

Rodney Graham
6.871
12 May 2005

Final Project Report: Film Recommendation System

System Overview

The domain chosen as a focus for implementing a knowledge-based system was that of film recommendation. Online websites such as Netflix and Amazon.com both employ recommendation systems that seek to provide users with recommendation for films based on their personal preferences. The systems employed by these sites served as the inspiration for the system described here. They function by maintaining large databases of previously purchased films and/or user-supplied ratings for previously viewed films. Given this collection information, the systems are able to recommend films to a user, which were favorably rated by other people with a similar rating and/or purchase history. The goal for this project was to design a system that performs a similar function, but that is a knowledge-based application system.

The original proposed task of the system was to recommend films to a person based on a detailed demographic profile and a wide-range of personality traits. That is, the envisioned system would take a detailed description of an individual and translate that into film recommendation by way of a model detailing how various characteristics influence a person's film preferences. Unfortunately, there appears to be no publicly available information concerning such a model of an individual's film preferences. Any such model information, if available, would seem to be privately held by individual film studios. As for publicly available information, much of it consists of mere speculation or collections of anecdotal testimony without any concrete research.

This fact required that the scope of the task for the system be scaled back to match the level of film preference information available. Information relating the demographic variables of race, gender, and age as well as the personality characteristic of personal fantasies was found. However, the incompleteness of the information relating race to film preferences led to the use of only *gender*, *age*, and *fantasies* as the variables defining the profile of a user.

- The variable **gender** describes the gender of an individual; it takes the value of either *male* or *female*.
- The variable **age** describes the age of an individual in years; it takes a positive numeric value.
- The variable **fantasy** describes the personal fantasy reflected in films to which an individual responds most strongly; it takes one of the following values: *adventure*, *hero*, *triumph of the underdog*, *romance*, *being loved*, and *music*.

While the lack of variables in the user profile does somewhat limit the effectiveness of the system, the nature of the problem of recommending a film, as will be discussed in more detail later, still allows for the development a relatively useful film recommendation system.

The system is implemented as a rule-based system using the Allegro Common Lisp interpreter and Joshua. The knowledge base consists of a collection of rules encoding knowledge concerning film preference as well as a small database of film information. The database consists of various properties (e.g. genre, actors) of thirteen films. It is from this set of films that the system chooses films to recommend. The set of films was chosen for variety and to thoroughly exercise the system. Despite the small amount of films, there are enough to demonstrate the functionality of the system. The films contained in the database are listed below.

- **Database of Films:** *Forrest Gump, Star Wars, The Godfather, Gigli, Broadcast News, Todo Sobre Mi Madre, Gone with the Wind, The Sound of Music, Casablanca, ¡Atame!, Jurassic Park, Pulp Fiction and Speed*

Example Interaction

An example interaction with the system is presented here. For a user Jane Doe, a 22-year-old female who responds most strongly to the fantasy of the “triumph of the underdog”, the system recommends the films *Broadcast News, Todo Sobre Mi Madre, Gone with the Wind, Casablanca* and *¡Atame!*.

- **Input:** gender=female, age=22, fantasy=triumph of the underdog
- **Output:** Broadcast News, Todo Sobre Mi Madre, Gone with the Wind, Casablanca, ¡Atame!

To begin interaction with the system, after starting the Lisp Listener and loading the film recommendation knowledge base, the user asks for a film recommendation for Jane Doe by entering the following command:

- (ask [film-recommendation jane-doe ?x] #'print-answer-with-certainty)

This prompts the system to find all films recommendable to the individual `jane-doe`, printing the certainty of the recommendation for each. (The certainty factor for all rules, and thus all recommendations, is 1.0 by default as not enough information was available to make a more meaningful determination of the certainty of the various rules. Nevertheless, the vast majority of the rules would probably have a certainty factor close to 1.0.) The top-level rule of the knowledge base encodes the criteria necessary for a film to be recommended. It is presented below in code, a higher-level representation, and “plain English” with the understood subject of predicates to be the person for whom the system is making a film recommendation.

Code

```
(defrule film-recommendation (:backward :certainty 1.0 :importance 90)
  if   [and [is-film ?film yes]
        [or [genre-film-recommendation ?who ?film]
            [fantasy-fulfillment-film-recommendation ?who ?film]]
        [gender-film-recommendation ?who ?film]
        [critical-opinion-film-recommendation ?who ?film]]
  then [film-recommendation ?who ?film])
```

Higher-level

```
if    Film [FILM] = YES and
      (Genre Film Recommendation = [FILM] or
       Fantasy Film Recommendation = [FILM]) and
       Gender Film Recommendation = [FILM] and
       Critic Film Recommendation = [FILM]
then  Film Recommendation = [FILM]
```

Plain English

```
if    "film FILM is recommended based on its genre congruence with the
      user's genre preferences or its fulfillment of the user's fantasy;
      and it is recommended based on its congruence with the user's
      gender orientation preferences; and it is recommended based on its
      positive critic rating"
then  "film FILM is recommended for the user"
```

It should be noted that the “plain English” description of the rule actually relies on the meaning that the sub-rules genre-film-recommendation, fantasy-film-recommendation, gender-film-recommendation, and critic-film-recommendation convey on the genre-film-recommendation, fantasy-fulfillment-film-recommendation, gender-film-recommendation, and critical-opinion-film-recommendation predicates, respectively.

The system proceeds to try to satisfy the rule film-recommendation for each film in the system database (i.e. each subject for which there exists an affirmative is-film predication) via backward chaining. A recommendation based on the genre of a film or the fantasies fulfilled by a film is checked for first. A recommendation of a film based on genre requires that the genre of the film match one of jane-doe’s genre preferences as specified by the set of genre-preference-### rules.

```
(defrule genre-preference-3 (:backward :certainty 1.0 :importance 38)
  if  [gender ?who female]
  then [genre-preference ?who romance])
```

These rules require knowledge of jane-doe’s “age category” and gender. An individual falls into one of three age categories (young, middle, older) based on age (in years). This information is encoded in the set of age-category-xxx rules. The system prompts the user, upon reaching this point, to ascertain jane-doe’s (numeric) age.

- What is jane-doe’s age? 22

With the age category of jane-doe determined, the user is then prompted for the gender of jane-doe.

- What is jane-doe’s gender? Female

After addressing genre recommendation, the system attempts to satisfy the rule fantasy-film-recommendation concerning recommendation by way of the fantasies fulfilled by a film.

```
(defrule fantasy-film-recommendation (:backward :certainty 1.0
:importance 87)
  if [and [fantasy ?who ?fantasy]
       [is-film ?film yes]
       [fulfills-fantasy ?film ?fantasy]]
  then [fantasy-fulfillment-film-recommendation ?who ?film])
```

A recommendation of a film based on fantasy fulfillment requires that the fantasy to which `jane-doe` has the strongest response be a fantasy fulfilled by the film. To determine the strongest fantasy of `jane-doe`, the user is prompted.

- What is `jane-doe`'s fantasy? `Triumph-of-underdog`

The set of `fantasy-fulfillment-###` rules encode which fantasies a given film fulfills based on its genre.

```
(defrule fantasy-fulfillment-3 (:backward :certainty 1.0 :importance 63)
  if [and [is-film ?film yes]
       [genre ?film drama]]
  then [fulfills-fantasy ?film triumph-of-underdog])
```

The system next determines whether a given film is recommended for `jane-doe` based on the congruence of the film's gender orientation type with `jane-doe`'s gender orientation preference. The gender orientation type of a film may be neutral, male or female depending on the degree to which its story reflects "maleness"/"femaleness" or has characteristics particularly favored by males/females. This is encoded in the set of `xxx-gender-type` rules.

```
(defrule female-gender-type (:backward :certainty 1.0 :importance 49)
  if [and [is-film ?film yes]
       [or [story-property ?film clear-female-protagonist]
           [story-property ?film focus-on-womens-issue]]]
  then [gender-type ?film female])
```

Similarly, the gender orientation preference of `jane-doe` indicates whether she has a preference for male-oriented or female-oriented films. This is encoded in the `gender-preference` rule.

```
(defrule gender-preference (:backward :certainty 1.0 :importance 47)
  if [gender ?who ?gender]
  then [gender-preference ?who ?gender])
```

According to the `female-gender-type` rule, a film is female-oriented if its story has a "clear female protagonist" or "focuses on women's issues."

```
(defrule female-gender-type (:backward :certainty 1.0 :importance 49)
  if [and [is-film ?film yes]
       [or [story-property ?film clear-female-protagonist]
           [story-property ?film focus-on-womens-issue]]]
  then [gender-type ?film female])
```

The set of `story-property-###` rules relate the "issues addressed" by a film as well as other film information to these generalized story properties. Unlike the other rules in

the knowledge base, these rules are forward-chaining as they depend solely on properties of films that are fully specified in the film database.

Finally, the system checks the critic rating of the film; a film is recommended only if it has a “positive” review. This is determined by the pair of `xxx-critic-rating` rules: a film is concluded to have a positive review if its rating is at least 60.

```
(defrule positive-critic-rating (:backward :certainty 1.0 :importance 94)
  if [and [is-film ?film yes]
        [critical-rating ?film ?rating]
        (>= ?rating 60)]
  then [critical-opinion ?film positive])
```

If a film satisfies this as well as the previous recommendation conditions, then it is recommended by the system.

Problem Domain

The recommendation system is able to provide a small list of films that a person would enjoy viewing based on a three-variable profile—specifying age, gender and a personal fantasy—of the individual. That is, the system is able solve the problem of determining *some* subset of films that an individual is likely to enjoy based on his personal profile. Even with the limited amount of personal information that the system collects, meaningful recommendations can still be made given the fairly large target provided by this particular problem: there is a wide selection of films that a person finds enjoyable. In addition, people tend enjoy a variety of different types or classes of films. Therefore, the problem of identifying anyone of these classes and touching some small portion of this space is a reasonable goal for the system.

The system necessarily encodes further knowledge (used to ultimately decide the films to recommend) that allows it to meaningfully answer some other problems in addition to its target problem. The following are some such problems that the system can solve.

- **Film Information:** answer questions concerning properties of the films in its database. This includes returning such basic information as the film’s genre, release year, MPAA-issued content rating, and actors. More importantly, the system can also provide answers to queries concerning more abstract topics such as the issues addressed by a film; the actor playing the protagonist of a film; whether a film received overall positive review from critics; and various characteristics of the film’s story (e.g. focuses on competition, clear female protagonist).
- **Fantasy Fulfillment:** provide a list of films that satisfy a particular fantasy (e.g. being loved) *based on the genre* of the film.
- **Male/Female Orientation:** indicate which films are particularly male-oriented or female-oriented.
- **Genre Preferences:** determine any significant genre preferences that an individual may have.

There are some problems very similar to target problem that the system, however, cannot handle. For example, while the system can reliably recommend a set of films that

the user should enjoy, it cannot distinguish among those films that it recommends (or does not recommend). It cannot determine which of the recommended films might be liked more so than the others. To be more specific, it is unable to meaningfully rank order films based on preference.

Lacking a richer model of exactly how personal characteristics affect film preferences, a hard threshold (i.e. likes or dislikes) is used throughout the implementation of the system. For example, as encoded by the system rules, it is assumed that young males uniformly dislike the romance genre. While this rule may hold in most cases, it obviously does not hold for every young male or every film in the romance genre. The system, thus, always removes any film in the romance genre from being considered for recommendation if the user is young and male.

This is acceptable since the system only needs to find *some* film or films to recommend the user. In fact, it is actually advantageous given the limited amount of information that the system has to weed out potentially bad recommendation choices. However, this feature also makes the system function poorly at the problem of finding a coverage or approximate coverage of all the films that would make a good recommendation to a person.

Another problem on which the system functions poorly is that of determining if any given film is a good choice. This differs from the general problem of recommending a film in that being able to determine if a given film fits an individual's film preferences provides a much smaller target than that of finding any film that fits an individual's film preferences. Being able to solve the former problem with some high degree of accuracy requires extremely detailed knowledge of an individual's film preferences; the system would need to be able to account for all of the various "types" of films that a person enjoys rather than being able to identify a single type of film from which to make a recommendation.

Knowledge Base

The knowledge base of the system was derived from various texts rather than a particular domain expert. The choice to do this was necessitated by the fact that it was not possible to identify an appropriate knowledge expert that could be interviewed. As such, three papers detailing studies done concerning factors that influence film preference served as the main sources of domain knowledge used to construct the system. The Internet websites Internet Movie Database (IMDb) and Rotten Tomatoes provided specific film data such as starring actors and critic ratings.

The film recommendation system is implemented as a backward chaining rule-based system. Its knowledge is represented in a collection of rules, which describes how a profile of a person translates into preferences for certain films, and a collection of predication, which describes various properties of the thirteen films from which recommendations are made. As an example, the predication that describe the film *Star Wars* are shown here.

```
[is-film star-wars yes]
[genre star-wars sci-fi]
[release-year star-wars 1977]
[mpaa-rating star-wars pg]
[critical-rating star-wars 93]
[story-property star-wars action-filled]
```

```

[story-property star-wars general]
[issue-addressed star-wars general]
[issue-addressed star-wars father-son-relationship]
[issue-addressed star-wars good-vs-evil]
[protagonist-actor star-wars mark-hamill]
[actor star-wars mark-hamill]
[actor star-wars harrison-ford]
[actor star-wars carrie-fisher]
[actor star-wars alec-guinness]
[actor star-wars anthony-daniels]
[actor star-wars james-earl-jones]

[gender mark-hamill male]
[gender harrison-ford male]
[gender carrie-fisher female]
[gender alec-guinness male]
[gender anthony-daniels male]
[gender james-earl-jones male]

```

As shown above, describing the various properties of a film are the predicates `is-film`, `genre`, `release-year`, `mpaa-rating`, `critical-rating`, `story-property`, `issue-addressed`, `protagonist-actor`, and `actor`. Additionally, for each actor in a film is predication indicating the gender of the actor. Each of the predicates is explained in detail below.

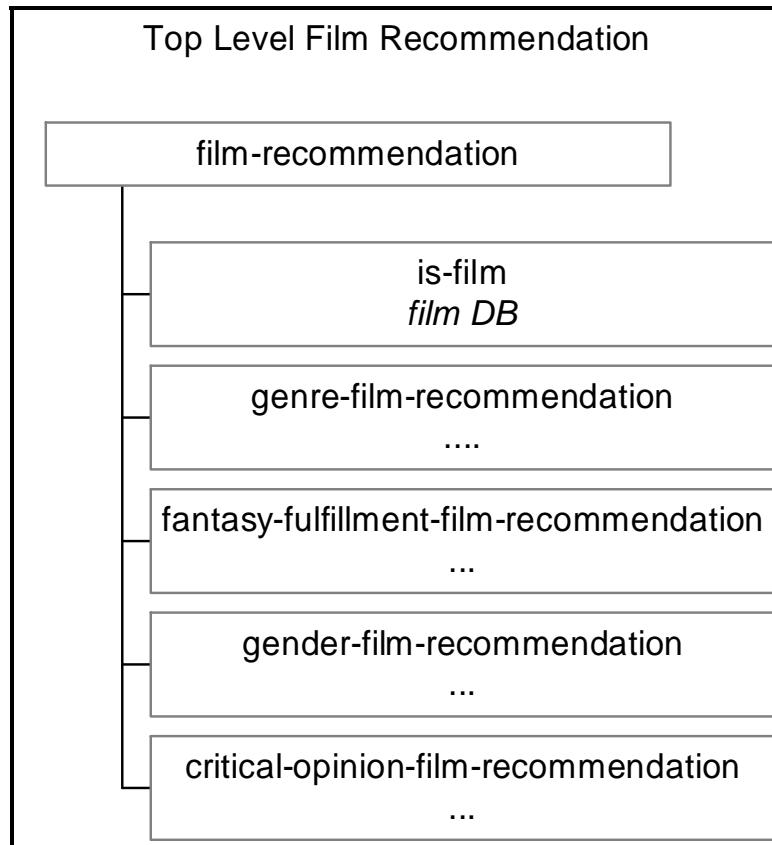
<code>is-film</code>	indicates whether the subject is a film values: yes, no
<code>genre</code>	genre of the film values: drama, comedy, romance, sci-fi, fantasy, adventure, musical
<code>release-year</code>	year in which the film was released values: <i>numeric</i>
<code>mpaa-rating</code>	content rating issued by MPAA values: g, pg, pg-13, r, nc-17
<code>critical-rating</code>	aggregate rating of film critics as indicated at Rotten Tomatoes values: <i>numeric</i>
<code>story-property</code>	general property of the story of the film values: general, action-filled, focus-on-competition, focus-on-womens-issues, clear-female-protagonist
<code>issue-addressed</code>	notable issues addressed by film values: general, motherhood, reproductive-rights, womens-rights, violence-against-women, feminism, mother-daughter-relationship, friendship-between-women, good-vs-evil, crime, war, american-history, father-son-relationship, family
<code>protagonist-actor</code>	actor that plays the protagonist of the film values: <i>actor's name</i>
<code>actor</code>	actor playing a key character in the film values: <i>actor's name</i>

The overall goal of the system is to match the predicate pattern

```
[film-recommendation ?who ?film]
```

where `?who` is the name of the person for which a film recommendation is being made. This goal is accomplished by satisfying the if-clause of the main rule `film-`

recommendation (shown above). This rule embodies the conditions necessary for the system to recommend a film. It specifies that a film is a good recommendation only if (1a) it can be recommended based on its genre or (1b) it can be recommended based on the fantasies that it fulfills; (2) it can be recommended based on its gender-orientation; and (3) it can be recommended based on critical opinion. (The genre condition and fantasy fulfillment condition are grouped together as they both depend chiefly on the genre of the film.) In other words, this rule provides the system with the knowledge that a film will make a good recommendation only if it can meet the above criteria.



Below the general film-recommendation rule that guides the system are four *xxx-film-recommendation* rules that describe when a film meets the criteria necessary to be recommended on each of the dimensions outlined above. The rule *genre-film-recommendation* states that a film can be recommended based on genre if its genre matches a genre for which a person has a particular preference. The rule *gender-film-recommendation* states that a film can be recommended based on gender-orientation if its gender-orientation (i.e. whether it is male-oriented or female-oriented) matches the gender-orientation preference of a person. The rule *fantasy-film-recommendation* states that a film can be recommended based on fantasy fulfillment if it fulfills a fantasy that a person has. Finally, the rule *critic-film-recommendation* states that a film can be recommended based on critical opinion if it has received a positive review by critics.

```

(defrule genre-film-recommendation (:backward :certainty 1.0 :importance
89)
  if [and [genre-preference ?who ?genre]
       [is-film ?film yes]
       [genre ?film ?genre]]
  then [genre-film-recommendation ?who ?film])

(defrule gender-film-recommendation (:backward :certainty 1.0
:importance 88)
  if [and [gender-preference ?who ?gender]
       [is-film ?film yes]
       [gender-type ?film ?gender]]
  then [gender-film-recommendation ?who ?film])

(defrule fantasy-film-recommendation (:backward :certainty 1.0
:importance 87)
  if [and [fantasy ?who ?fantasy]
       [is-film ?film yes]
       [fulfills-fantasy ?film ?fantasy]]
  then [fantasy-fulfillment-film-recommendation ?who ?film])

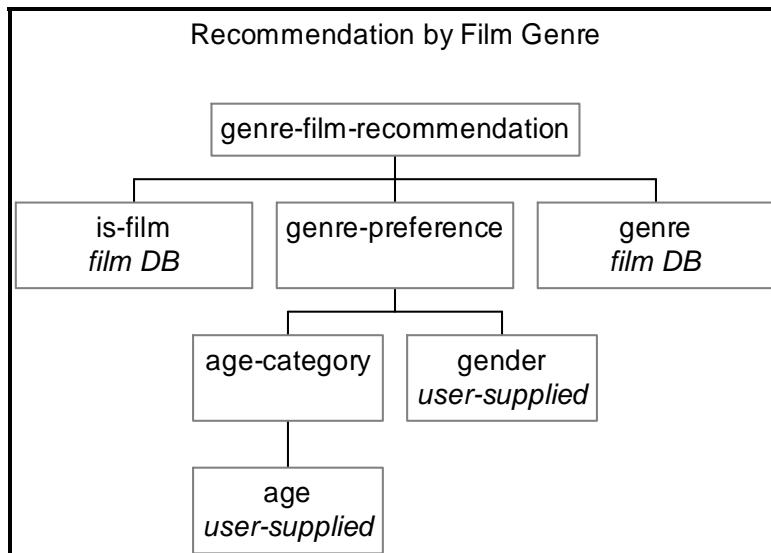
(defrule critic-film-recommendation (:backward :certainty 1.0
:importance 86)
  if [and [is-film ?film yes]
       [critical-opinion ?film positive]]
  then [critical-opinion-film-recommendation ?who ?film])

```

Below these rules in the hierarchy are rules concerned with further detailing the concepts of genre preference, fantasy fulfillment, gender-orientation, and critic opinion.

Genre

The set of rules genre-preference-### describe what personal characteristics correspond to particular genre preferences. For example, rule genre-preference-1 relates the knowledge that individuals that are young and male have a preference for the sci-fi genre. It should be noted that the rules genre-preference-9 and genre-preference-10 indicate that, in general, people have a preference for the genres drama and comedy.



```

(defrule genre-preference-1 (:backward :certainty 1.0 :importance 40)
  if [and [age-category ?who young]
       [gender ?who male]]
  then [genre-preference ?who sci-fi])

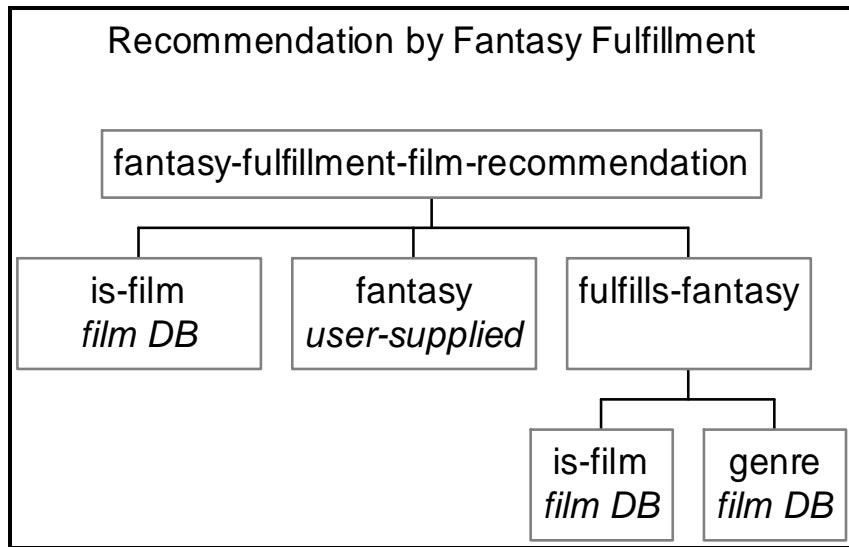
(defrule genre-preference-9 (:backward :certainty 1.0 :importance 32)
  if *true*
  then [genre-preference ?who drama])

(defrule genre-preference-10 (:backward :certainty 1.0 :importance 31)
  if *true*
  then [genre-preference ?who comedy])

```

Fantasy Fulfillment

The set of rules `fantasy-fulfillment-###` describe which types of films fulfill a particular fantasy. More specifically, the rules specify whether a film fulfills a particular fantasy based on its genre. Therefore, for example, rule `fantasy-fulfillment-1` states that a film belonging to the genre action-adventure fulfills the fantasy of adventure.



```

(defrule fantasy-fulfillment-1 (:backward :certainty 1.0 :importance 65)
  if [and [is-film ?film yes]
       [genre ?film action-adventure]]
  then [fulfills-fantasy ?film adventure])

```

Gender Orientation

The rules `male-gender-type` and `female-gender-type` specify the properties that the story of a film must have in order to be male-oriented or female-oriented, respectively. Furthermore, the rule `neutral-gender-type` specifies all films as having a neutral gender orientation. As such, a film is considered to have a neutral gender orientation by default and is regarded as male-oriented or female-oriented if it is later discovered that its story has the necessary properties.

```

(defrule neutral-gender-type (:backward :certainty 1.0 :importance 50)
  if [is-film ?film yes]
  then [gender-type ?film neutral])

(defrule female-gender-type (:backward :certainty 1.0 :importance 49)
  if [and [is-film ?film yes]
       [or [story-property ?film clear-female-protagonist]
           [story-property ?film focus-on-womens-issue]]]
  then [gender-type ?film female])

(defrule male-gender-type (:backward :certainty 1.0 :importance 48)
  if [and [is-film ?film yes]
       [or [story-property ?film action-filled]
           [story-property ?film focus-on-competition]]]
  then [gender-type ?film male])

```

The rule gender-preference encodes which gender-orientation preference an individual demonstrates. As the rule states, in general, people demonstrate a film preference that is congruent with their gender identity. Therefore, males prefer male-oriented films and females prefer female-oriented films.

```

(defrule gender-preference (:backward :certainty 1.0 :importance 47)
  if [gender ?who ?gender]
  then [gender-preference ?who ?gender])

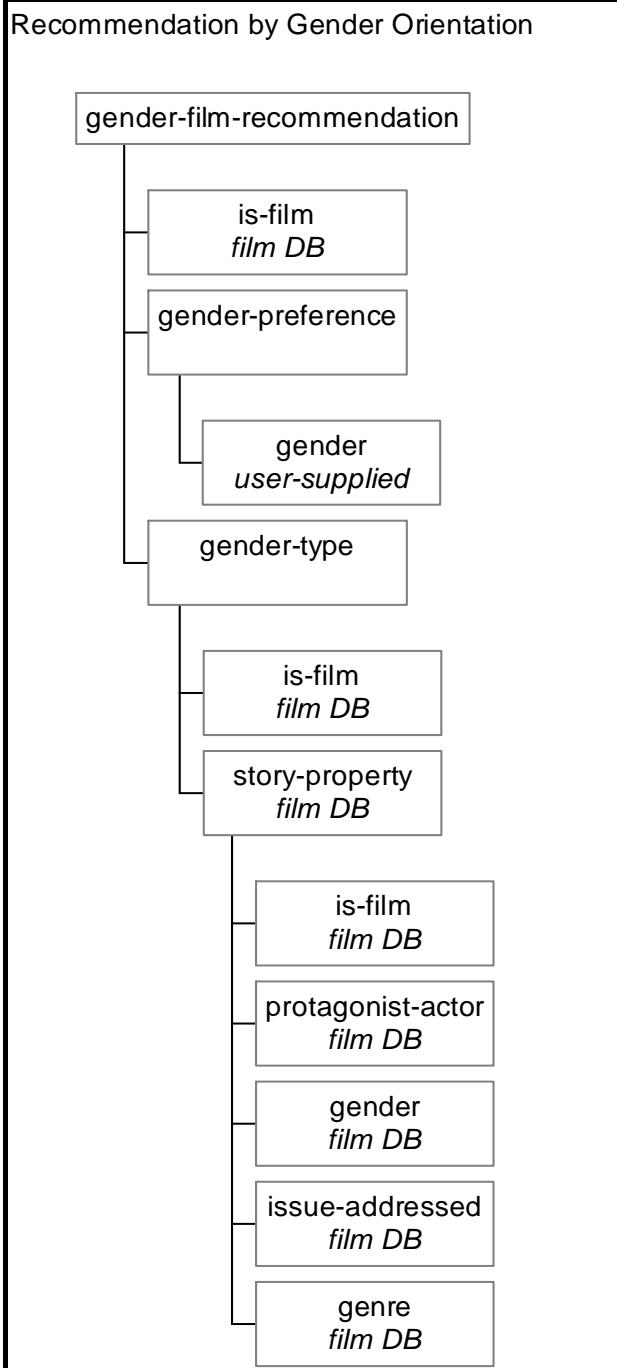
```

Below these rules are the story-property-### rules that encode which basic film properties necessitate particular designations regarding a film's story. For example, the rule story-property-1 encodes the knowledge that a film's story has a clear female protagonist if the actor playing the film's protagonist is female.

```

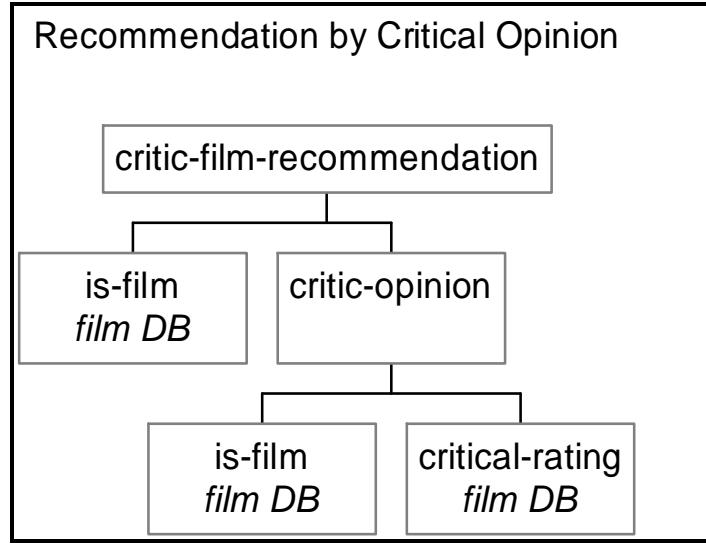
(defrule story-property-1 (:forward)
  if [and [is-film ?film yes]
      [protagonist-actor ?film ?actor]
      [gender ?actor female]]
  then [story-property ?film clear-female-protagonist])

```



Critic Opinion

The positive-critic-rating and negative-critic-rating rules specify what aggregate critic ratings correspond to a positive review and negative review, respectively. Specifically, a film with a critic rating of 60 or better is considered to have been reviewed positively while those with a critic rating below 60 is considered to have been reviewed negatively. This threshold reflects the “fresh” and “rotten” classifications used by the Rotten Tomatoes website.



The final set of rules, `age-category-young`, `age-category-middle`, and `age-category-older`, in the knowledge base concern the age category into which an individual is placed based on his age (in years). As the rule `age-category-young` specifies, a person age 25 or below is considered young; according to the rule `age-category-middle`, a person older than 25 but younger than 50 is considered middle-aged; and according to the rule `age-category-older`, a person 50 years old or older is considered to be older-aged.

```

(defrule age-category-young (:backward :certainty 1.0 :importance 99)
  if [and [age ?who ?x]
       (<= ?x 25)]
  then [age-category ?who young])

(defrule age-category-middle (:backward :certainty 1.0 :importance 98)
  if [and [age ?who ?x]
       (> ?x 25)
       (<= ?x 49)]
  then [age-category ?who middle])

(defrule age-category-older (:backward :certainty 1.0 :importance 97)
  if [and [age ?who ?x]
       (> ?x 49)]
  then [age-category ?who older])
  
```

Problem Solving Approach

The system manages to perform the task of providing a film recommendation by attempting to generate a `film-recommendation` predication for each of the films in its database. Therefore, for each film in the database, the system tries to satisfy the predication of the `if`-clause of the main rule `film-recommendation`, which has a `film-recommendation` predication as its `then`-clause, in a backward chaining fashion. (The system identifies each film in its database by a separate `is-film` predication.) As the system searches depth-first through the rule hierarchy, for the `then`-clause of each rule encountered, it attempts to satisfy the rule for all films in the database. That is, the system does not consider each film separately.

The only deviation from this approach occurs with the group of story-property-### rules. These rules were made forward chaining as they rely simply on information regarding the films in the database, and in effect, their function is to conclude additional useful information about films using the base data that is explicitly entered.

Results

The system in general worked well. The use of a rule-based knowledge representation along with Joshua allowed for simple uniform representation of both the film preference knowledge and basic data concerning films: the film preference knowledge was able to be embodied in if-then rules and the film data represented as individual predication. Given that most of the knowledge that was able to be mined from the expert sources consisted of independent, simple concepts gleamed from examining various aspects of quantitative research (e.g. young males tended to strongly dislike films of the romance gene), the knowledge was easily captured in a collection of if-then rules. Default knowledge could be easily captured by using generate rules (i.e. rules with a satisfied if-clause).

While the film data could be easily expressed as individual predication, having a database filled with hundreds of tell commands to enter each fact about a film into the knowledge base of the system is somewhat cumbersome. It would be desirable, particularly for the maintenance of the knowledge base, to use an implementation that allows a film to be handled as whole object with associated individual properties. Related to this issue is the problem of having to explicitly add each possible value (of hundreds) for film predicates like actor or issue-addressed to their definition before being able to use them.

While most of the knowledge could be easily expressed, knowledge that required treating predication somewhat as entities themselves was not easily expressed. For example, it was difficult to express a notion such as a film where the majority of the actors are female has a female dominated cast. Encoding this knowledge requires not simply being able to match against a particular predication pattern, but being able to determine how many such predication exist and ascertain specific properties of that set.

A more developed version of this system would obviously offer, among other things, a more extensive collection of films. However, the current method in which the system determines film recommendations might not scale well with an increase in the number of films. The system currently attempts to satisfy each rule against every film simultaneously as it chains through the rule hierarchy. Therefore, even if a film has been effectively “ruled out” by failing to match against a necessary predicate that has already been addressed, the system continues to consider the film when attempting to satisfy later predicates. It would be advantageous if the system maintained a set of films that are still “viable” and only considered those films when trying to satisfy clauses.

After building this system, one lesson that will be taken away is the importance of selecting a domain for which there are established domain experts or there is otherwise established expert knowledge. The task of building an expert system should be limited to formalizing pre-existing knowledge and automating reasoning with this knowledge. Attempting to create expert knowledge where it does not exist should not be part of the knowledge engineering task. Moreover, given the lack of knowledge available, it should be considered whether a knowledge-based implementation is the most appropriate for the

task of film recommendation, particularly if any of the related tasks of “recommendation of a particular given film” or “coverage” is a desirable feature for the system. It may be the case that an approach such as that used by the recommendation systems of Amazon.com or Netflix is more appropriate.

Knowledge Sources

Fischoff, S., Lewis, D. and Antonio, J. (1997). Favorite films and film genres as function of race, age, and gender. APA Convention, Chicago, August 1997.

Fischoff, S. (1998). Favorite films choices: influences of the beholder and the beheld. APA Convention, Chicago, August 1998.

Litman (1983, Spring). Predicting Success of Theatrical Movies: An Empirical Study. Journal of Popular Culture, 159-175.

Internet Movie Database (IMDb). <http://www.imdb.com>

Rotten Tomatoes. <http://www.rottentomatoes.com>

Appendix

Film Recommendation

```
(defrule film-recommendation (:backward :certainty 1.0 :importance 90)
  if [and [is-film ?film yes]
       [or [genre-film-recommendation ?who ?film]
           [fantasy-fulfillment-film-recommendation ?who ?film]]
       [gender-film-recommendation ?who ?film]
       [critical-opinion-film-recommendation ?who ?film]]
  then [film-recommendation ?who ?film])

(defrule genre-film-recommendation (:backward :certainty 1.0 :importance 89)
  if [and [genre-preference ?who ?genre]
       [is-film ?film yes]
       [genre ?film ?genre]]
  then [genre-film-recommendation ?who ?film])

(defrule gender-film-recommendation (:backward :certainty 1.0 :importance 88)
  if [and [gender-preference ?who ?gender]
       [is-film ?film yes]
       [gender-type ?film ?gender]]
  then [gender-film-recommendation ?who ?film])

(defrule fantasy-film-recommendation (:backward :certainty 1.0 :importance 87)
  if [and [fantasy ?who ?fantasy]
       [is-film ?film yes]
       [fulfills-fantasy ?film ?fantasy]]
  then [fantasy-fulfillment-film-recommendation ?who ?film])

(defrule critic-film-recommendation (:backward :certainty 1.0 :importance 86)
  if [and [is-film ?film yes]
       [critical-opinion ?film positive]]
  then [critical-opinion-film-recommendation ?who ?film])
```

Fantasy Fulfillment

```
(defrule fantasy-fulfillment-1 (:backward :certainty 1.0 :importance 65)
  if [and [is-film ?film yes]
       [genre ?film action-adventure]]
  then [fulfills-fantasy ?film adventure])

(defrule fantasy-fulfillment-2 (:backward :certainty 1.0 :importance 64)
  if [and [is-film ?film yes]
       [genre ?film action-adventure]]
  then [fulfills-fantasy ?film hero])

(defrule fantasy-fulfillment-3 (:backward :certainty 1.0 :importance 63)
  if [and [is-film ?film yes]
       [genre ?film drama]]
  then [fulfills-fantasy ?film triumph-of-underdog])

(defrule fantasy-fulfillment-4 (:backward :certainty 1.0 :importance 62)
  if [and [is-film ?film yes]
       [genre ?film comedy]]
  then [fulfills-fantasy ?film triumph-of-underdog])

(defrule fantasy-fulfillment-5 (:backward :certainty 1.0 :importance 61)
  if [and [is-film ?film yes]
       [genre ?film romance]]
```

```

        then [fulfills-fantasy ?film romance])

(defrule fantasy-fulfillment-6 (:backward :certainty 1.0 :importance 60)
  if [and [is-film ?film yes]
      [genre ?film romance]]
  then [fulfills-fantasy ?film being-loved])

(defrule fantasy-fulfillment-7 (:backward :certainty 1.0 :importance 59)
  if [and [is-film ?film yes]
      [genre ?film musical]]
  then [fulfills-fantasy ?film musical])

(defrule fantasy-fulfillment-8 (:backward :certainty 1.0 :importance 58)
  if [and [is-film ?film yes]
      [genre ?film sci-fi]]
  then [fulfills-fantasy ?film adventure])

(defrule fantasy-fulfillment-9 (:backward :certainty 1.0 :importance 57)
  if [and [is-film ?film yes]
      [genre ?film sci-fi]]
  then [fulfills-fantasy ?film triumph-of-underdog])

(defrule fantasy-fulfillment-10 (:backward :certainty 1.0 :importance 56)
  if [and [is-film ?film yes]
      [genre ?film fantasy]]
  then [fulfills-fantasy ?film adventure])

(defrule fantasy-fulfillment-11 (:backward :certainty 1.0 :importance 55)
  if [and [is-film ?film yes]
      [genre ?film fantasy]]
  then [fulfills-fantasy ?film triumph-of-underdog])

```

Gender Orientation

```

(defrule neutral-gender-type (:backward :certainty 1.0 :importance 50)
  if [is-film ?film yes]
  then [gender-type ?film neutral])

(defrule female-gender-type (:backward :certainty 1.0 :importance 49)
  if [and [is-film ?film yes]
      [or [story-property ?film clear-female-protagonist]
          [story-property ?film focus-on-womens-issue]]]
  then [gender-type ?film female])

(defrule male-gender-type (:backward :certainty 1.0 :importance 48)
  if [and [is-film ?film yes]
      [or [story-property ?film action-filled]
          [story-property ?film focus-on-competition]]]
  then [gender-type ?film male])

(defrule gender-preference (:backward :certainty 1.0 :importance 47)
  if [gender ?who ?gender]
  then [gender-preference ?who ?gender])

```

Properties of a Film's Story

```

(defrule story-property-1 (:forward)
  if [and [is-film ?film yes]
      [protagonist-actor ?film ?actor]
      [gender ?actor female]]
  then [story-property ?film clear-female-protagonist])

(defrule story-property-2 (:forward)

```

```

if [and [is-film ?film yes]
    [or [issue-addressed ?film motherhood]
        [issue-addressed ?film reproductive-rights]
        [issue-addressed ?film womens-rights]
        [issue-addressed ?film violence-against-women]
        [issue-addressed ?film feminism]
        [issue-addressed ?film mother-daughter-relationship]
        [issue-addressed ?film friendship-between-women]]]
then [story-property ?film focus-on-womens-issue])

(defrule story-property-3 (:forward)
  if [and [is-film ?film yes]
      [genre ?film action-adventure]]
  then [story-property ?film action-filled] )

(defrule story-property-4 (:forward)
  if [and [is-film ?film yes]
      [or [issue-addressed ?film good-vs-evil]
          [issue-addressed ?film crime]
          [issue-addressed ?film war]]]
  then [story-property ?film focus-on-competition] )

```

Genre Preferences

```

(defrule genre-preference-1 (:backward :certainty 1.0 :importance 40)
  if [and [age-category ?who young]
        [gender ?who male]]
  then [genre-preference ?who sci-fi] )

(defrule genre-preference-2 (:backward :certainty 1.0 :importance 39)
  if [and [age-category ?who young]
      [gender ?who male]]
  then [not [genre-preference ?who romance]] )

(defrule genre-preference-3 (:backward :certainty 1.0 :importance 38)
  if [gender ?who female]
  then [genre-preference ?who romance] )

(defrule genre-preference-4 (:backward :certainty 1.0 :importance 37)
  if [age-category ?who young]
  then [genre-preference ?who action-adventure] )

(defrule genre-preference-5 (:backward :certainty 1.0 :importance 36)
  if [and [age-category ?who older]
        [gender ?who female]]
  then [not [genre-preference ?who action-adventure]] )

(defrule genre-preference-6 (:backward :certainty 1.0 :importance 35)
  if [age-category ?who young]
  then [not [genre-preference ?who musical]] )

(defrule genre-preference-7 (:backward :certainty 1.0 :importance 34)
  if [age-category ?who older]
  then [genre-preference ?who drama] )

(defrule genre-preference-8 (:backward :certainty 1.0 :importance 33)
  if [age-category ?who older]
  then [genre-preference ?who romance] )

(defrule genre-preference-9 (:backward :certainty 1.0 :importance 32)
  if *true*
  then [genre-preference ?who drama] )

(defrule genre-preference-10 (:backward :certainty 1.0 :importance 31)
  if *true*

```

```
    then [genre-preference ?who comedy])
```

Age Category

```
(defrule age-category-young (:backward :certainty 1.0 :importance 99)
  if [and [age ?who ?x]
        (<= ?x 25)]
  then [age-category ?who young])

(defrule age-category-middle (:backward :certainty 1.0 :importance 98)
  if [and [age ?who ?x]
      (> ?x 25)
      (<= ?x 49)]
  then [age-category ?who middle])

(defrule age-category-older (:backward :certainty 1.0 :importance 97)
  if [and [age ?who ?x]
      (> ?x 49)]
  then [age-category ?who older])
```

Recency of Film Release

```
(defrule film-release-recent (:backward :certainty 1.0 :importance 96)
  if [and [current-year sys ?cy]
      [is-film ?film yes]
      [release-year ?film ?ry]
      (< (- ?cr ?ry) 20)]
  then [is-recent-release ?film yes])

(defrule film-release-not-recent (:backward :certainty 1.0 :importance 95)
  if [and [current-year sys ?cy]
      [is-film ?film yes]
      [release-year ?film ?ry]
      (>= (- ?cy ?ry) 20)]
  then [is-recent-release ?film no])
```

Critical Opinion

```
(defrule positive-critic-rating (:backward :certainty 1.0 :importance 94)
  if [and [is-film ?film yes]
      [critical-rating ?film ?rating]
      (>= ?rating 60)]
  then [critical-opinion ?film positive])

(defrule negative-critic-rating (:backward :certainty 1.0 :importance 93)
  if [and [is-film ?film yes]
      [critical-rating ?film ?rating]
      (< ?rating 60)]
  then [critical-opinion ?film negative])
```