance mappings are instances of this class.
- TRPerformance – all values of performance mappings are instances of subclasses of this abstract class.
    - TRBlock – piece of code that can process performance values.
    - TRFactor – numeric factor for performance computation.
    - TRTranslationTime – translation rate for translators (with respect to the required quality).
    - TREditTime – performance value for revision, pre- and postediting
    - TRMTTime – translation rate for machine translation.
    - TRTranslationTime – translation rate for human translation.
    - TRTranslationCost – abstract class for translation costs.
        - TRTranslationWordCost – cost per word

        - TRTranslationHourCost - cost per hour
    - TRTranslationQuality - general aspects of translation quality
        - TRMTQuality – refinement of quality for machine translation.
- TRPerformanceMapping – maps access keys to performance values.
- TRTranslationRoute – a sequence of translation steps. Translation routes may be compared with respect to translation time, cost and quality.
- TRTranslationStep – abstract class for translation steps. Each route consists of three translation steps – pre-processing, translation, post-processing.
    - TRAuxStep – abstract class for all pre- and post-editing steps.
        - TRNoStep – do nothing.
        - TRPostEditStep – abstract class for post-edit.
            - TRPostHTStep – revise text translated by a translator.
            - TRPostMTStep – post-edit text translated by a machine translation system.
            - TRPostTMStep – post-edit text taken from a translation memory in raw translation mode.
        - TRPreEditStep – abstract class for per-edit.
            - TRPreHTStep – prepare a text for a translator.
            - TRPreMTStep – prepare a text for a machine translation system
    - TRHTStep – translation by a translator
    - TRTranslationSupplierStep – translation by an external translation supplier.
    - TRToolStep – translation by some tool.
        - TRMTStep – translation by a machine translation system
        - TRTMStep – extract text from a translation memory in raw translation mode.

### *10.2. Interface Objects*

- Object – The most abstract class of the entire Smalltalk system.
    - Model – this abstract class covers dependencies between objects.
        - ApplicationModel – this is an abstract class applications employing user interfaces.
            - TRApplicationModel – this class covers aspects of user interfaces that are specific for TransRouter.
                - TRAddressViewer – interface to addresses.
                - TransrouterViewer – interface to the repository
                - TRARViewer – abstract interface class for agents and resources.
                    - TRAgentViewer – abstract interface class for agent profiles
                        - TRTranslationSupplierViewer – interface to translation suppliers.
                        - TRToolViewer – abstract interface class for all tools.
                            - TRAlignmentViewer – interface to alignment tool profiles.
                            - TRToolWithResourceViewer – abstract interface class to all tools that use data resources.
                                - TRMTViewer – interface to machine translation system profiles.
                                - TRTermBankSystemViewer – interface to terminology management system profiles.

        - TRTMSViewer – interface to translation memory system profiles.
    - TRTranslatorViewer – interface to translator profiles.
        - TRTeamViewer – interface to translator team profiles
- TRResourceViewer – abstract interface class for resource profiles
    - TRDictionaryViewer – interface to term-banks.
        - TRDictionaryGroupViewer – interface to term-bank groups.
    - TRTMViewer – interface to translation memory profiles.
        - TRTMGroupViewer – interface to translation memory groups.
- TRGeneralSetupViewer – allows the manipulation of identifier lists etc.
- TRGroupViewer – manipulation of group membership.
- TRPerformanceKeyViewer – access to performance keys.
- TRPerformanceMappingViewer – access to performance mappings
- TRPerformanceSetupViewer – access to default values.
- TRPerformanceViewer – access to performance values.
- TRProjectViewer – access to projects
- TRProtocolViewer – access to report texts.
- TRResourceAssessmentViewer – interface to resource assessments.
- TRRetrievedObjectsViewer – access to lists of relevant objects
- TRRouteSetViewer – access to routes sets.
- TRRouteViewer – interface to individual routes.
- TRStepViewer – interface to route steps
- TRToolReqViewer - access to tool requirements.
- TRUserViewer – access to user profiles.

# 11. Discussion and outlook

The TransRouter prototype as described in this report is a partial implementation of the software that is envisioned in the TransRouter project. The prototype falls short of the initial project specification with respect to the following issues:

- The tools that were developed in the course of the project are not integrated in the prototype. The architecture of the prototype however would allow tool integration with reasonable effort.
- The software platform of the prototype (VisualWorks) is exotic. It will find little acceptance in commercial environments. Since the TransRouter software still is rather experimental and leaves a couple of important questions open, the efficiency in rapid prototyping that can be achieved within the chosen software environment seems to be of greater importance. The object model of the prototype is a sound base for the further development of the system, even if a complete re-implementation should be considered.

The prototype gives a full account of the project, agent and resource profiles as defined in work package 3. The decision model of TransRouter is based on a hierarchy of selection rules and sorting criteria. It allows a considerable amount of user participation in the decision process. The decision model is embedded into a concise model of the workflow occurring in the decision process. All activities occurring in the workflow – search for agents, projects or resources – manipulation of routes, the update of agent or resource profiles are covered.

As the formalization process went on, some of the basic concepts were slightly modified. TransRouter was no longer conceived of as an instrument of quality prognosis but as a tool for quality management. Quality is no longer a feature of the outcome of a translation but of the translation procedures. Since the basis of the decision process is built on rough estimates of translation rates or text complexity, the result cannot be precise either. The next step on the course that was taken in the TransRouter project would be the design of a learning component for the system. This goal, however, is out of the scope of the current project.