The Starpicker Expert System — A Problem in Expertise Capture

David A. Smith and G. Martin Hudson
Hughes Information Technology Corporation
16800 E. Centre Tech Parkway
Aurora, CO 80011, U.S.A.
Tel: (303) 344-6175 Fax: (303) 344-2903
smith@redwood.hac.com

KEY WORDS AND PHRASES

Expert system, spacecraft planning, attitude determination.

ABSTRACT

This paper describes the Starpicker expert system, a tool for spacecraft operations planning. Both programmatic and technical aspects are discussed.

BACKGROUND

The Space Precision Attitude Control System (SPACS) Star Sensor was designed and developed by Hughes for use on the HS-318 satellite bus. This is a spin-stabilized spacecraft whose purpose is to provide an accurately positionable platform in earth orbit. The Star Sensor serves as the primary attitude reference.

The function of the Star Sensor is to determine the orientation of the spacecraft spin axis in three-dimensional space, as shown in Figure 1. The sensor operates by measuring the elevation of two selected stars relative to the equatorial spin plane of the spacecraft. These stars are chosen near the spin plane and are ideally separated by about 90 degrees of rotation. Using a catalog of absolute star positions on the celestial sphere, the spin axis of the spacecraft can be accurately determined.

Two sensors are placed on the rotating portion of the spacecraft. Each sensor has a vertical field of view spanning six degrees. One sensor is centered three degrees above the spin plane and the other is centered three degrees below, resulting in twelve degrees of total coverage. The sensor in use is programmed to "open" or "gate" at fixed moments during the rotational period of the spacecraft — once for the primary star and once for the secondary star. During each gate, the elevation of any bright object appearing in the sensor will be measured.

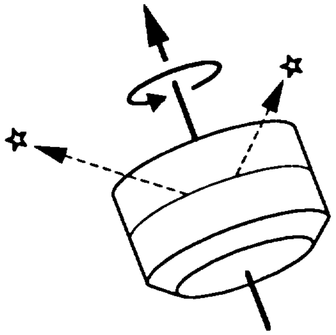

Figure 1. Spacecraft attitude determination

THE PROBLEM

It would seem that with an estimated 200 billion stars in our Galaxy, there would be plenty of stars to choose from. However, a variety of constraints combine to make this a challenging problem in operations planning:

- The sensor has programmable sensitivity — at its most sensitive, the sensor can gate on about the 300th brightest star in the sky.
- Both stars must be within the same sensor's field of view (either above or below the spin plane).
- The separation between the two stars should be between 30 and 150 degrees — the closer to 90 degrees the better.
- The sensor cannot discriminate between stars in the sensor which are less than 4 degrees apart. In this case, neither star is usable.

244

- The previous constraint also applies when one of the bright planets appears in the sensor — i.e., Mercury through Saturn.
- Glare generally prevents use of any star within 60 degrees of the sun in any direction, although the brightest stars are still usable somewhat closer than this. The motion of the sun by about one degree per day frequently limits the number of days that a star can be used.
- When the moon is in the sensor, glare generally wipes out any stars 15 to 20 degrees before or after the moon. This effect depends on the phase (and therefore brightness) of the moon.
- The appearance of the earth in the sensor during the spacecraft orbit may obstruct visibility of stars. The glare of the sun shining on the earth makes the affected area larger.
- Over time, the sensor becomes degraded in sensitivity, making dimmer stars unusable and reducing the glare-immunity of brighter stars.
- Some stars vary in brightness over periods ranging from hours to months, making their use problematic. Some other stars seem to yield low-quality data, presumably from the presence of nebulosity or other sources of sensor noise.

The above constraints must all be accommodated in order to achieve nominal operations. Unfortunately, there are times when not all constraints can be satisfied. In these cases it is necessary to find the best possible fall-back solution so that operations can continue.

EXPERT SYSTEM DEVELOPMENT

The Starpicker expert system was built to help choose attitude determination stars. The expert system captures both the nominal selection criteria described above and the fall-back heuristic methods.

The development of Starpicker is outlined in Figure 2. The idea grew out of a study that focused on automated capture of human operations expertise. Starpicker is the first such tool to be identified and built.

Two prototype versions of Starpicker were built using Nexpert Object on a 386-SX PC platform. The first prototype was built in the space of about 6 weeks and captured the nominal criteria for star selection. The second prototype required another 6 weeks and implemented a revised control structure. This second version was organized as a hierarchy of computational strategies so that progressively more "desperate" measures could be applied in difficult cases. These prototypes served as a credible proof of concept, but fell short of an operational capability.

The operational version of Starpicker was built using ART-IM on a Sun SPARCstation platform. Development of the operational version of Starpicker required about 15 months and resulted in 4400 lines of ART-IM code and 6800 lines of C code. The ART-IM code comprises 127 rules and 172 functions.

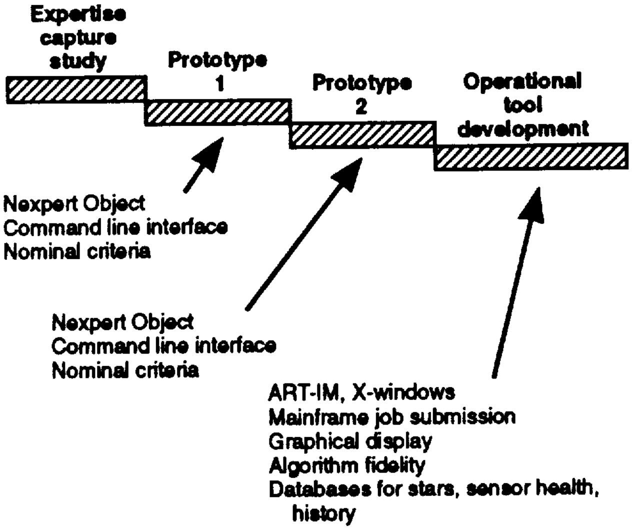

Figure 2. Starpicker development overview

IMPLEMENTATION TECHNIQUES

During the expertise analysis phase of this project it became evident that numerous rules of thumb are used by the expert — for example, estimating the range of glare interference in various situations. A design goal was to avoid discontinuities in the program behavior when a star is found to be just inside or just outside such a range threshold. To do this, a "fuzzy logic" model is used. The glare near the moon, for example, is characterized by a fuzzy region. At one edge of the region a star is considered "certainly unusable," and at the other edge it is "certainly usable." In between, the star is assigned a usability that is between zero and one. (See Figure 3.) Using this technique, the expert's heuristics are represented directly, and the system behavior is not highly sensitive to small variations in the exact values chosen. This formalism was found to be a useful knowledge representation, although only a minimum amount of "fuzzy inferencing" is done in the system.

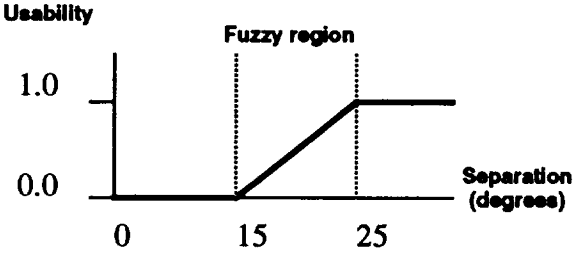

Figure 3. Example of fuzzy transition region — usability of a star in the presence of moon glare

A major concern in the Starpicker design is to prevent combinatorial explosion in the generation of candidate star pairs. To do this, two dynamic lists of stars are maintained, a list of candidate primary stars and a list of candidate secondary stars. At any given time, the currently enabled pair-formation rules generate all admissible pairs using these two lists. Membership in the two lists is gradually augmented until a desired number of pairs has been generated. This process is heuristically organized so that the better pairs are likely to be generated first. The final list of pairs is then ranked based on pair quality.

Star lists are implemented using the dynamic class-membership facilities of ART-IM. The cyclic process of adding stars and generating pairs is implemented with phased rule firings, using the ART-IM rule "salience" mechanism. ART-IM rules are organized into levels of priority or "salience," so that at each execution step the eligible rule with the highest salience is the one that is fired. In Starpicker, a low-salience rule examines the number of pairs generated so far. If more pairs are desired, the next strategy is taken from a list of strategies, appropriate rules are enabled, and the higher-salience rules are allowed to fire again to generate more pairs. Successive strategies from the strategy list will therefore be applied until enough pairs have been generated or the strategy list has been exhausted.

This architecture for the rule base is both easy to understand and easy to use. Changes in the overall problem-solving approach are easily implemented by editing the initial strategy list. This has proven to be a useful vehicle for explaining the implementation to the expert and incorporating his feedback.

A typical strategy list is shown in Figure 4. Two kinds of information are recorded in a strategy list — rule groups and parameter threshold settings. A list item with two elements, such as

```
(strategy dual-sec),
```

denotes a rule group to be enabled. When this strategy is enacted, a fact that enables a selected group of rules is added to the data base. A list item with three elements, such as

```
(pri-thresh -0.1 0.0),
```

is used to control a numeric parameter in the pair generation process. When this strategy is enacted, the specified parameter is progressively stepped (in this case by -0.1) until the specified ending value has been reached (in this case, 0.0). A strategy item of this form may therefore cause several passes through the pair generation rules, one for each iterated parameter value. (Terms in Figure 4 beginning with a question mark are global values defined elsewhere in the code.)

EXPERIENCE

A key factor in the success of this development was the availability of a domain expert who was both supportive of the goals of the project and physically available for consultation. During the development, the domain expert and the principal knowledge engineer were located in the same office area so the knowledge engineer could observe the expert's working practices and quickly resolve questions about the implementation. This close interaction with the expert may have contributed to schedule delays, but the resulting product was significantly improved.

User reaction to Starpicker has been generally favorable. The primary user, the "Star Analyst," uses Starpicker on a regular basis. This individual has extensive experience in the problem domain and has defined many of the current practices. Not surprisingly, therefore, the user does not view Starpicker as a black box for planning solutions. Instead, the user sees Starpicker as a "source of confirmation," since he frequently has a tentative solution in mind before starting to use the tool. He values Starpicker for its convenient access to pertinent information, its "conservative estimates," and the fact that it "doesn't make mistakes."

An important factor in the acceptance of this tool is that its conclusions can be overridden when necessary. The user can also easily update the external data files to reflect experience with new stars and changes in sensor health.

Equally important user feedback comes from individuals who serve as backup Star Analysts in the absence of the primary expert. The reaction from these users has also been generally positive, but it is interesting to note occasional differences in approach. For example, one user states that he is much more willing than others to "push the rules" regarding the star selection criteria. Observing these occasional users, it seems that an on-line help facility would be desirable as an alternative to the written documentation. A tutorial user mode would also be helpful.

Neither the expert nor the occasional users seem inclined to accept Starpicker's recommendations on blind faith. The users prefer to have access to as much supporting information as possible in order to evaluate for themselves the recommendations of the system. A certain degree of subjective judgement appears to go into the final choice from among the available solutions. This judgment process, which has not yet been formalized, trades off such factors as the quality of the stars versus the expected duration of the solution. The users have expressed general satisfaction with Starpicker as both a source of recommendations and supporting information, and it has become a standard resource in day-to-day operations.

CONCLUSIONS

In conclusion, we attribute the success of this program to a combination of programmatic and technical factors. The initial prototyping cycle was useful in defining the concept of the tool, establishing its scope and operation, and providing a convincing demonstration prior to development. ART-IM was found to be powerful, stable, and well suited to this project. Close physical access to the domain expert during the development and the expert's positive and helpful disposition contributed significantly to the quality and usefulness of the final product.

DISCLAIMER

None of the descriptions of commercial software products in this article should be considered an endorsement or criticism by Hughes Information Technology Corporation. These remarks are derived from experience which may or may not be directly transferrable to other applications.

```
(deffacts strategy-list
  (strategy-list
    (strategy nominal)                      ; nominal
    (pri-thresh ?*pri-delta* ?*quality-g*)  ; decrease quality by steps
    (abbrev-limit ?*abbr-delta* ?*abbr-lim*) ; permit abbreviated use
    (strategy dual-sec)                     ; permit dual secondaries during rev
    (strategy relax-sep)                    ; relax separation
    (strategy use-planets)                  ; enable use of planet
    (pri-thresh ?*pri-delta* ?*quality-p*)  ; further decrease quality
    (abbrev-limit ?*abbr-delta* ?*abbr-max*) ; further relax abbrev use
    (pri-thresh ?*pri-delta* 0.0)           ; further decrease quality to zero
    (strategy really-relax-sep) ))          ; relax separation to max
```

Figure 4. Sample strategy list

SESSION SE

Servicing

Servicing

Satellite Servicing in GEO by Robotic Service Vehicle

W. De Peuter and G. Visentin

ESA
Noordwijk, The Netherlands

THIS PAPER WAS NOT SUBMITTED IN TIME FOR PUBLICATION.

250