# Profile of TEIRESIAS

## Category 1 – General

### *Domain*
The domain in which the system was originally developed was medicine. In particular, it was used in conjunction with MYCIN, a system designed to determine the identity of a bacterial infection in a patient and to then suggest an appropriate drug treatment. TEIRESIAS was used as a means of augmenting the knowledge base of MYCIN.

### *Main General Function*
The main function of TEIRESIAS is knowledge acquisition. It is a means by which new rules and facts can be added to the knowledge base of a performance program, in this case MYCIN.

### *System Name*
The system is called TEIRESIAS.

### *Dates*
The system was developed between 1976 and 1978.

### *Researchers*
The main developer of the system was Randall Davis.

### *Location*
The system was developed at the Computer Science Department at Stanford University, Stanford, CA 94305, USA, as part of the Stanford Heuristic Programming Project (a project which sought to develop the knowledge based paradigm).

### *Language*
TEIRESIAS is written in INTERLISP, an advanced dialect of the LISP language.

### *Machine*
The system runs on a DEC PDP-10 under Tenex.

### *Brief Summary*
TEIRESIAS is a program designed to function as an assistant to a domain expert in the task of building large knowledge bases. TEIRESIAS facilitates the interactive transfer of knowledge from a human expert to the knowledge base of a performance program (MYCIN), in a high level dialog conducted in a restricted subset of natural language. By using this high level dialog, the expert need have no knowledge of programming – all programming is performed by TEIRESIAS.

The knowledge base of MYCIN consists of facts (attributes, objects, and values which form a vocabulary of domain specific conceptual primitives) and inference rules expressed in terms of these known primitives. Thus TEIRESIAS is used to acquire new rules, which may also involve the acquisition of new primitives.

A central feature of TEIRESIAS is the use of a concept labelled meta-level knowledge. This can be defined as "knowledge about knowledge". In essence, this concept allows TEIRESIAS to use its knowledge directly, as well as to examine its knowledge, abstract it, and direct its application.

***Related Systems***

There are no systems from which TEIRESIAS was specifically developed, nor are there systems developed by others based upon TEIRESIAS (although TEIRESIAS was included in EMYCIN). TEIRESIAS was developed to be used in conjunction with MYCIN to allow the transfer of knowledge from an expert to MYCIN's knowledge base.

Other knowledge acquisition systems include COMPASS (Prerau, 1990) and OPAL. COMPASS is an example of a conventional knowledge acquisition method, that of structured interview to elicit knowledge which is then documented and tested, whereas TEIRESIAS concentrates on identifying errors in the knowledge base, drawing the experts attention to this and allowing the expert to add new rules to rectify the fault. OPAL uses a model of the domain to acquire knowledge directly from an expert via a graphical interface.

## Category 2

***Characterizations of givens***

TEIRESIAS consists of various types of information, including rule models, knowledge of representations (schemata), as well as various heuristic rules. Rule models contain lists of information, including a list of rules from which that particular rule model was developed, as well as a list of typical characteristics of rules from the subset.

Schemata are information about the representations of the conceptual primitives that form the vocabulary of the inference rules that TEIRESIAS transfers to the knowledge base. These schemata can be considered as record-like or frame structures which detail the structure of primitives as well as the interrelationships between different schemata.

Also included are various heuristic rules, held in the form of production rules, which are used in various circumstances, such as when trying to decipher the text of a new rule. TEIRESIAS also has sets of procedural knowledge, which can be used for a variety of purposes, for example generating the rule models or for creating new instances of the various hierarchies discussed in Category 5.

TEIRESIAS requires the following information to be given by either the user or the performance program. Information required from the performance program include the rules and facts in MYCIN's knowledge base, as well as information from MYCIN's inference engine to allow TEIRESIAS to generate explanations.

Information given to TEIRESIAS by the user are new inference rules which may or may not include new conceptual primitives, information regarding the primitves, plus various information used during bookkeeping i.e why was this new rule required.

***Characterization of Output***

TEIRESIAS outputs information both to the expert and to the knowledge base of the performance program. To the expert, TEIRESIAS outputs:

- Its interpretation of a new rule in natural language
- Questions requesting information that TEIRESIAS requires to facilitate knowledge transfer
- An explanation of how the performance program reached its conclusion

To the knowledge base, TEIRESIAS outputs (in INTERLISP dialect):

- New rules
- New conceptual primitives

TEIRESIAS also generates information that is held internally within TEIRESIAS. That is, it updates its rule models in accordance with changes to the knowledge base of MYCIN to create more accurate rule models.

***Characterization of Data***

The rules output by TEIRESIAS are its interpretations of the LISP forms of the rules. These rules and TEIRESIAS' interpretation of these rules are validated as accurate by the expert, and indeed TEIRESIAS also checks that the rules rectify a particular identified shortcoming in the knowledge base. Thus the output can be considered reliable. It can also be considered complete in the sense that TEIRESIAS' duty is to transfer a rule to MYCIN – this it does an

The updated rule models are a further output of TEIRESIAS. These are incomplete as the knowledge base upon which they are based is incomplete, thus it cannot form a complete picture of what the typical characterizations of a particular subset of rules are. However, over time as the knowledge base expands and the number of rules in a subset increase, the system's rule models should become more reliable as a result of the greater number of rules upon which they are based.

***Generic Tasks***

There are three generic tasks that TEIRESIAS uses. These are classification, recognition, and learning (through generalization). The first task is classification. When attempting to acquire a new rule, TEIRESIAS first attempts to determine what rule model this new rule will belong to i.e. it attempts to classify this rule as belonging to one of its set of rule models – this helps the system during the rule acquisition process.

Recognition is applied during the rule acquisition process by TEIRESIAS using the rule model for that type of rule as a means of guiding the rule acquisition process. It has expectations about this new rule and it uses these biases to direct the acquisition process in a particular way.

The third generic task mentioned was learning through generalization. This is performed in the maintenance of the rule models and is discussed more in Category 10.

***Theoretical Commitment***

The central theme of TEIRESIAS is the exploration and use of meta-level knowledge. It exists in several different forms in the system but can be summed up generally as "knowing what you know". This knowledge makes possible a system which has both the capacity to use its knowledge directly, and the ability to examine it, abstract it, and direct its application. An example of such knowledge is the rule models.

***Reality***

TEIRESIAS attempts to represent the interaction between the domain expert and the performance program as being similar to the interaction between a teacher and a student. The teacher (the expert) continually challenges a student with new problems to solve and carefully observes the student's (MYCIN) performance. The teacher may interrupt the student to request a justification (an explanation) of some particular step the student has taken in solving the problem or may challenge the final result. This process may uncover a fault in the student's knowledge of the subject and result in the transfer of information to correct this fault i.e. TEIRESIAS is used as a means of allowing knowledge to be transferred from the expert to the performance program. Thus, TEIRESIAS' approach involves knowledge transfer that is interactive, and that is set in the context of a shortcoming in the knowledge base.

## Category 3

***Completeness***

The system has been fully implemented within the developer's environment and has been used in conjunction with MYCIN to successfully allow MYCIN's knowledge base to be expanded.

***Use***

TEIRESIAS was not tested either by experts in the developer's own working environment or by experts from outside the original development situation.

***Performance***

The system developer performed an evaluation of TEIRESIAS. The developer's experience with manual knowledge acquisition in a performance program was used to provide a perspective on how effective a tool TEIRESIAS was. The general conclusions were that interactive knowledge transfer seemed best suited to task domains involving problem solving that is primarily a high-level cognitive task, such as the medical diagnosis domain, and that knowledge acquisition in context (i.e. using the rule models as a basis for expectations) appeared to be a useful guide whenever knowledge of the domain was as yet un-specified.

Certain shortcomings that were found were that it was only possible to add one rule at a time to MYCIN, whereas in some cases, specifically during the early stages of knowledge base creation, it would have been preferable to deal with groups of rules. One further problem was that the natural language capabilities were limited. They were based on a simple keyword approach and contained no grammatical knowledge.

**Category 4**

***Phases***

TEIRESIAS has one main phase – acquisition of new inference rules. This can consist of 4 main subphases. These are:

1. Tracking down the bug – this is determining what knowledge is missing
2. Interpreting the new rule – interpreting the new rule and converting it into the correct LISP form for MYCIN
3. "Second Guessing" – checking the rule to see if it fits with the system's view of what rules of this type should look like.
4. Adding the rule to MYCIN's knowledge base and updating its own rule models

Accepting a new rule may also involve the acquisition of new instances of data types or of new instances of types of data types (i.e. additions to the vocabulary of MYCIN knowledge base from which the inference rules are formed)

***Subfunctions***

Various subfunctions are involved in the process of adding a new rule. One of these is classification. For example, during debugging, the system attempts (with the assistance of the expert) to determine where the shortcoming in the knowledge base is, and then determine what the conclusion of the new rule should be. It therefore classifies this new rule for example, as being a "rule which should conclude organism identity" and then calls up the rule model corresponding to rules that make a conclusion about organism identity (rule models are based around the contents of the action part of rules). This is selection i.e. the system has selected a rule model with which to guide the knowledge acquisition process.

Another example of a problem-solving approach is that previous experience is used to guide the rule acquisition process. Rule models are in essence what the system has understood rules of this type to be, based upon its experience of what rules of this type have contained in the past. Thus based upon a set of previous experiences, the system has certain expectations about what the new rule will contain, and this expectation is used as a means for guiding the process. This is model-driven reasoning to solve a problem. The system also uses a data driven approach when deciphering the English text. Based upon its set of standard templates of predicate functions, the system can generate various interpretations of a line of text and give each these interpretations a score that reflects how it was assembled (e.g. if assumptions had to be made by the system then the score will be low). Thus, the data has been allowed to suggest interpretations, but the system maintains certain biases (obtained from the rule model) about which interpretation is likely to be correct. Thus during text processing, interpretations suggested by the text (data-driven, "bottom up" mode) are intersected with the expectations (hypothesis-driven, "top down") provided by the classification process.

***Use of Simulation or Analysis***

During the rule acquisition process, TEIRESIAS performs a simulation to ensure that the new rule does in fact rectify the shortcoming in the knowledge base, and that the performance program will indeed now invoke the rule in the current context and reach the correct conclusion. It also performs an analysis of the knowledge base of the performance program whenever the base is modified. A statistical analysis is performed and the results of this analysis are then used as the basis upon which the rule models are then re-evaluated and updated to reflect the current contents of the knowledge base

***System/Control Implementation Architecture***

The system architecture consists of meta-level knowledge which includes associated procedures and heuristic rules that are applied as and when called to do so by the meta-level knowledge. In essence, this meta-level knowledge can direct its own implementation.

## Category 5

***Characterization of Structure Knowledge***

TEIRESIAS distinguishes between 3 different levels of knowledge. These are levels 0, 1, and 2. The first level of the hierarchy (Level 0) contains the object level knowledge – medical knowledge of cultures, organisms, drugs, etc. This is the knowledge base of MYCIN. The next level (Level 1) is concerned with the conceptual building blocks of the knowledge representation – the predicate functions, attributes, values, rules and so on. Level 1 knowledge consists of the rule models and the schema. Level 2 knowledge contains knowledge about the schema themselves. Levels 1 and 2 are examples of meta level knowledge – the concept underpinning TEIRESIAS. These levels of knowledge correspond to different phases of the system. For example level 1 knowledge is used to create a new rule or a new instance of a schema, whereas level 2 knowledge is used to create a new type of schema.

***Characterization of Process Knowledge***

For active knowledge, the effect of that knowledge is classification as well as recognition. Active knowledge during debugging helps to classify the rule as being a member of a particular type of rule model. During debugging, the system determines that it needs a new rule which makes a certain conclusion. It does not know what is contained in the premise of the rule, but it does know what the conclusion should be (this may be in general or specific terms) and from this partial situation description, the system can classify which rule model the rule will belong to. When the rule model has been selected (it can be argued that selection is a type of classification), that model then becomes active knowledge in the system and the effect of this knowledge is that it allows the system to recognize the situation it is in. That is, it has expectations about the world, and it uses these expectations to guide the knowledge acquisition process.

***Deep or Surface***

There are 2 levels of meta-level knowledge in TEIRESIAS. The base of knowledge at level 1 consists of the building blocks of knowledge representation e.g. the schemata, rule models. Since no reference is made at this level to any specific instance of any of the primitives (e.g. an actual rule), this level of knowledge has a small degree of domain independence. Knowledge at level 2 is a deeper level of knowledge that contains knowledge about representations in general, and about the process of specifying them via declarations. Surface knowledge can be considered the rules and facts in the knowledge base of MYCIN, thus the base of knowledge at level 1 can be considered deeper than surface knowledge since it has a degree of domain independence, and the base of knowledge at level 2 is at a deeper level than that at level 1. However, it is difficult to state explicitly exactly where between deep and surface the knowledge can be placed. It can be said however that TEIRESIAS attempts to use deeper levels of knowledge of representations.

## Category 6

***Search Space***

The system searches through various search spaces. The first space to be searched through is the rule model trees to determine which rule model should be used during the rule acquisition process. Each state is a rule model, which contains the systems's expectations about what rules in this subset should contain. The system also searches through the tree of parses which it generates, to create of the new rule. Each node of the tree represents an individual parse, and each path of the tree represents a consistent interpretation of the rule. Also, the system may have to search through the schema hierarchy if a new conceptual primitive is to be added to the system. Each node in the schema represents a data structure of a particular type.

***Space Traversal***

The rule model trees are searched to obtain the relevant rule model. The system starts at the top of the tree and descends until reaching a model of the appropriate type or encountering a leaf of the tree. As the system descends the tree, the rule models become more and more specific with regards to the action part of their rule (i.e. at the top of the tree, the model may be rules which conclude about a category of an organism; at the bottom of the tree, the models may be about rules that conclude that the category of the organism is enteriobacteriaceae. The schema hierarchy involves the use of inheritance, in which each node represents an instantiation of the level above it in the tree, from which it inherits the same properties, which are augmented with some new properties specific to this new instance. Traversal of this tree is achieved by the procedures associated with the slots in the tree and through using the parent/child links that are explicitly stated at each level of the hierarchy. In the tree of parses generated by TEIRESIAS, each node represents the LISP interpretation of a single line of text of the new rule. Travelling from the root of the tree to a leaf via a single path represents a single, consistent interpretation of the complete rule.

***Search Control Category***

The system uses no strategy that is unique to the problem it attempts to solve. All techniques are general AI techniques that have been used in AI systems before.

***Standard Search***

The system adopts a generate and test approach. For example, it generates the tree of parses which consists of all likely meanings of each line of natural language text of the new rule. It then uses the tree to generate all possible consistent interpretations of the entire rule and tests them against its expectations of what the rule should contain, based upon the rule model the system is using to guide the rule acquisition process.

***Search Control Characterization***

The basic type of search strategies employed by TEIRESIAS are standard depth first searches, used for example when travelling down the schema or when generating the tree of parses i.e. new directions are followed as they arise. It also exhibits the use backtracking, specifically when generating the tree of parses during rule interpretation. The system can undo its current set of meaning assignments and tries all other possible alternatives to generate the tree.

There are various additional mechanisms that are in place to help prune the searching process. A simple first pass strategy is built into the driver routines that interpret each line of natural text of a rule, which seeks to determine which words have unambiguous meanings first. This reduces the number of possible interpretations of the line of text and thus results in a smaller parse tree. Various heuristics rules are also employed both during the debugging phase and during rule interpretation which help to focus the system's efforts on the problem at hand in a particular manner and attempt to make the acquisition process more streamlined and effective. The use of rule models is a strategy that also helps to focus the systems efforts i.e. the system has expectations about the new rule and it uses these expectations to guide the acquisition process.

***Subproblems***

Evaluation of partial solutions is possible in a limited sense. The system is unable to evaluate a partial rule to determine if it will fix a shortcoming in the knowledge base of the performance program. To perform this check, the complete rule is required. However when generating the parse tree, the generation of each parse (the LISP clause for a single line of text) is in a sense a subgoal of achieving the overall goal of interpreting the new rule. Each individual parse is evaluated and scored on the basis of the how well it was suggested by the text. Each parse is evaluated on its own and is thus independent from the other parses. However, when attempting to interpret the rule as a whole, the parses are linked together via the tree and scored collectively according to expectations from the rule model. TEIRESIAS gives greater emphasis to this score rather than that generated by each individual parse, thus the evaluation of subproblems is fairly limited.

***Search Control Representation***

Search control knowledge is expressed explicitly in the form of rules and procedures. Heuristics are expressed as rules and procedures exist, such as the procedures associated with each slot in the schemata, or the procedures used in the texst interpretation process.

***Search Control Strength***

Through the use of meta-level knowledge, TEIRESIAS attempts to use a search control that is partially domain independent and knowledge free. For example, the rule models are an example of knowledge of primitives for a specific representation i.e. they understand what a rule is, what role it plays etc. However, at this level of knowledge, no reference is made to any specific instance of this primitive, thus it can be said that this knowledge has a certain degree of independence and it is this knowledge which is primarily used to guide the acquisition process. However, the heuristics and procedures that are invoked during knowledge acquisition are invoked as a result of specific domain information.

**Category 7**

***Failure Method***

TEIRESIAS seeks to protect itself from reaching an incorrect conclusion by requesting information or confirmation from the user. For example, it outputs its interpretation of a new rule to the expert and seeks verification of its understanding, rather than interpreting the rule and charging ahead by implementing it in the knowledge base. When the user accepts an interpretation, the system still tests the rule to ensure that it does in fact correct the identified knowledge shortcoming in the knowledge base before adding it to MYCIN. If an incorrect rule is added to the knowledge base, then the rule must be removed interactively by the expert.

***Uncertainty***

Uncertainty can exist in terms of:

- the interpretation of a new rule entered by the expert
- where in the search space the system should now go

***Management of Uncertainty***

The system has several methods of dealing with uncertainty. The first is to seek advice from the expert. For example, if the system is at a node in the search space that has 3 possible branches and the system is does not have sufficient information it can request the required data from the expert. Alternatively sometimes the system can hop over some parts of the hierarchy of which it is unsure.

When interpreting a new rule, the system scores possible interpretations and outputs to the user the interpretation it believes to be the most accurate. The higher the score, the more the interpretation matches the system’s expectations of the new rule according to the rule model the system is using to guide the acquisition process. The score is a combination of two scores. The first score is determined from the data-driven “bottom up” approach to generating the interpretations of each line of the text. The score rates each LISP clause based upon how strongly it was suggested by the English text. The interpretations are arranged in a parse tree, and following the tree from its root along a single path to a leaf represents a consistent interpretation of the rule. The rule model is used to determine a score for an entire path that indicates how well the set of clauses meets expectations. The two scores are added in a way that emphasizes the expectations (i.e. the system hears what it wants to hear) and the highest score is ranked the most likely interpretation.

***Management of Time***

TEIRESIAS does not have any issues related to the management of time

**Category 8**

***Knowledge Representation Method***

The system uses models to represents types of rules. These models are lists, detailing the rules which form the subset, as well as the characterizations that are considered to be typical of this set. The system also holds knowledge in terms of procedures, for example knowledge of how to generate these rule models. Rules are also present in the system in the form of heuristics that are employed for example during the rule interpretation and debugging phases of knowledge acquisition. Schemata are also used to represent knowledge about conceptual primitives. These are essentially record-like structures which have slots and associated values which describe the structure of its instances. Associated with each slot is a procedure which actually tells the system how to fill in the slots to create an instance.

***Knowledge Representation Generality***

All knowledge in the system is expressed in coded INTERLISP structures. When conversing with the expert, the system either converts the coded INTERLISP structures into natural language or converts the natural language into coded INTERLISP structures, depending upon the information flow. Communication with MYCIN is performed using INTERLISP code.

***Knowledge Structuring***

There are various hierarchies used in TEIRESIAS. The set of rule models is grouped into a number of tree structures. At the root of each tree is the model made from all the rules that conclude, for example about a given attribute. Below this are two models dealing with all affirmative and all negative rules Below this are models dealing with rules that affirm or deny specific values of the attribute and so on. The further down the tree the system traverses, the more specific the models become. Models are organized around the content of their action part. The schemata are organized into a generalization hierarchy that indicates categories of representations and interrelations. In this hierarchy, extensive use is made of the concept of inheritance of properties. Both these structures exist in TEIRESIAS all the time, although their contents may change (branches may extend or shorten for example) in accordance with changes in MYCIN's knowledge base. When deciphering natural language text, TEIRESIAS generates a tree of parses from scratch. TEIRESIAS generates all possible interpretations of each line of text and then arranges them into a tree of parses. The links in the tree represent consistent sets of interpretations of the text. That is, any single path through the tree from the root to a leaf is a consistent interpretation of the rule. This type of structure is obviously domain dependent as the generation of all the parses is dependent upon the actual instances of conceptual primitives i.e. it is derived on the basis of object (domain) level knowledge.

## Category 9

### *Alternative Representations*

TEIRESIAS does not use alternative representations for the same piece of knowledge.

### *Alternative Solution Methods*

TEIRESIAS does not use alternative solution methods to reach the same solution.

### *Optimization*

The system does not always produce the best answer. One output from TEIRESIAS is a new rule that has been interpreted correctly. The system does not always produce the best answer (i.e. its interpretation is sometimes wrong), but it is able to get to the right answer in a reasonable number of steps (usually after a maximum of 4 attempts). A further output from the system is its updated set of rule models. These models can be judged to be the best answer *at hand* since they reflect the current state of the knowledge base of MYCIN.

### *Multiple Results*

The system can produce multiple results i.e. it develops several different interpretations of a new rule when deciphering the natural text. It attempts to order these through scoring the possible interpretations according to the biases expressed by the rule model being used to guide the acquisition process, and by the interpretations suggested by the data-driven approach. The interpretation with the highest score is deemed to be the likely best result.

## Category 10

### *Interaction*

TEIRESIAS interacts with the user through a high-level dialogue conducted in a restricted subset of natural language. All questions and responses from TEIRESIAS are either pre-formed or based on a simple template completion mechanism (which can result in phrases such as "a infection", instead of "an infection"). Responses from the user are of three general types: single token answers to multiple choice questions, strings belonging to a synthetic language with a formal grammar, and heavily stylized natural language sentences using a restricted vocabulary. Sentences are interpreted using a straightforward keyword analysis.

There are no other forms of communication with TEIRESIAS i.e. there are no pull down menus, or graphs, or diagrams of goal trees etc.

***Data Collection***

When acquiring a new inference rule, the final execution consists of placing the rule in the knowledge base. Before doing this, the system checks the rule to ensure that it fills the identified hole in the knowledge base. To pass this check successfully, the rule must be complete (in the current context) and fully interpreted by TEIRESIAS otherwise the rule will not rectify the identified shortcoming in the knowledge base. Thus, the system requires that the rule is complete and fully interpreted by TEIRESIAS before it can attempt execution. Partial rules cannot be executed.

***Data Format***

Information is given to TEIRESIAS in a restricted subset of natural language form

***Acquisition***

Knowledge acquisition is the fundamental goal of TEIRESIAS – it is designed to allow an expert to augment the knowledge base of a performance program without the expert having to know any programming. The system adopts an interactive approach to augmenting the knowledge base of a performance program. The central theme of the system is meta-level knowledge, which allows a system to use its knowledge directly, as well as giving it the ability to examine it, abstract it, and direct its application. The primary example of meta-level knowledge in TEIRESIAS is the rule models employed to guide the acquisition process. These rule models allow the system to have certain expectations about what a new rule should contain, and it uses this expectation to guide the process and also to evaluate this new rule in terms of how well it meets the systems expectations about rules of this type.

***Learning***

TEIRESIAS exhibits learning in the manner in which it maintains its rule models. Rule acquisition is a combination of model directed understanding and concept formation. When an expert inputs a new rule, TEIRESIAS uses a rule model to help interpret the rule. The interpretation is then deposited in the knowledge base of the performance program and is subsequently used to update the rule model via the concept formation process – the system produces a revised concept of what constitutes a typical rule in this subset as a result of the addition of the new rule. The system is thus constructing its models on the basis of experience, keeping those models up to date with the current knowledge base and using them to help acquire new knowledge. Thus, performance on the acquisition of the next rule of this subset may be better as a result of the system having a better picture of what such a rule should look like. The rule models are now computed from a larger set of instances and their generalizations are likely to be more valid. As a result of the being revised whenever the knowledge base changes, these models change in accordance with the changes of the knowledge base and are an accurate reflection of the shifting patterns in the knowledge base.

***Explanation***

One interesting paradox of TEIRESIAS is that it has an explanation facility that allows it to explain to the expert the behaviour of MYCIN. This explanation of behaviour is defined in terms of the derived goal tree and the rules that MYCIN invoked. This proves particularly helpful during the debugging phase of knowledge acquisition. However, TEIRESIAS is unable to explain to the expert its own behaviour. For example, the expert may be unaware of the existence of the rule models and there is no way that TEIRESIAS

**Category 11**

***Strengths***

The strengths of TEIRESIAS are:

- It allows knowledge to transferred to a knowledge base in a high level language, thus the expert need have no programming knowledge
- The system has a learning capability which ensures that its rule models are constantly updated and a reflection of the current state of knowledge in the knowledge base i.e. its own knowledge base is dynamic
- It seems well suited to domains involving problem solving that is primarily a high level cognitive task, with a number of distinct, specifiable principles
- It uses its previous experience to aid the expert via the second guessing facility it has i.e. it attempts to validate rules according to its rule models
- It has the ability to examine its own knowledge, abstract it and direct its application.
- Its own knowledge base is dynamic and changing in accordance with changes to the performance program's knowledge base

***Weaknesses***

The weaknesses of TEIRESIAS are:

- The natural language processing capability is limited. It is based on a keyword approach and contains no grammatical techniques
- The rule models are created using a purely statistical approach, which assumes that frequent appearance equates to importance, which may not always be the case.
- Also, the use of a statistical approach can eliminate variations and the system may lose information about subsets of rules as a result of this averaging process
- The system can only accept one rule at a time, whereas in the early stages of knowledge base development it may be more convenient to deal with sets of rules
- The explanation facility has a limited understanding of what constitutes an understanding (although it is still an effective tool for determining deficiencies in the knowledge base).