

N92-233594

Detecting Opportunities for Parallel Observations on the Hubble Space Telescope

Michael Lucks
Space Telescope Science Institute*
3700 San Martin Drive
Baltimore, MD 21218

Abstract

The presence of multiple scientific instruments aboard the Hubble Space Telescope provides opportunities for *parallel science*, i.e., the simultaneous use of different instruments for different observations. Determining whether candidate observations are suitable for parallel execution depends on numerous criteria (some involving quantitative tradeoffs) that may change frequently. This paper presents a knowledge based approach for constructing a scoring function to rank candidate pairs of observations for parallel science. In the Parallel Observation Matching System (POMS), spacecraft knowledge and schedulers' preferences are represented using a uniform set of mappings, or *knowledge functions*. Assessment of parallel science opportunities is achieved via composition of the knowledge functions in a prescribed manner. The knowledge representation, knowledge acquisition, and explanation facilities of the system are presented. The methodology is applicable to many other multiple-criteria assessment problems.

1. Introduction

Despite a well-known manufacturing flaw in its primary mirror, NASA's orbiting

Hubble Space Telescope (HST) has produced images of unprecedented clarity since its launch in 1990 [Kin91]. Repair of the telescope's ability to resolve very faint objects is planned during a maintenance mission in 1993, and demands for observation time on the HST have remained high.

In order to maximize the efficient use of observation time, the Space Telescope Science Institute (STScI) has promoted the research and development of advanced methods for scheduling astronomical observations. Two software systems -- Spike and Transformation -- developed at STScI have applied artificial intelligence techniques toward this end. Transformation [Ger91] is a planning system for the grouping and ordering of observation tasks. Transformation converts observers' requests for spacecraft activities into hierarchical structures called *scheduling units* (SUs) containing multiple sequential tasks that may be subsequently treated as indivisible entities for scheduling purposes. Spike [Mil91, Joh90a, Mil88] is a knowledge-based system for long-range scheduling. Using *suitability functions* [Joh90b] to represent scheduling constraints and preferences, Spike determines week-long segments into which each SU may be scheduled. The output of Spike is later refined into a second-by-second calendar using the Science Planning Scheduling System (SPSS) [Tay91].

The presence of six scientific instruments aboard the HST provides opportunities for *parallel science*, i.e., the simultaneous use of different instruments to observe different targets. By overlapping multiple observations, this concept clearly has the potential to increase throughput. Parallel science is particularly useful for scheduling important exploratory surveys. Without it, such explorations consume considerable resources at the expense of many other shorter and more specific observations. If executed in parallel with other pre-scheduled activities, however, such endeavors may be undertaken at

* Operated by the Association of Universities for Research in Astronomy for the National Aeronautics and Space Administration

opportune times without excessive resource consumption. The parallel science effort at STScI has been a substantial undertaking, involving major extensions to several existing software systems, including Spike, Transformation, and SPSS.

To utilize parallel science, an observer must specifically request that an observation is to be conducted in parallel. It is then the responsibility of schedulers at STScI to find a corresponding non-parallel (or *primary*) observation somewhere on the HST calendar with which the parallel task may be matched. The large number of observations (on the order of 10^4 primaries and 10^3 parallels) and the wide diversity in their requirements makes this a formidable task. This paper describes the Parallel Observation Matching System (POMS), a knowledge-based advisory system embedded into Spike that assists schedulers in finding such matches between primaries and parallels. For each primary SU scheduled by Spike, POMS ranks available parallel SUs according to their compatibility with the requirements of the primary.

Although compatibility between a primary and a parallel depends on certain obvious factors such as instrument constraints (e.g., both tasks must not require the use of the same instrument) and pointing similarity (the two targets must be sufficiently close), it also involves many other more subtle criteria, some of which are quantitative in nature and introduce tradeoffs into the assessment. The advisory system must be able to represent and aggregate the effects of such criteria.

Another salient characteristic of the problem is that the match assessment criteria are likely to be vague, particularly in the early stages of the parallel science project, since parallel scheduling policies have not been firmly established and the full effects of the criteria are not yet well understood. Hence, knowledge is likely to be tentative, incomplete, and subject to frequent change. Ease of incremental

extension and modification of the knowledge base is therefore crucial to the success of the system.

The knowledge representation scheme used in POMS permits the construction of a scoring function for ranking primary-parallel matches. The scoring function is built from modular units of knowledge about individual criteria and from a modifiable aggregation formula that is an explicit part of the knowledge base. The approach has been used previously in a very different application, i.e., a prototype advisory system for selecting mathematical software from numerical subroutine libraries [Luc92]. POMS is the first production level application of the technology. The approach extends existing representation media in its capabilities to express quantitative tradeoffs and complex interactions among multiple criteria [Luc90], while retaining the traditional advantages of expert systems for incremental modification and explanation.

POMS is implemented in the Common Lisp Object System (CLOS) programming language.

The remainder of the paper is organized as follows. Section 2 describes the architecture and high-level functionalities of the system. Section 3 introduces some terminology and notation required to describe the knowledge representation. The assessment criteria are described in Section 4. The knowledge representation, explanation, and knowledge acquisition facilities are presented in sections 5, 6, and 7, respectively. In section 8 we discuss the preliminary results of the POMS project.

2. System Overview

The architecture of POMS is depicted in Figure 1. The three major components of the system are: (1) the parallel database, (2) the parallel knowledge base, and (3) the parallel matcher.

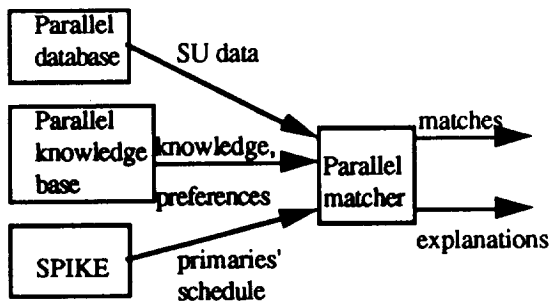


Figure 1. Architecture of POMS

The *parallel database* is a relational database that stores descriptive information about each primary and parallel (e.g., pointing, instrument, and timing requirements). This information is accessible to the matcher when needed at run time.

The *parallel knowledge base* contains the assessment criteria used to evaluate the compatibility between primaries and parallels. The knowledge base was initially constructed by the POMS developers (with input from systems engineers), however it may be modified by users (schedulers).

The *parallel matcher* is the POMS control mechanism. The matcher is invoked after the primaries have been scheduled by Spike to week-long segments, but before these primaries are delivered to SPSS for short-term scheduling refinement. The matcher evaluates the compatibility of each scheduled primary with each available parallel in the database, based on the descriptions of the SUs contained in the database and the scheduling knowledge/preferences contained in the knowledge base. Each primary-parallel pair is assigned a score, and the highest ranked matches for each primary are delivered to SPSS.

Upon request, the matcher also generates explanations of its advice for inspection by users and knowledge engineers.

3. Terminology and notation

The following notation and terminology are used in the subsequent description of POMS.

P is a set of primaries that have been scheduled for a specific week by Spike.

p is a primary SU in P .

Q is a set of parallel SUs to be matched with P .

q is a parallel SU in Q .

$R = \{r_1, r_2 \dots, r_k\}$ is a set of *properties* that characterize primaries and/or parallels (target, time duration, instrument, etc.). Properties may be viewed as functions on $P \cup Q$, e.g.:

if $r_1 =$ "primary-target", $r_1(p)$ returns a list of two elements containing the celestial longitude and latitude of the primary's target;

if $r_2 =$ "parallel-instrument", $r_2(q)$ returns the name of the scientific instrument required by the parallel;

if $r_3 =$ "primary-duration" $r_3(p)$ returns number of seconds allotted to the primary during which parallel science may be conducted;

if $r_4 =$ "primary-week" $r_4(p)$ returns the week to which p has been scheduled.

To avoid ambiguity, properties that apply to both primaries and parallels (e.g., "target" or "instrument") are represented as two distinct properties ("primary-target," "parallel-instrument"). For notation, assume that properties $1, 2, \dots, h$ apply to primaries, and properties $h+1, h+2, \dots, k$ apply to parallels, i.e., $R = r_1, r_2, \dots, r_h, r_{h+1}, r_{h+2}, \dots, r_k$. Property values either exist explicitly in the parallel database (e.g., the primary or parallel instrument), or are determined by the Spike scheduler (e.g., the week to which a primary has been scheduled).

$F = \{f_1, f_2, \dots, f_n\}$ is a set of evaluation criteria or *features* upon which the compatibility between a primary and a parallel is assessed. Examples of features for a primary p and a parallel q are: "pointing" (the targets of P and Q must be sufficiently close to one another), "instruments" (p and q must not both require the use of the same scientific instrument), and "timing" (p and q match best when they each require approximately the same amount of execution time). Note the distinction between features and properties, i.e., properties are characteristics of a primary or a parallel (e.g., the instrument used), while features are characteristics of a match (e.g., whether the instruments used by the primary and the parallel are the same). Property values are required for the evaluation of features (see below).

The *evaluation interval* $L = [L_l, L_h]$ is a sub-interval of the real numbers, where $L_l < L_h$. L_l is the lower bound of L and L_h is the upper bound. L is the range of the scoring function, as well as the range of some of the mappings used in the knowledge base. In the POMS knowledge base $L = [0, 1]$, although this restriction need not hold in general.

The *neutrality point* $L_N \in L$ indicates a "neutral" value in the evaluation interval. This reflects a "moderate" compatibility score (i.e., not particularly compatible, nor particularly incompatible). In POMS, $L_N = 0.5$.

The *scoring function* $H : P \times Q \rightarrow L$ evaluates the compatibility of any primary p with any parallel q . For all $p, p' \in P$ and $q, q' \in Q$, $H(p, q) > H(p', q')$ iff the match between p and q is considered better than the match between p' and q' . Since any p or q is described by properties, we may consider the domain of H to be vectors of property values, i.e., $H(p, q) = H(r_1(p), r_2(p), \dots, r_h(p), r_{h+1}(q), \dots, r_k(q))$.

The *matching score* for a primary p and a parallel q is the value returned by the

scoring function H when applied to a primary-parallel pair.

Feature evaluation is the assignment of a value to a feature for a particular primary-parallel pair. This assignment is an expression of the compatibility between p and q with respect to a single feature. For example, if p and q both require the same instrument, then the feature "instruments" would evaluate to 0, indicating that the pair is incompatible with respect to this feature. If different instruments are required, then "instruments" evaluates to 1 for the pair. For certain features, intermediate values are possible. Some features are *qualitative*, i.e., they are either totally present or totally absent. For example, the instruments used by p and q either are or are not the same. The absence of a single qualitative feature may be sufficient to disqualify p and q from simultaneous execution. Other features are *quantitative*, i.e., they are exhibited to varying degrees, possibly on a continuous scale. The feature "timing", for instance, is associated with a goodness-of-fit measure, i.e., the closer the timing requirements of p and q , the better the match.

4. Matching Criteria

Currently, the POMS knowledge based represents the effects of ten features. Five features ("timing", "priority", "roll", "mechanism-motion", "pcs-mode") are quantitative, and the others ("instruments", "pointing", "nssc-usage", "permits-parallels", and "manual-match") are qualitative.

"Timing" is the degree to which the primary's time available for parallel science matches the time required by the parallel. The most compatible situation occurs when p has slightly more time available than is required by q . If p is much longer than q , then the match is not as good, since the extra time would be better utilized in a match with a longer parallel. If p is much shorter than the q , then the primary and parallel are clearly incompatible. Total incompatibility does

not exist, however, when p is only moderately shorter than q, since the short-term schedulers may have reason to delay the initiation of the next primary, thereby permitting the parallel to finish. "Timing" is a continuous feature that, in general, assumes values intermediate to the stereotyped situations described above.

"Priority" is number indicating the scientific importance of a parallel SU, as determined by a peer review committee and the Director of the STScI. The higher the number, the higher the compatibility of a parallel with any primary.

When both the primary and parallel have fixed point targets (as opposed to target regions -- see below), it is usually necessary to roll the spacecraft (rotate it about an axis parallel to its bore) in order to bring both targets into the fields of view of both instruments. "Roll" is the number of degrees of spacecraft roll required to do this. Since rolling the spacecraft takes time, the greater the roll, the lesser the compatibility.

The HST has a pointing control system (PCS) that stabilizes the spacecraft during observations. The PCS operates in three different modes, depending on the degree of stability required by the observation. The more stable modes require successively greater overhead time. The policy has been adopted that no matches are permitted between parallels whose PCS requirements are more stringent than the primaries, since the additional overhead would delay the pre-scheduled primary. The best case occurs when the primary and parallel have the same PCS requirement. Matches are permitted in cases where p has a stricter requirement than q, however this is less desirable than the above case, since p's strict requirement is "wasted" on a parallel that doesn't really need it. The feature "pcs-mode" reflects these considerations. If p's PCS mode is greater than q's PCS mode, then p and q are incompatible with respect to "pcs-mode". Otherwise, the

greater the difference between the two modes, the lower the compatibility.

The feature "instruments" expresses whether or not the scientific instruments required by p and q are a legal combination for parallel science. Certain instruments are currently precluded entirely, however this is subject to change. If the parallel instrument is allowable and is not the same as the primary instrument, then p and q are totally compatible with respect to this feature.

The feature "pointing" exhibits total compatibility when the targets for p and q are positioned sufficiently close to permit parallel observations. Otherwise, p and q are incompatible with respect to the feature.

Some observations use instruments with moving parts, which causes the HST to vibrate slightly. Other observations cannot tolerate such small vibrations. Obviously, these two types cannot be executed in parallel. The feature "mechanism-motion" exhibits total incompatibility in such cases, and total compatibility otherwise.

An NSSC-1 computer aboard the spacecraft is used to store a variety of temporary data. Since only one instrument at a time may access the computer, p and q are incompatible if they both require its use. The feature "nssc-usage" expresses this criterion.

There are numerous reasons why certain primaries are excluded out of hand from consideration for parallel science. In such cases, the feature "permits-parallels" exhibits total incompatibility.

By setting appropriate fields in the parallel database, the scheduler is permitted to force a match between a particular p and q. In such cases, the highest matching score is assigned to the pair, thereby guaranteeing its delivery to SPSS. In such cases, the feature

"manual-match" exhibits total compatibility.

5. Knowledge representation

The POMS knowledge base contains expertise about the semantics and influences of matching criteria, encoded via a structured set of expert-supplied numerical mappings, or *knowledge functions*. The knowledge functions generate a network, whose traversal implements the application of the knowledge base to a candidate primary-parallel pair. The output of the network is a numerical score that estimates the degree to which the primary and parallel are compatible.

There are four types of knowledge functions, each with a specific representational task:

(1) *measurement functions*, which quantify the degree to which a feature is present in a primary-parallel pair;

(2) *intensity functions*, which normalize the degree of each feature's presence;

(3) *compatibility functions*, which describe relationships between feature intensity and the goodness of a match, with respect to a single feature;

(4) an *aggregation function*, which combines individual feature compatibilities into an overall assessment of the match.

Application of the knowledge base to a primary and a parallel is achieved via the composition of the knowledge functions in a prescribed manner. Figure 2 depicts how this composition may be viewed as a traversal of a network in which the arcs are knowledge functions and the nodes are function inputs/outputs. Processing of the knowledge base corresponds to traversal of the network from bottom to top. The inputs to the network are property values for a primary p and a parallel q . The n measurement functions accept these inputs and return n

measurement values. The intensity functions accept the measurement values and return n *intensity values*. Compatibility functions accept intensity values and return n *compatibility values*. Finally, the compatibility values are mapped into a single number $H(p,q)$ at the figure's top.

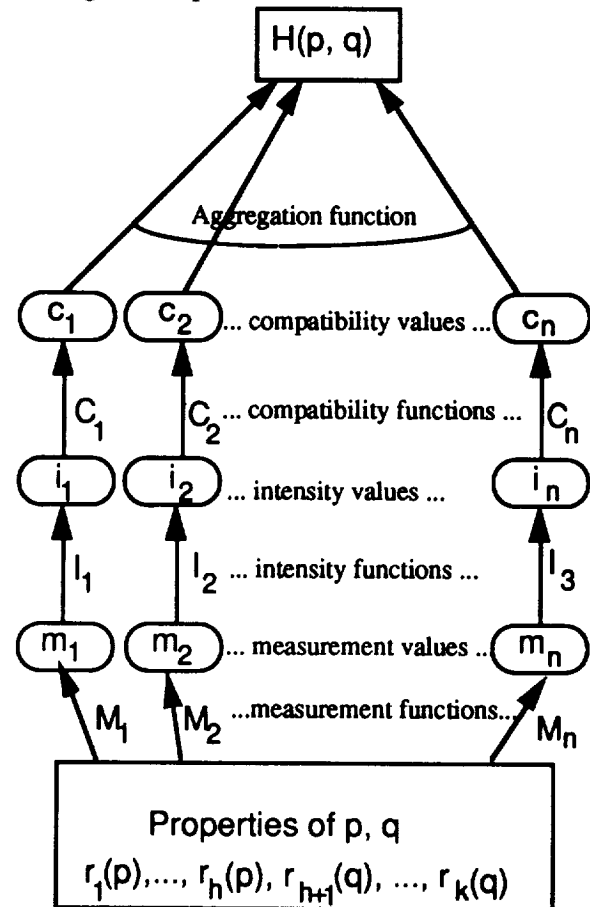


Figure 2. Network model of scoring function H for primary p and parallel q

5.1 Measurement functions

For each feature f_j , there is a measurement function $M_j: P \times Q \rightarrow S_j$ where S_j is the (feature-dependent) range of M_j . M_j is a procedure to measure f_j . The inputs to M_j are property values and the output is a measurement value m_j that expresses (in feature-dependent units) the presence of f_j in a primary-parallel pair. The measurement function for $f_j =$ "instruments", for instance, returns T or NIL , depending on whether the

instruments used by the primary and parallel are different or the same. As an example for a quantitative feature, M_j for the feature $f_j = \text{"timing"}$ accepts two inputs: t_q , the length of time (in seconds) required by the parallel for its execution, and t_p , the number of seconds available for parallel science beginning with the start of the primary. t_p and t_q are property values stored in the parallels database. M_j returns a measurement value $m_j = t_p - t_q$, i.e., the difference between the length of time available and the length of time required for a primary-parallel pair.

In certain cases, the measurement function involves more complicated calculations, e.g., the measurement function for the feature "roll" might compute the number of degrees of roll required by the spacecraft to bring both targets into view. The criteria and methods used for measuring features are chosen by the domain expert.

5.2 Intensity functions

For each feature f_j , there is an intensity function $I_j : S_j \rightarrow L$, where S_j is the range of the measurement function M_j . I_j normalizes the measurement value m_j into a uniform scale. In POMS, intensity functions are defined as sets of points provided by the domain expert. For qualitative features, the mapping is frequently trivial, i.e., the measurement value is either 0 (signifying total absence of f_j) or 1 (signifying total presence). For example, if $f_j = \text{"instruments"}$ (as described in the previous section), I_j simply maps $S_j = \{T, NIL\}$ into $\{0,1\}$ (Table 1). For quantitative features that assume continuous values, the ordered pairs generate a piecewise linear function where additional points may be defined to arbitrarily fine gradation at the discretion of the domain expert. In such cases, a "neutral" intensity value $i_j = 0.5$ indicates moderate presence of f_j . For example, intensity for the feature $f_j = \text{"timing"}$, refers to the degree to which the time available exceeds the time required, i.e., if p is a very short primary SU and q is a

very long parallel SU, then the feature timing is considered to be weakly present in the pair. Conversely, "timing" is strongly evidenced when a very long primary is matched with a very short parallel, and the feature is neutral when the time available is equal to the time required. This behavior is represented in a table of ordered pairs defining I_j (Table 2). Domain values not explicitly represented in the table are derived by linear interpolation (e.g., $I_j(200) = 0.40$). Since linear interpolation requires finite values, the infinite values at the extremes must be compromised by finite approximations -- in this case 100,000 and -100,000 -- beyond which the function value will not change. The particular data points chosen by the expert are somewhat arbitrary and represent an approximation to the expert's interpretation of the feature's semantics.

Note that intensity is a function only of the feature, not the goodness of a match, i.e., high intensity does not necessarily imply that the primary and parallel are well-matched with respect to the feature. For example, neutral intensity for "timing" (i.e., a situation where the primary and parallel have the same timing requirements) is a more compatible situation than high intensity (i.e., a situation where the time available from the primary greatly exceeds the time required by the parallel). This is illustrated in the following section.

Table 1. Intensity function for feature $f_j = \text{"instruments"}$

m_j	$i_j = I_j(m_j)$
NIL	0.0
T	1.0

Table 2. Intensity function for feature $f_j = \text{"timing"}$

m_j	$i_j = I_j(m_j)$
$-\infty$	0.00
-100,000	0.00
-10,000	0.03
-1,000	0.20
-400	0.30
0	0.50
400	0.70
1,000	0.80
10,000	0.97
100,000	1.00
$+\infty$	1.00

5.3 Compatibility functions

For each feature f_j , there is a compatibility function $C_j : L \rightarrow L$. C_j represents the compatibility between a primary and a parallel with respect only to feature f_j , as a function of the feature's intensity. The input to C_j is the intensity value i_j . The compatibility value $c_j = C_j(i_j)$ represents the goodness of the match between p and q with respect to f_j . Like intensity functions, compatibility functions are represented as sets of ordered pairs. Table 3 shows the definition of C_j for the feature $f_j = \text{"instruments"}$. When the feature is absent in the primary-parallel pair (i.e., $i_j = 0$, denoting that the pair uses the same instrument), the pair is incompatible with respect to "instruments". If the feature is present with maximum intensity (i.e., i_j

$= 1.0$, denoting that p and q use different instruments), then the pair is judged to be totally compatible ($c_j = 1.0$) with respect to the feature. Table 4 shows a more complicated compatibility function for the quantitative feature "timing".

Table 3. Compatibility function for feature $f_j = \text{"instruments"}$

i_j	$c_j = C_j(i_j)$
0.0	0.0
1.0	1.0

Table 4. Compatibility function for feature $f_j = \text{"timing"}$

i_j	$c_j = C_j(i_j)$
0.0	0.0
0.4	0.3
0.5	0.45
0.55	0.5
0.6	0.7
0.8	0.5
0.9	0.42
1.0	0.25

Here C_j encodes the belief that the optimal compatibility (0.7) is achieved when the intensity is moderately high (0.6). This reflects a situation where the time available is greater, but not too much greater, than the time required. Compatibility declines for higher values because the extra time available would be wasted (and used better with a longer parallel). Situations where the time available is less than the time required are not considered incompatible because SPSS may choose to delay the initiation

of the next primary, in which case the parallel may continue to completion. The compatibility declines sharply as the negative timing disparity increases, however.

The precise values chosen by the expert to define quantitative compatibility functions are not arbitrary. The method for defining these mappings is described in section 7.

5.4 Aggregation function

The knowledge base contains a single aggregation function $A : L^n \rightarrow L$ whose purpose is to combine the compatibility values c_1, c_2, \dots, c_n into an overall matching score for a primary-parallel pair. The aggregation function is not a predetermined formula, but rather it is defined by the expert as an explicit part of the knowledge base. It depends on the evaluation features, and may be changed at the expert's discretion. To capture the particular semantics of different features, the aggregation function is built up from operators called *aggregation primitives*, MIN, MAX and α . The primitive operators reflect three different modes of aggregation between features.

For simplicity, we describe each primitive as a binary operator, but since all three are associative and commutative, they may be generalized in a straightforward manner to n-ary functions whose arguments may be evaluated in any order. Each primitive accepts two compatibility values c_i and c_j as arguments and returns a value equal to the aggregate effect of c_i and c_j . The semantics of the three operators are shown in Table 5. MIN simply returns the minimum of c_i and c_j , while MAX returns the maximum. MIN and MAX represent cases where the compatibility of either feature always dominates the aggregation, e.g., if $c_i \leq c_j$, then $\text{MIN}(c_i, c_j) = c_i$, regardless of the precise value of c_j , i.e., no tradeoffs are exhibited. α is used to express tradeoffs and compensations, i.e., neither feature dominates and the aggregate effect

depends on the precise level of each feature.

An example of MIN occurs between the qualitative features $f_i = \text{"instruments"}$ and $f_j = \text{"permits-parallels"}$. If a candidate primary-parallel pair is incompatible with respect to instruments, (i.e., $c_i = 0$) then the aggregate effect is always incompatible, whether or not the primary permits parallel science. Similarly, if parallel science is not permitted on the primary (i.e., $c_j = 0$), then the aggregate effect is always incompatible, whether or not the instruments are compatible. These semantics are captured by $\text{MIN}(c_i, c_j)$. Note that for qualitative features, MIN is equivalent to the logical conjunction (i.e., overall compatibility requires individual compatibility from both features), hence this mode of aggregation is called *conjunctive*.

An example of MAX occurs between $f_i = \text{"instruments"}$ and $f_j = \text{"manual-match"}$. Since "manual-match" overrides all other features, if a user requests a manual match between a primary-parallel pair (i.e., $c_j = 1$), then the aggregate effect is always totally compatible, regardless of "instruments" compatibility. (Obviously, this feature assumes some special knowledge on the part of the user, and is not to be used carelessly.) These semantics are captured by $\text{MAX}(c_i, c_j)$. Note that for qualitative features, MAX is equivalent to the logical disjunction (i.e., compatibility from either feature implies aggregate compatibility), hence this mode of aggregation is called *disjunctive*.

An example of α occurs between the features "timing" and "priority". Regardless of how compatible the candidate pair is with respect to timing, the aggregate effect may be raised (lowered) by the influence of a high (low) priority. The influence of "priority" is similarly modified by the effect of timing compatibility. A significant property of α is that neutral compatibility in either feature has no effect on the aggregation, i.e., $\alpha(c_j, L_N) = \alpha(L_N, c_j)$ for all c_j . This

mode of feature interaction is called *compensatory*.

The formula used to model these semantics depends on the choice of the evaluation interval L and the neutrality point L_N . In POMS ($L = [0,1]$ and $L_N = 0.5$), we use $\alpha : L \rightarrow L$ defined as

$$\alpha(c_i, c_j) = \frac{c_i c_j}{c_i c_j + (1 - c_i)(1 - c_j)}$$

α is a special case of a symmetric sums operator [Sil79]. Its properties are discussed in [Luc90]. This definition for α is not unique, i.e., other formulas might yield qualitatively similar results. See [Che88] for a formal mathematical treatment of related families of aggregation operators. Note that α creates a zero-divide condition when one of the inputs is 0 and the other is 1. To avoid this possibility, compatibility functions that return a value of 1 for any finite measurement value are not allowed for compensatory features. This restriction reflects an assumption that no single compensatory feature is sufficient to dominate the matching process.

The aggregation primitives constitute a simple language for expressing complex interactions among multiple features. In principle, additional operators may be added to the language, although MIN, MAX, and α have thus far sufficed.

Table 5. Behavior of aggregation primitives

c_i	c_j	MAX	MIN	α
high	high	high	high	very high
high	low	high	low	moderate
low	high	high	low	moderate
low	low	low	low	very low

The aggregation function A is defined as a composition of aggregation primitives.

This process may be visualized as a parse tree in which the leaves are compatibility values and the internal nodes are aggregation primitives. Figure 3 is an example of an aggregation parse tree for the features $f_1 =$ "manual-match", $f_2 =$ "permits-pars", $f_3 =$ "priority" and $f_4 =$ "timing". Here $A = \max(c_1, (\min(c_2, \alpha(c_3, c_4))))$. The aggregate effect of "priority" and "timing" is compensatory, and their combined effect is conjunctive with "permits-parallel". Finally, the aggregate effect of the three lower features is disjunctive with "manual-match". For example, if:

- (1) a manual match is not requested (i.e., $c_1 = 0$);
- (2) the primary permits parallel science (i.e., $c_2 = 1$);
- (3) the parallel has been assigned a high priority (e.g., $c_3 = 0.8$);
- (4) the timing compatibility is fairly low (e.g., $c_4 = 0.3$);

then the matching score for a candidate pair displaying these features is $\text{MAX}(0, \text{MIN}(1, \alpha(0.8, 0.3))) = 0.63$.

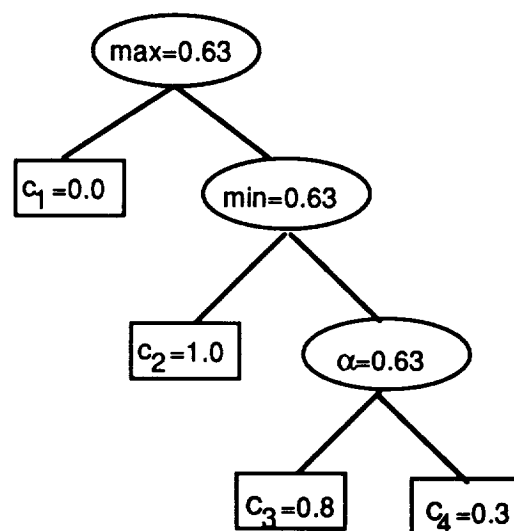


Figure 3. Example of aggregation function parse tree. $A(c_1, c_2, c_3, c_4) = 0.63$

The full aggregation function (expressed in the n-ary format) currently used in the POMS knowledge base is:

$MAX(c_{10}, MIN(c_6, c_7, c_8, c_9, \alpha(c_1, c_2, c_3, c_4, c_5)))$,

where c_1 = "timing", c_2 = "priority", c_3 = "roll", c_4 = "mechanism-motion", c_5 = "pcs-mode", c_6 = "instruments", c_7 = "pointing", c_8 = "nssc-usage", c_9 = "permits-parallels" and c_{10} = "manual-match". This formula reflects that (1) c_1 through c_5 are compensatory with respect to one another; (2) c_6 through c_9 and the aggregate of c_1 through c_5 are mutually conjunctive; and (3) c_{10} is disjunctive with all other features.

6. Explanation

Upon request, the matcher generates tables that explain its assessments. Each table summarizes the reasoning that went into the matcher's analysis of the compatibility between a particular primary-parallel pair. An example (reformatted) is shown in Table 6. The header of the table contains the unique SU identification numbers for p and q, and the matching score. In this case, the score of 0.565 indicates that p and q are moderately compatible.

Each row in the table corresponds to the evaluation of a particular feature. The first column contains the feature's name. The second and third columns contain relevant property values for the primary and the parallel, respectively. The primary column in row 1, for example, shows the number of seconds available for parallel science for p (9700), and the parallel column shows the time required by q (700 seconds). For legibility, the name of the property and the units of measurement are not listed in the table, but they are understood by schedulers who are familiar with POMS.

Columns 4, 5 and 6 contain the measurement, intensity and compatibility values, respectively, for each feature. Recalling the "timing" measurement

function (Section 5.1), the measurement value in row 1 is the difference between the time available (column 2) and the time required (column 3), or 9000 seconds. By Table 2, this yields an intensity value of 0.95, indicating that there is quite a lot of excess time available. POMS recognizes (via the "timing" compatibility function, Table 4) that this large excess is undesirable and assigns a low "timing" compatibility value (0.33) to the primary-parallel pair.

Table 6. Explanation of match assessment

Primary 0091401 Parallel 0091505 Score = 0.556					
Feature	Primary	Parallel	Mea	Int	Com
timing	9700	700	9000	0.95	0.33
priority	-	3.8	3.8	0.70	0.62
roll	131.4	0, 360	0	0.0	0.50
mech-motion	Y, Y	N, Y	1	0.75	0.53
pcs-mode	fine	fine	0	1.0	0.58
instruments	WFPC	FOC	T	1.0	1.0
pointing	2.0, 6.0	r,9.0,5.0 8.0,2.0	T	1.0	1.0
nssc-usage	N	N	N	0.0	1.0
permits-parallels	Y	-	T	1.0	1.0
manual-match	-	-	N	0.0	0.0

For the next feature, "priority", the only significant property is the scientific priority of the parallel, 3.8. No primary properties are relevant, hence the primary column for "priority" is blank. The measurement function for "priority" simply returns the priority property value, hence it is also equal to 3.8. Priorities are assigned on a scale of 1 to 5, hence 3.8 is considered to be a fairly important observation, i.e., the intensity value for "priority" is high (0.70). This high intensity is viewed by POMS as having a

fairly strong positive influence (compatibility value = 0.62).

The primary column for "roll" contains the orientation of the spacecraft (in degrees, relative to an HST-specific coordinate system) that is intended for observing p. In this case, this so-called "nominal orientation" is equal to 131.4 degrees. The parallel column contains an orientation interval that is sufficient to view both the primary and the parallel. In this case, the interval includes all possible orientations, i.e., 0 to 360 degrees. (Any orientation is sufficient because the parallel has a *region target* that contains the primary target. See "pointing", below). The roll measurement function computes the minimum number of degrees that the spacecraft must rotate from the nominal orientation, in order to reach the desired interval. In this case, the nominal orientation is within the desired range already, hence no roll is required. This fact is recorded in the "roll" measurement value (0 degrees). For "roll", this minimal intensity implies neutral compatibility, i.e., the feature will have no effect on the overall assessment (compatibility value = 0.5).

The primary column for "mechanism-motion" refers to two properties: (1) whether or not the primary requires mechanism motion ("Y indicates motion is required); and (2) whether or not the primary can tolerate motion ("Y" means "yes"). The same scheme is used for the q, hence the parallel column for "mechanism-motion" indicates that the parallel can tolerate motion, but does not require it. The measurement function for "mechanism motion" recognizes this situation and returns a tag value of 1 to the measurement value. (The simple tagging scheme for measuring this feature and its related intensity function are not described here.) This combination of motion requirements is considered to be mildly favorable, hence a slight positive compatibility value (0.53) is assigned.

Both p and q require the strictest PCS mode. This is a favorable situation, hence

"pcs-mode" exhibits a compatibility value of 0.58.

The remaining features are qualitative. p and q use different instruments (the wide-field planetary camera (WFPC) and the faint object camera (FOC)). Neither uses the NSCC-1 computer, while the primary permits parallel observations, and no manual match has been specified. Hence, the conjunctive features "instruments", "nssc-usage" and "permits-parallels" each exhibit maximal compatibility, and the disjunctive feature "manual-match" exhibits minimal compatibility.

The primary column for the feature "pointing" contains the celestial coordinates (2.0 degrees longitude, 6.0 degrees latitude) of the primary target. The parallel properties indicate that q has a *region target*, i.e., a region of space in which any specific pointing is sufficient. (Region targets are quite common for parallels.) The parallel column describes this region as a rectangular area ("r") centered at 9.0 longitude and 5.0 latitude, with a longitudinal extent 10.0 degrees from center, and a latitudinal extent 2.0 degrees from center. Since the primary pointing is contained well within the parallel's target region, p and q are compatible with respect to pointing. (This also explains why any orientation of the spacecraft is acceptable.)

Application of the POMS aggregation function (Section 5.4) yields a slightly positive matching score of 0.556, despite the poor time-fit. POMS has concluded that the timing problem is outweighed by the combined positive influences a high scientific priority, favorable mechanism motion and pointing control requirements.

7. Knowledge acquisition

In this section, we describe the methods used in POMS for acquiring new expertise and for modifying an existing knowledge base.

7.1 Adding new knowledge

The process of acquiring new knowledge in POMS is relatively structured, compared to conventional knowledge-based systems where acquisition usually requires informal (and often lengthy) dialogs between the domain expert and the knowledge engineer. New knowledge in POMS always comes in the form of a new feature to be added to the assessment process. Support for the new feature requires the expert to provide definitions for the new measurement, intensity and compatibility functions, and to extend the existing aggregation function to include the new feature. To illustrate this process, assume that a new feature f_{n+1} is being added to a knowledge base containing f_1, f_2, \dots, f_n .

First, the domain expert selects a name for f_{n+1} , decides on a procedure for measuring the new feature, and determines what SU properties are required for the analysis. The knowledge engineer then implements the procedure in CLOS. A pointer to this code is added to the knowledge base so that it is invoked whenever the new measurement function M_{n+1} is applied.

For the intensity function I_{n+1} , the expert provides a set of ordered pairs as in Section 5.2. For qualitative features, this is straightforward. For quantitative features, the function should reflect the expert's intuitive understanding of the feature's semantics.

Extension of the aggregation function A to incorporate f_{n+1} requires the addition of a new branch and leaf (and possibly a new internal node) to the existing parse tree for A . The expert is requested to identify the modes of interaction between the new and existing features. Based on this analysis, the placement for the new leaf c_{n+1} is identified. (For difficult cases, this ad hoc extension technique may be assisted by a partially mechanical procedure [Luc90].) The engineer then makes a corresponding change in the

formula that implements the aggregation function in the knowledge base.

To elicit a new compatibility function C_{n+1} for f_{n+1} , a context is constructed in which the levels of the existing features are constrained such that the new feature's effect completely dominates the aggregate compatibility of f_1, f_2, \dots, f_n . Under these assumed constraints, $A(c_1, c_2, \dots, c_n, c_{n+1})$ is exactly equal to $c_{n+1} = C_{n+1}(i_{n+1})$, for all i_{n+1} (see below). Hence, in the assumed context, $A = C_{n+1}$. To illustrate such a context, we consider the case where all features are compensatory, i.e., where $A(c_1, c_2, \dots, c_{n+1}) = \alpha(c_1, c_2, \dots, c_{n+1})$. In this case, the desired context is achieved by assuming that the aggregate effect of f_1, f_2, \dots, f_n is neutral, i.e., $A(c_1, c_2, \dots, c_n) = \alpha(c_1, c_2, \dots, c_n) = L_N$. Under this assumption, $A(c_1, c_2, \dots, c_{n+1}) = c_{n+1} = C_{n+1}(i_{n+1})$, for all intensity values i_{n+1} , i.e., the aggregate compatibility of f_1, f_2, \dots, f_{n+1} is completely dominated by f_{n+1} . This equality, which is easy to verify formally, is consistent with the intuitive interpretation that neutral compatibility has a neutral aggregative effect among compensatory features.

Once the context has been established, the expert is then asked to estimate the matching score for various selected intensity values i_{n+1} , with the other features fixed at their assumed levels. Since $A = C_{n+1}$, each matching score is equivalent to a compatibility value c_{n+1} . Hence, each $\langle i_{n+1}, \text{matching score} \rangle$ pair is equivalent to a $\langle i_{n+1}, c_{n+1} \rangle$ pair. The set of such pairs becomes the new compatibility function C_{n+1} .

The rationale for the above strategy is that it yields the expert's opinion of the direction and degree to which the new feature displaces the neutral effects of the other features. This "strength of displacement" is the essential heuristic used to estimate a feature's significance to the overall aggregation.

Consider a simple case in which $f_2 = \text{"priority"}$ is being added to the

existing knowledge base which contains the single feature $f_1 = \text{"timing"}$. The aggregation function is $A(c_1, c_2) = \alpha(c_1, c_2)$ and the context contains the assumption that $c_1 = 0.5$. The expert is asked to score a match in which timing is neutrally compatible, i.e., $c_1 = 0.5$ for varying intensities of i_2 for "priority". If "priority" is present with minimum intensity (i.e., $i_2 = 0$, the lowest possible priority), then the aggregate matching score is very low, say 0.1. This assessment states the expert's opinion that very low priority significantly degrades the neutral effect of "timing". Some other feature might have a less significant effect. At neutral intensity for "priority" (i.e., $i_2 = 0.5$), the matching score might be 0.5, implying that a moderate priority has neither a positive nor negative effect on the matching score. The highest priority (i.e., $i_2 = 1.0$) might create an aggregate score of 0.8, representing the opinion that high priority has a strong positive effect on compatibility. The data pairs $[(0, 0.1), (0.5, 0.5), (1.0, 0.8)]$ are included in the new compatibility function C_2 . Additional points may be provided to whatever granularity is deemed necessary by the domain expert.

The assumed context of aggregate neutrality described above is appropriate only if all features are compensatory. Different assumptions are required for non-compensatory features. If f_1, f_2, \dots, f_n is conjunctive with respect to f_{n+1} , then the aggregate effect of f_1, f_2, \dots, f_n is assumed to be totally compatible (i.e., $A(c_1, c_2, \dots, c_n) = 1.0$). In this case, $A(c_1, c_2, \dots, c_{n+1}) = \text{MIN}(A(c_1, c_2, \dots, c_n), c_{n+1}) = \text{MIN}(1.0, c_{n+1}) = c_{n+1}$. Hence, the aggregate effect of f_1, f_2, \dots, f_{n+1} is dominated by f_{n+1} , as desired. In this case, the matching scores provided by the expert represent the new feature's strength to displace the total compatibility of f_1, \dots, f_n .

Disjunctive features are handled by assuming that f_1, \dots, f_n are present with minimal compatibility, and the matching score represents the new feature's

strength in displacing the minimal compatibility of f_1, \dots, f_n .

In the general case, where all three modes of interaction may be present in f_1, \dots, f_n , the context requires all three forms of assumptions. For example, suppose that the feature $f_4 = \text{"timing"}$ is added to an existing knowledge base containing $f_1 = \text{"manual-match"}$, $f_2 = \text{"permits-parallels"}$ and $f_3 = \text{"priority"}$. The modes of interaction among these features is shown in Figure 4. The context for defining C_4 consists of the following assumptions:

- (1) f_3 exhibits neutral compatibility (the parallel has moderate priority, i.e., $c_3 = 0.5$);
- (2) f_2 exhibits total compatibility (the primary permits parallel science, i.e., $c_2 = 1$);
- (3) f_1 exhibits total incompatibility (a manual match is not requested, i.e., $c_1 = 0$).

It is easily verified that under these assumptions $A(c_1, c_2, c_3, c_4) = C_4(i_4)$, hence by describing the overall compatibility with variances in the intensity of f_4 , the expert expresses a compatibility function for f_4 .

The method described above is a heuristic for approximating compatibility functions. The degree to which the defined function actually represents the true effects of a feature depends on how closely the feature's general behavior is modeled by its behavior under the assumed constraints. In certain cases, more restrictive assumptions (such as assigning specific measurement values to certain features) are necessary in order for the context to make sense to the domain expert [Luc90]. Other heuristics are possible, but in practice the above strategy has worked satisfactorily.

7.2 Changing existing knowledge

Any of the existing knowledge functions may be modified incrementally. Most

commonly, changes are made to compatibility functions in order to fine-tune the relative effects of features. Suppose, for example, that the scheduler, upon review of many matches, decides that POMS is underscoring matches in which p has a large excess of time available for parallel science (as in the example of Section 6). This deficiency is easily addressed by dampening the negative values defined for such cases in the "timing" compatibility function. For instance, the last entry in the function definition (Table 4) might be adjusted from (1.0, 0.25) to (1.0, 0.30), thereby lessening the worst case effect of the feature. The second-from-last entry might also be modified slightly from (0.9, 0.42) to (0.9, 0.45). Using this modified compatibility function for "timing", the example in Section 6 would yield a matching score of 0.602, somewhat higher than the original score of 0.556. Changes to intensity functions (reflecting a reassessment of a feature's semantics) are made in a similar manner.

Currently, such changes are made by manually editing a file that contains the knowledge base, although an interface is planned that will permit such changes to be made interactively by the scheduler.

Changes in the way features are believed to interact are made by rewriting the aggregation formula in the knowledge base file. Changes to a feature's measurement function usually involves the recoding of the CLOS procedure that implements it. This requires intervention by Spike system developers.

Features may be deleted from the knowledge base simply by removal from the argument list of the aggregation function.

8. Results

Since the HST parallel science program is still in the early testing stage, it is too early to make conclusive statements about the performance of POMS. Currently the system has been used successfully to

verify matches made by human schedulers. POMS has also been employed to analyze the frequency of good matches in a large pool of proposals to be executed in 1992. Although much improvement is required before POMS can assume a more autonomous role in scheduling, the system's rate of improvement has been highly encouraging thus far.

The most apparent strength of the system in this early phase (during which it has been subjected to frequent changes in the matching criteria) has been its capability for incremental refinement. The explanation facility has been quite useful in identifying assessment errors and these errors have, in most cases, been easily corrected by adjusting a compatibility function. Furthermore, the corrections have, in almost all cases, been made without destroying the prior integrity of the knowledge base, i.e., without invalidating previously correct assessments. This same amenability to local refinement was also observed in the previous application of the technique [9,10] and seems to be a generic advantage of the approach. As the human knowledge sources for POMS become more familiar with the parallel science problem, we expect that POMS will be able to represent and use this expertise in an accurate fashion.

There is no particular dependence between the POMS methodology and the parallel science problem. Hence, if successful in the present arena, the knowledge representation scheme should be applicable to other problems involving the assessment of multiple quantitative criteria. We are currently investigating several potential applications, including the detection of duplicate scientific requests in the HST proposal pool, and the matching of scientific observations to point spread functions for image restoration.

Acknowledgements

The parallel science evaluation criteria were originally identified by Mark Johnston of the STSci Science Engineering and Scheduling Division. Jim Mainard, John Baum, John Isaacs and Brian Ross were instrumental in developing the POMS knowledge base. The author is particularly grateful to Glenn Miller, Jeffrey Sponsler, Tony Krueger and Mark Giuliano for many useful suggestions and for their careful readings of the draft.

References

- Cheng Y. and Kashyap R. L. (1988). An Axiomatic Approach for Combining Evidence from a Variety of Sources. *Journal of Intelligent and Robotic Systems*, 1, 17-33. [Che88]
- Gerb A. (1991). Transformation Reborn: A New Generation Expert System for Planning HST Operations, *Telematics and Informatics*, to appear. [Ger91].
- Johnston, M., Miller, G., Sponsler, J., Vick, S., and Jackson, R. (1990). Spike: Artificial Intelligence Scheduling for Hubble Space Telescope". In S.L. O'Dell (ed.). *Proceedings of the Fifth Conference on Artificial Intelligence for Space Applications* (pp. 11-18). NASA Conference Publication 3073. [Joh90a].
- Johnston, M. (1990). SPIKE: AI Scheduling for NASA's Hubble Space Telescope. In *Proceedings of the Sixth IEEE Conference on Artificial Intelligence Applications* (pp. 184-190). Los Alamitos, California: IEEE Computer Society Press. [Joh90b]
- Kinney, A. L. and Blades, J. C. (Eds.). (1991). *The First Year of HST Observations*. Space Telescope Science Institute, Baltimore, Maryland. [Kin91].
- Lucks, M. and Gladwell, I. (1992). "Automated Selection of Mathematical Software, *ACM Transactions on Mathematical Software*, 18 to appear. [Luc92].
- Lucks, M. (1990). A Knowledge-Based Framework for the Selection of Mathematical Software, Ph. D. dissertation, Dept. of Computer Science, Southern Methodist University. [Luc90].
- Miller, G., Johnston, M., Vick S., Sponsler, J., and Lindenmayer, K. (1988). Knowledge Based Tools for Hubble Space Telescope Planning and Scheduling: Constraints and Strategies. *Telematics and Informatics*, 5, 197-212. [Mil88].
- Miller, G. and Johnston, M. (1991). Long Range Science Scheduling for the Hubble Space Telescope, *Telematics and Informatics*, to appear. [Mil91]
- Silvert, W. (1979). Symmetric Summation: A Class of Operations on Fuzzy Sets. *IEEE Transactions on Systems, Man and Cybernetics* 9(10), 657-659. [Sil79]
- Taylor, D. K., Reinhard, K. E., Lanning, H. H. and Chance, D. R. (1991). The Scheduling of Science Activities for the Hubble Space Telescope. In A. L. Kinney, J. C. Blades (Eds.). *The First Year of HST Observations* (pp. 281-287). Space Telescope Science Institute, Baltimore, Maryland. [Tay91]