



De La Salle University
Term 3 Academic Year 2023-2024

In Partial fulfillment of the course
CSINTSY - S13

MCO2 - Logic Based Model Report

Submitted By:
Ariaga, Marian Ricci N.
Guillarte, Dana Louise A.
Llorando, Yesha P.
O'Neil, Samantha Erica R.
So, Chryssille Grace L.

Submitted to:
Dr. Norshuhani Zamin
July 26, 2024

I. Introduction

Love is a universal language understood by everyone. As human beings, it is in our nature to love, whether it is something or someone, and we all deserve to be loved. Love is present everywhere, from our family members to our friends. However, many people believe that there is someone out there in the universe who will offer us a kind of love beyond familial and platonic love—a profound, romantic love that takes time to find and nurture.

In the journey to find these deep connections, we often reflect on our own personality traits and character to see if they align with a potential partner. Recognizing the importance of this, we have chosen relationships and love matching as the domain for our expert system. To assist in finding suitable matches based on individual personalities, we have developed a comprehensive expert system. This system considers numerous factors, such as personality types and personal preferences, to find the perfect love match tailored to each person's unique qualities and needs.

To gain a comprehensive understanding of a user's personality and preferences, we base our approach on six key factors, each providing crucial insights. The first factor is the Myers-Briggs Type Indicator (MBTI), which categorizes individuals into 16 distinct personality types based on preferences in four dimensions: Introversion/Extraversion, Sensing/Intuition, Thinking/Feeling, and Judging/Perceiving (*The 16 MBTI® Personality Types*, 2023). Understanding a user's MBTI type offers valuable insights into their personality, aiding in aligning them with potential matches. The second factor is attachment style, reflecting the type of bonding learned in childhood and carried into adult relationships. There are four primary attachment styles: secure, anxious, avoidant, and fearful-avoidant, which help predict relationship dynamics (*Attachment styles*, n.d.). The third factor is love languages, categorized into Words of Affirmation, Acts of Service, Receiving Gifts, Quality Time, and Physical Touch (*Discover Your Love Language® - the 5 Love Languages®*, n.d.). Understanding each other's love language enhances communication of affection. Shared interests form the fourth factor, providing a foundation for bonding and mutual enjoyment. The fifth factor involves personal values, including family, career, friends, personal growth, adventure, stability, independence, and community. By ranking these values, we ensure partners' priorities align. The final factor is communication styles, encompassing assertive, aggressive, passive, and passive-aggressive (Herrity, 2023). Ultimately, the idea is to utilize a combination of psychometric assessments like

MBTI, love languages, attachment styles, and communication styles, as well as user preferences like interests and values.

By integrating these factors—MBTI types, attachment styles, love languages, interests, personal values, and communication styles—our expert system creates a holistic profile of each individual. This comprehensive approach allows us to identify potential matches that are deeply compatible on multiple levels, increasing the likelihood of finding a truly perfect love match.

II. Pseudocode

Below are code snippets illustrating the group of facts declared in Prolog to create the expert system:

```
% Facts about MBTI types
mbti(intj).
mbti(entp).
mbti(infj).
mbti(enfp).
mbti(isfj).
mbti(estj).
mbti(esfp).
mbti(istp).
mbti(intp).
mbti(entj).
mbti(infp).
mbti(enfj).
mbti(istj).
mbti(isfp).
mbti(estp).
mbti(esfj).
```

First, we declared a predicate called `mbti` to denote MBTI types. For example, `mbti(intj)` asserts that INTJ is a valid MBTI type. The system asks the user for their MBTI type and takes this into account.

```
% Facts about attachment styles
attachment_style(secure).
attachment_style(anxious).
attachment_style(avoidant).
```

```
attachment_style(fearful_avoidant).
```

Next, we set facts for the four attachment styles and declared a predicate called `attachment_style`. The arguments `secure`, `anxious`, `avoidant`, and `fearful_avoidant` are valid attachment styles in the system.

```
% Facts for recommending attachment style based on MBTI
possible_attachment_style(intj, avoidant).
possible_attachment_style(entp, secure).
possible_attachment_style(infj, anxious).
possible_attachment_style(enfp, anxious).
possible_attachment_style(isfj, secure).
possible_attachment_style(estj, secure).
possible_attachment_style(esfp, secure).
possible_attachment_style(istp, avoidant).
possible_attachment_style(intp, avoidant).
possible_attachment_style(entj, secure).
possible_attachment_style(infp, anxious).
possible_attachment_style(enfj, secure).
possible_attachment_style(istj, secure).
possible_attachment_style(isfp, anxious).
possible_attachment_style(estp, secure).
possible_attachment_style(esfj, secure).
```

Using this set of facts from the predicate `possible_attachment_style`, the system automatically infers the user's attachment style based on their MBTI type, as individuals with the same MBTI type often exhibit similar attachment styles, making them identifiable based on MBTI type.

```
% Facts about love languages
love_language(words_of_affirmation).
love_language(acts_of_service).
love_language(receiving_gifts).
love_language(quality_time).
love_language(physical_touch).
```

The snippet above shows facts about the six love languages and assigns them to the predicate `love_language`. The system asks the user which love language they prefer to receive from a partner.

```
% Facts about interests
interest(reading).
interest(traveling).
interest(sports).
interest(music).
interest(cooking).
interest(art).
interest(gaming).
interest(pet_care).
interest(writing).
interest(science).
```

Next, there are facts about interests, with different hobbies categorized under the predicate `interest`. For this system, we have limited these to eight interests and asked the user to select their top three interests.

```
% Facts about values
value(family).
value(career).
value(friends).
value(personal_growth).
value(independence).
value(community).
```

The following set of facts pertains to different values that a person may prioritize in life. The system asks the user to rank these eight values according to their order of priority.

```
% Facts about communication styles
communication_style(assertive).
communication_style(passive).
communication_style(aggressive).
communication_style(passive_aggressive).
```

Next, the facts about the four communication styles are presented. These include assertive, passive, aggressive, and passive-aggressive.

```
% Facts for recommending communication style based on MBTI
possible_communication_style(intj, assertive).
possible_communication_style(entp, assertive).
```

```

possible_communication_style(infj, passive).
possible_communication_style(enfp, assertive).
possible_communication_style(isfj, passive).
possible_communication_style(estj, assertive).
possible_communication_style(esfp, assertive).
possible_communication_style(istp, assertive).
possible_communication_style(intp, passive).
possible_communication_style(entj, assertive).
possible_communication_style(infp, passive).
possible_communication_style(enfj, assertive).
possible_communication_style(istj, assertive).
possible_communication_style(isfp, passive).
possible_communication_style(estp, assertive).
possible_communication_style(esfj, assertive).

```

These communication styles are also automatically assigned by the system to the user according to their MBTI, as individuals with the same MBTI type typically share similar communication styles. This is done by creating the predicate `possible_communication_style` with the MBTI type and the matching communication style as arguments

```

% Facts about MBTI compatibility
compatible(intj, entp).
compatible(entp, intj).
compatible(infj, enfp).
compatible(enfp, infj).
compatible(isfj, estj).
compatible(estj, isfj).
compatible(esfp, istp).
compatible(istp, esfp).
compatible(intp, enfj).
compatible(entj, infp).
compatible(infp, enfj).
compatible(enfj, intp).
compatible(istj, esfj).
compatible(isfp, estp).
compatible(estp, isfp).
compatible(esfj, istj).

```

The set of facts above matches the 16 MBTI personalities with their compatible MBTI personalities using the predicate `compatible`, with the two MBTI personalities as arguments.

```
% Facts for complementing love language
complementing_love_language(acts_of_service, receiving_gifts).
complementing_love_language(receiving_gifts, acts_of_service).
complementing_love_language(words_of_affirmation, quality_time).
complementing_love_language(quality_time, words_of_affirmation).
complementing_love_language(physical_touch, quality_time).
```

To match the love languages with their complementing counterparts, the facts above create a predicate called `complementing_love_language`, which pairs love languages as arguments.

```
% Facts about values complements
complementing_value(independence, personal_growth).
complementing_value(personal_growth, independence).
complementing_value(friends, family).
complementing_value(family, friends).
complementing_value(community, career).
complementing_value(career, community).
```

The facts above match the user's values using the predicate `complementing_value`.

```
% Facts about interests complements
complementing_interest(reading, cooking).
complementing_interest(cooking, reading).
complementing_interest(sports, gaming).
complementing_interest(gaming, sports).
complementing_interest(music, art).
complementing_interest(art, music).
complementing_interest(traveling, pet_care).
complementing_interest(pet_care, traveling).
complementing_interest(writing, science).
complementing_interest(science, writing).
```

The last set of facts matches the user's interests with their complementing interests using the predicate `complementing_interest`, with the interests as arguments.

The next set of code snippets shows the rules used for the love-matching expert system.

```
% Rules
```

```
recommended_partner(MBTI, Partner) :- compatible(MBTI, Partner).
```

The Prolog rule above states that a partner (Partner) is recommended for someone with a given MBTI personality type (MBTI) if their MBTI types are compatible according to the predicate compatible.

```
recommended_attachment_style(MBTI, AttachmentStyle) :-  
possible_attachment_style(MBTI, AttachmentStyle).
```

This rule states that an attachment style (AttachmentStyle) is recommended for someone with a given MBTI personality type (MBTI) if that attachment style is considered possible for that MBTI type based on the predicate possible_attachment_style.

```
recommended_communication_style(MBTI, CommunicationStyle) :-  
possible_communication_style(MBTI, CommunicationStyle).
```

This rule states that a communication style (CommunicationStyle) is recommended for someone with a given MBTI personality type (MBTI) if that communication style is considered a possibility for that MBTI type based on the predicate possible_communication_style.

```
recommended_partner_love_language(LoveLanguage, PartnerLoveLanguage) :-  
complementing_love_language(LoveLanguage, PartnerLoveLanguage).
```

This rule states that a partner's love language (PartnerLoveLanguage) is recommended if it complements the user's love language (LoveLanguage) based on the complementing_love_language relationship.

```
recommended_partner_value(Value, PartnerValue) :-  
complementing_value(Value, PartnerValue).
```

This rule states that the matching partner's value (PartnerValue) is recommended if it complements the values of the user (Value) based on the complementing_value relationship.

```
recommended_partner_interest(Interest, PartnerInterest) :-  
complementing_interest(Interest, PartnerInterest).
```

This rule states that a partner's interest (PartnerInterest) is recommended if it complements the given interest (Interest) based on the complementing_interest relationship.

These rules form the foundation of the expert system, enabling it to recommend ideal partner types for users. The system provides results for compatible MBTI types, attachment styles, communication styles, love languages, values, and interests, guiding users towards finding suitable partners based on these compatibility factors.

III. Results and Analysis

The Prolog system employs a knowledge base consisting of facts and rules to model various personality and relationship attributes. Key rules define compatibility between MBTI types, potential attachment styles based on MBTI, recommended communication styles, and complementary love languages, values, and interests. The system processes user input by querying the knowledge base for relevant information. For instance, the system retrieves compatible MBTI types with the user, then it will generate their possible attachment styles, and recommended communication styles. It then identifies potential partners based on compatibility and generates recommendations by considering factors such as complementary love languages, shared values, and common interests.

The graphical user interface (GUI), implemented in Python, presents a clean and user-friendly interface that effectively communicates the recommended partner profile. It effectively garners information such as MBTI type, love language, interests, and values, then it will display a results page providing users with a comprehensive overview of their potential match.



Figure 1: GUI.

Breaking down the process, the program will first ask the user for their gender. After that, they are asked their MBTI type.



Figure 2: Gender Selection.

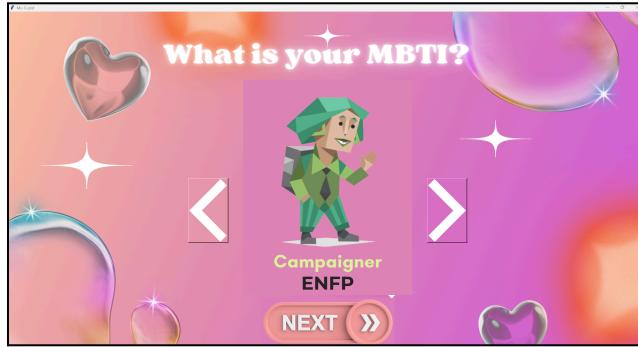


Figure 3: MBTI Type Selection.

It will then display their attachment style and communication style:

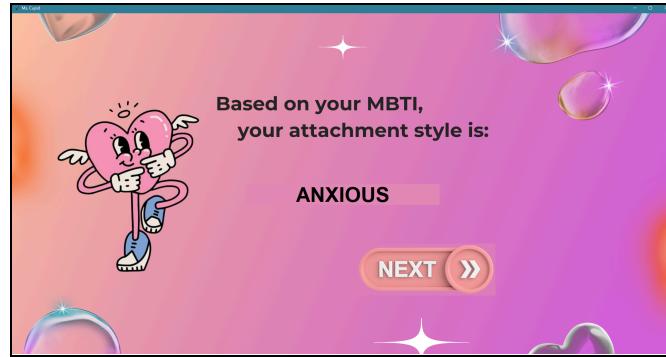


Figure 4: Attachment Style based on MBTI Type.

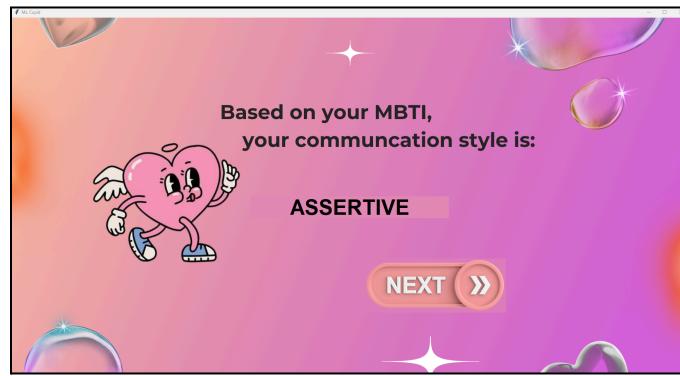


Figure 5: Communication style based on MBTI Type.

After collecting this initial information, the program will shift to obtaining the information needed to recommend a partner to the user, it will first ask them for their love language.

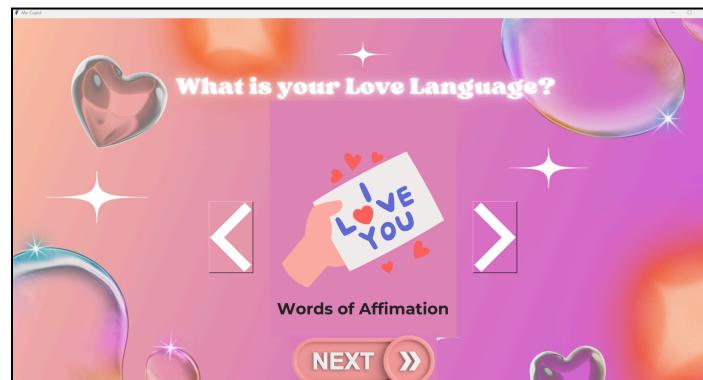


Figure 6: Love Language Input.

Secondly, it will ask them to rank Family, Friends, Career, Personal growth, Independence, and Community according to their respective values.



Figure 7: Ranking of values.

Lastly, it will ask the user to choose 3 main interests from a list of various activities, and based on all these it will generate comprehensive love recommendations for the user.



Figure 8: Ranking of Interests.

To further visualize this process, let's say the user is an ENFP: their most compatible MBTI type would be INFJ, their communication style would be assertive, and their attachment style would be anxious, all based on the rules and facts established in PROLOG. It would then evaluate the user's interests (Refer to Figure 8) and values (Refer to Figure 7) to determine a compatible partner for them and offer advice. The results of the program after inputting the aforementioned details can be seen in the figure below.

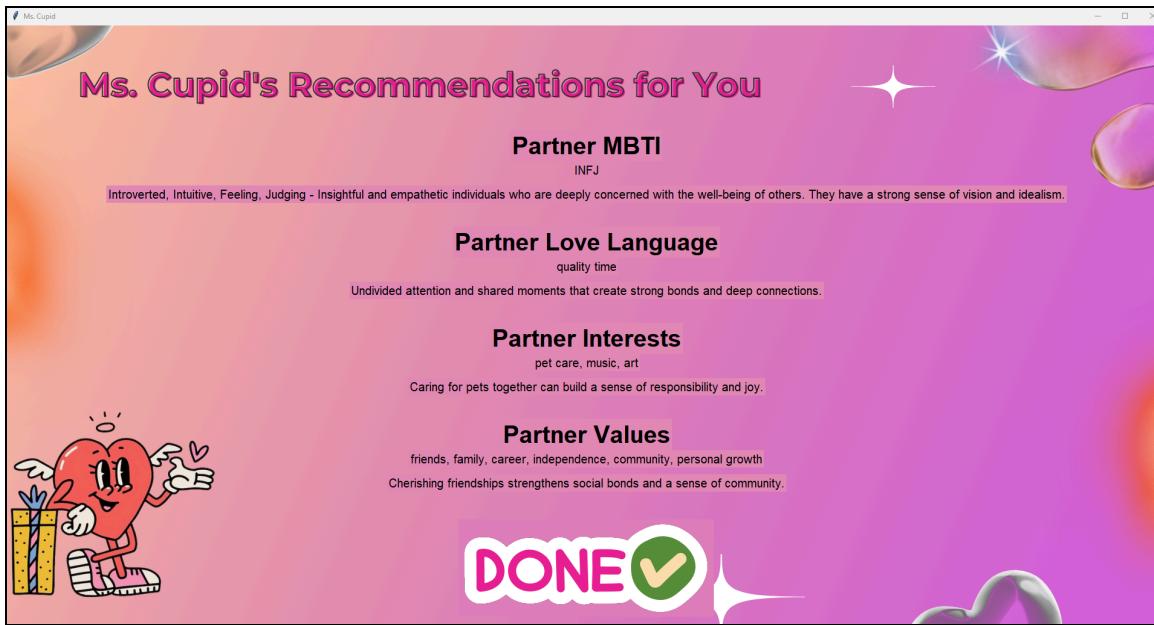


Figure 9: Love recommendations garnered based off user input.

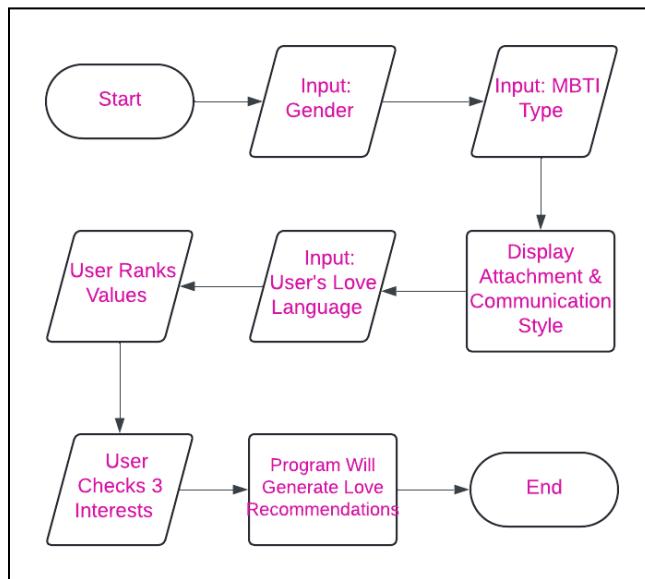


Figure 10: Flowchart for the Prolog-based program (Lucid Chart).

```

(Start) -> [Input: User's gender and MBTI type] -> [Display attachment and communication style] -> [Shift to partner recommendation] -> [Input: User's love language] -> [Input: Rank values] -> [Input: Top 3 interests] -> [Generate love recommendations] -> (End)
  
```

Based on the flowchart and walkthrough of the program above, the Prolog-based program effectively utilizes a rule-based approach to collect and process user information, culminating in tailored love recommendations. By sequentially gathering data on gender, MBTI type, attachment and communication styles, love language, values, and interests, the system efficiently generates suggestions.

Since this application is limited to simple Prolog rules and facts, the accuracy of the results may not be fully comprehensive. Additionally, the group decided to focus on MBTI personalities, values, and interests which are all voluntarily selected by the user, increasing the risk of human-error. Moreover, the values and interests displayed are not based on psychometric assessments or psychological typologies, unlike MBTI and the rest of the facts used in this project. Thus, the user might have certain values or interests that can not be found in the choices given by the program. Nevertheless, this project showcases the potential of Prolog in developing intelligent matchmaking applications, demonstrating its suitability for knowledge representation and logical reasoning tasks.

IV. References

The 16 MBTI® Personality Types. (2023, July 20). 2003-2024, Myers & Briggs Foundation.

<https://www.myersbriggs.org/my-mbti-personality-type/the-16-mbti-personality-types/>

Attachment styles. (n.d.). Cleveland Clinic.

<https://my.clevelandclinic.org/health/articles/25170-attachment-styles>

Discover your love language® - The 5 Love Languages®. (n.d.). Northfield Publishing.

<https://5lovelanguages.com/>

Herrity, J. (2023). *4 types of communication styles and how to improve yours.* Indeed.

<https://www.indeed.com/career-advice/career-development/communication-styles>

V. Contributions of Each Member

Name	Contribution
Ariaga, Marian Ricci N.	<ul style="list-style-type: none">• Paper Writing (Introduction, Pseudocode)

Guillarte, Dana Louise A.	<ul style="list-style-type: none"> • PROLOG Code
Llorando, Yesha P.	<ul style="list-style-type: none"> • Tkinter GUI Code
O'Neil, Samantha Erica R.	<ul style="list-style-type: none"> • Paper Writing (Results and Analysis)
So, Chrysille Grace L.	<ul style="list-style-type: none"> • Controller Code, Helped in PROLOG Code,