In Proceedings of the Multiagent-based Simulation (MABS) Workshop at Autonomous Agents and Multiagent Systems, Toronto, Canada, May 2010. Post-proceedings.

Toward a Myers-Briggs Type Indicator Model of Agent Behavior in Multiagent Teams

Jordan Salvit[1] and Elizabeth Sklar[1,2]

[1] Brooklyn College, City University of New York, Brooklyn NY 11210 USA,
jordan@jordansalvit.com
[2] The Graduate Center, City University of New York, New York NY 10016 USA
sklar@sci.brooklyn.cuny.edu

Abstract. This paper explores the use of the Myers-Briggs Type Indicator (MBTI) as the basis for defining the *personality* of an agent. The MBTI is a well-known psychological theory of human personality. In the MBTI model, four axes are defined to explain how humans perceive their environment, how they interact with others and how they make decisions based on these traits. The work described here presents a preliminary model of agent behavior in which two of the axes are implemented, combining to reflect four distinct agent personality types. Experiments were conducted under three environmental conditions: single agent setting, homogeneous multiagent team, and heterogeneous multiagent team. Results are presented for each condition and are analyzed in comparison with the other conditions, as well as within the context of the expected MBTI behaviors given each environment and the simulated task. It is demonstrated that agents of each personality type produce very different results, distinct for and characteristic of each MBTI personality type.

1 Introduction

We explore the use of the Myers-Briggs Type Indicator (MBTI) as the basis for defining the *personality* of an agent. The MBTI is a well-known psychological theory of human personality developed in the mid 1900's by Katharine Myers and Isabel Briggs Myers [1], based on an earlier theory developed by Carl Jung [2]. Four axes are defined to explain how humans perceive their environment, how they interact with others and how they make decisions based on these traits.

Jung's theory states that human mental activity essentially involves receiving information and processing that information to make decisions. The input of information ("perceiving", according to Jung) can be handled in one of two ways, either by overtly *sensing* or by using *intuition*. The process of making decisions ("judging", according to Jung) can be driven by logical *thinking* or by emotional *feelings*. Some people derive their energy for these processes from the influences of the external world around them (*extroversion*), while others rely on internal mechanisms such as thoughts or memories (*introversion*). Briggs and Myers expanded on these three dichotomies by adding a fourth "lifestyle" axis which distinguishes between people whose personalities rely more on either perception or judging.

Typical results of MBTI tests label individuals using one-character abbreviations for each pole on each axis, as follows:

– *Extraversion* (E) versus *Introversion* (I)
– *Sensing* (S) versus *iNtuition* (N)
– *Thinking* (T) versus *Feeling* (F)
– *Judging* (J) versus *Perceiving* (P)

So, for example, an individual whose personality is labeled ENTJ is someone who gets their energy from interacting with others, who makes decisions based on observations of their environment, who solves problems using logical reasoning and is organized and methodical about what they do. An ENTJ individual makes a commitment to complete a certain task in a certain way and sticks with their plan until the task is complete. In contrast, an individual whose personality is labeled ISFP is someone who gets their energy from inside, who learns from experience and focuses on facts, and lets emotions influence their decision-making. An ISFP individual commits to a task, but constantly re-evaluates to decide if there is a better way to complete the task or a better task to address.

The Meyers-Briggs hypothesis is that all combinations of $4^2 = 16$ personality types exist in humans, and knowledge of which personality type corresponds to an individual can help that individual make life and career decisions. For example, certain personality types tend to be well-suited to particular types of jobs; certain pairings of personality types tend to work better than others for business or life partners. People use the MBTI model to influence decisions or explain how decisions they have made in the past or actions they have taken have been driven.

We are interested in applying MBTI to agent-based systems by implementing agents with different personality types. Although there exist in the literature a range of frameworks and some widely accepted methodologies for agent modeling (e.g., [3, 4]), most models abstractly describe how an agent processes inputs and executes outputs, leaving the details to the discretion of the developer. We speculate that it may be the case that a developer will, subconsciously, encode in the agents her own personality type. The work presented here demonstrates that each personality type performs differently, even on a simple task in a simplified environment. The resulting observation in our simulated environment is that some personality types are better suited to the task—the same observation that psychologists make about humans. The implication in the agent modeling and agent-based simulation communities is that the success or failure of an experiment could be affected by the agents' inherent personality types, rather than (necessarily or exclusively) the underlying theory driving the experiment. Thus the need for a concise model of personality type arises.

In the long term, we envision an additional step in agent modeling in which *personality type* plays a factor. When constructing a system, after selecting an agent's behavioral model, the agent's environment, its tasks and goals, the developer can determine experimentally which (set of) personality type(s) would best be suited to accomplish those goals. MBTI is generally used to help develop people's understanding of each other and how differences are not flaws but features, when recognized as such. MBTI is a tool to help people, organizations and/or teams learn how best to leverage each other's personality preferences to accomplish their goals together. Our aim is to bring these ideas, in the context of agent design, to the agent modeling and agent-based simulation fields. We believe that MBTI can provide a clear methodology for expressing and applying agent personality types.

The work described here presents a preliminary model of agent behavior where the MBTI personality types are employed as the basis for defining different agent personalities. As a first step, we focus here on the two axes that do not look at other agents, namely: sensing (S) versus intuition (N) and judging (J) versus perceiving (P). We implement agents exhibiting each of the $2^2 = 4$ personality types: SJ, SP, NJ and NP. The agents are deployed in a simple environment and given simple tasks to complete. The results show marked differences in the way agents of each personality type address the given task.

The remainder of the paper is organized as follows. Section 2 outlines our approach, describing the simulated environment used for experimentation and explaining how each of the four personality types are implemented within this environment. Section 3 presents experiments in which agents of each personality type perform tasks in the simulated environment and produce very different results, distinct for each personality type. Section 4 describes some related work in the literature. Finally, we close with a summary and discussion of future work.

2 Approach

This section introduces our simulated environment and describes how each of the personality types are exhibited within that context. The implementation details for each of the two personality preference axes studied here are explained.

Our methodology first considers how each personality preference axis (S versus N and J versus P) applies within the given environment. Then a set of rules is defined for each axis that modulates the interpretation of input and the production of output, according to the characteristic personality preferences of the two extremes along that axis. Rather than engineering four separate rule sets, one for each of the four personality types (i.e., SJ, SP, NJ and NP), instead two separate rule sets are composed: one that distinguishes between S and N, and one that distinguishes between J and P. Each agent invokes task-dependent functions at run-time, and the behavior of each function is affected by the combined influence of the agents' two separate personality preference rule sets. Details are discussed below, within the context of our simulated environment.

Our environment is based on an existing model from the artificial life community in which termites are simulated [5]. The termites' task is to gather food from their environment and place it in piles. We modify the baseline termite model by using pre-determined locations (instead of allowing the number and locations of piles to emerge as the simulation runs) in order to help illustrate the distinguishing characteristics of the different agent personality types. The environment is represented as a two-dimensional grid, where each (x, y) location in the grid is referred to as a "patch". The differences between the personalities should be revealed quantitatively in terms of the amount of food gathered and delivered to a pile, the number of different patches visited, and the time interval between gathering a food particle and delivering it to a pile.

The basic agent behavior employs a classic *sense-plan-act* model [6, 7]. At each time step in the simulation, an agent senses its environment, then decides what to do, and then does it. The agents can sense the following properties:

- am I holding food?
- am I "at" food (i.e., on the same patch as a piece of food)?
- distance to food
- distance to pile

They can sense the world around them within a specified radius. Their sensing function sorts the detected locations of food according to the agents' priority system and returns the coordinates of a single patch. A sensing (S) agent returns the closest patch whereas an intuitive (N) agent returns the patch with the largest surrounding cluster of nearby patches containing food. The agents can perform the following actions:

- move forward
- turn
- pick up food
- drop off food
- wiggle (turn randomly and move forward)

Personality preference axis: S versus N. An agent with a sensing (S) personality preference is concrete. It looks at proximity and focuses on what is closest. For example, it will move toward the closest food pile, even if it is small. This agent also looks at the past. It has a short-term (1 timestep) memory of what it saw in the past. In contrast, an agent with an intuitive (N) personality preference is more abstract. It looks at density and focuses on what is largest. For example, it will move toward the largest food pile, even if it is far away. This agent does not have any memory of the past.

Personality preference axis: J versus P. An agent with a judging (J) personality preference makes a decision about where to go and commits to its decision until it reaches its target location. It does not attempt to sense (perceive) the world again until the target is reached. In contrast, an agent with a perceiving (P) personality preference makes a decision about where to go and commits to it, but only for one timestep. After moving toward the target for one timestep, it perceives the world again and potentially changes its target if conditions dictate.

Pseudo code. The simulation is controlled by a main loop that iterates over a fixed number of timesteps[3]. Each iteration consists of calls to `sense()`, `plan()` and `act()` functions, one for each of the agents in the simulation. The differences between agent personality types are evident in the `sense()` and `plan()` functions, as detailed below. The `plan()` function generates a plan and the `act()` function executes the plan. The `act()` function is the same for all agents.

Figure 1 illustrates the perception functionality of the agents. Note that the term "perception" is used in the classic sense of agent-based or robotic systems, meaning that its execution causes the agent to use its sensors to evaluate its environment. For example, a robot might use its sonar to detect distance to obstacles. The `sense()` function correlates very well to N versus S. The intuitive agent looks at every piece of food in its radius of vision and calculates which patch is surrounded by the most food. The patch with the largest cluster is sent to the `plan()` function. On the opposite spectrum, the sensing agent calculates the distance between itself and each patch of food in its radius of vision. The closest patch to the agent is sent to the `plan()` function. The only time the J and P preferences affect the sensing function is when a decision has been committed to and is not yet complete. This is the case when an agent with a judging preference has already set a path in motion and completely bypasses the `sense()` function until it reaches its destination.

Figure 2 illustrates the planning functionality of the agents. The `plan()` function takes the inputs from the `sense()` function and decides how to proceed. The biggest difference in the `plan()` function is that if sensing agents are looking for either food or a pile and cannot see one, they rely on their memory to lead them backwards to where they came from. Intuitive agents do exactly the opposite: they try to explore new territory. This distinction emphasizes the *exploitation* versus *exploration* trade-off frequently discussed in the evolutionary computation and artificial life communities. Similar to the `sense()` function, if a judging agent has already made a decision and has yet to complete the task at hand, the decision step is completely bypassed. On the other hand, perceiving agents always re-evaluate their decisions.

[3] Note that the number of timesteps was fixed only for experimental purposes. Other termination conditions could be used.

```
function sense() {
  if ( not J ) or ( J and plan is empty ) {
    holdingFood <- am I holding food?
    atFood <- am I at food?
    if ( S ) {
      locFood <- location of closest food source
      locPile <- location of closest pile
    }
    else { // N
      locFood <- location of largest food source
      locPile <- location of largest pile
    }
}
```

Fig. 1. Pseudo code for agents' perception functionality

```
function plan() {
  if ( not J ) or ( J and plan is empty ) {
    if ( holdingFood )
      if ( distance to locPile = 0 )
        plan <- put food down
      else // not at pile
        plan <- go toward locPile
    else // not holding food
      if ( atFood )
        plan <- pick up food
      else
        if ( I can see food )
          plan <- go toward locFood
        else
          if ( S )
            plan <- go toward last location where food was found
          else // N
            plan <- go toward a new (unexplored) location
}
```

Fig. 2. Pseudo code for agents' planning functionality

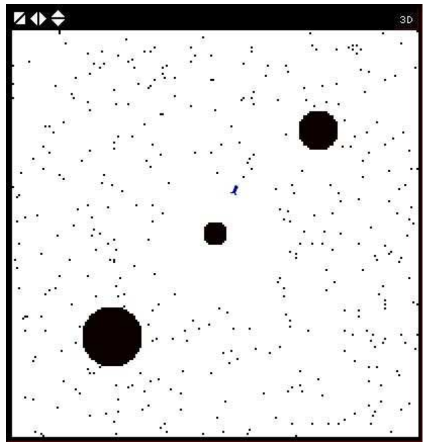


Fig. 3. Sample screen shot of "termite world"

3 Results

Our experimental system was implemented as a prototype in NetLogo [8]. An illustration of the "termite world" is shown in Figure 3. The small dots represent particles of food. The large circles represent food piles. A single agent is shown near the center, a very rough visual approximation of an insect. Experiments were run with different sets of agent populations consisting of five different personality types: SJ, SP, NJ, NP, and random (for comparison). Each experimental condition was run for 1000 timesteps. To illustrate the model and set a baseline for multiagent sets, Section 3.1 shows the results for experiments in which one agent of each type was simulated. These results were discussed in detail in [9]. Section 3.2 describes the new results, for homogeneous teams of agents, and Section 3.3 presents the new results for heterogeneous teams.

3.1 Single Agent Results

The first set of experiments replicated the results from [9] and are shown here as the basis for comparison with the new results presented in the rest of this section. The agents' world is a 200×200 patch arena. Five scenarios were run, each with one agent of each type. The agent started each run in the center of the arena.

Table 1 contains average values and standard deviation (in parenthesis), over 4 experimental runs. The first data column shows the average number of food particles collected and deposited in piles. A higher number is better. The SJ agent delivered the most food particles. The second data column shows the length of the path traveled. A lower number means that the agent did not explore much territory, whereas a larger number means that the agent explored more. The

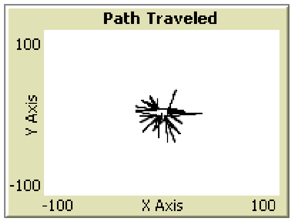
Table 1. Single agent, 200x200 world

	food delivered		path length		path efficiency	team size	team efficiency
SJ	**23.25**	(2.22)	764.25	(22.87)	32.87	1	**23.25**
SP	12.00	(2.00)	309.50	(3.70)	**25.79**	1	12.00
NJ	11.75	(0.96)	**880.50**	(9.57)	74.94	1	11.75
NP	2.25	(1.50)	329.25	(3.20)	146.33	1	2.25
random	0.25	(0.50)	332.50	(1.00)	1330.00	1	0.25

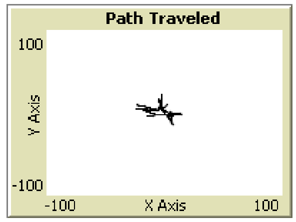
larger number is better, because it shows that the agent covered more of its environment; since the food particles do not move during the simulation (unless the agent moves them), the agent will only be able to gather more food particles if it also explores more area. The NJ agent traveled the furthest. The third data column shows the agent's "path efficiency". This divides the length of the path traveled by the number of food particles delivered, producing a value that indicates how much the agent was able to accomplish given its effort expended. Lower values are better. The SP agent is the most efficient. This is because it goes to the closest food location from its current position, so it does not spend a lot of time wandering around. The fourth data column contains the team size, i.e., the number of members on the team. In the case of these single agent baseline runs, the size of the team is, of course, 1; but the table format is used throughout this section, so this columns is included for consistency.

The final data column shows the "team efficiency". This is an indication of how efficiently the team members perform as a group. It is calculated as the total amount of food delivered by the team divided by the number of team members. For example, if a team of 4 agents can deliver 100 particles of food in the same time that a larger team of 10 agents delivers the same amount of food, then the first team is considered more efficient; i.e., $(100/4 = 25) > (100/10 = 10)$. Higher values are better. In the case of a single-agent team, the team efficiency is the same as the amount of food delivered (shown here in the first data column); but again, the table format is used for consistency, to enable easy comparison with the tables that appear later in this section.

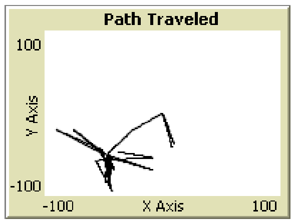
Figure 4 shows the paths each agent in a single agent experiment takes while collecting food and bringing it back to the piles. The agents' actions create straight or squiggly lines, depending on their approach and commitment. This figure illustrates where each agent's focus lies. For both sensing agents, SJ and SP, Figure 4b and 4c, respectively, their paths are short and they do not stray far from their starting point. The graphs illustrate how the focus of sensing types is based on proximity and that they prefer to concentrate on the details in front of them. On the other hand, intuitive types tend to focus on the bigger picture and try to look for patterns or clusters. Their paths are typically longer because they are willing to travel further out to find the largest cluster of food. Notice how both the NJ and the NP do not stay near their starting points for long. They are quickly pulled towards the largest pile. This again illustrates how the N's focus is not defined by proximity, but cluster size.



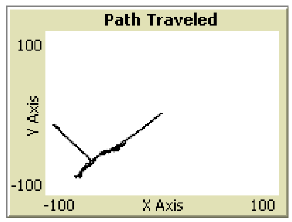
(a) SJ



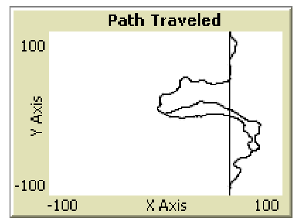
(b) SP



(c) NJ



(d) NP



random

Fig. 4. Typical paths taken by each agent personality. Different path shapes and lengths reflect different decisions about where to go.

Looking at both the NJ's and NP's paths side by side and the SJ's and SP's paths side by side, it is also clear that aside from the length of their paths, there are other differences between the types. The other differences can be attributed to the judging and perceiving function. As explained in Section 1, judging types prefer to make a decision and commit to it. Perceiving types prefer to continue researching and are not committed to their decisions. Looking at the NJ's and NP's paths, we can see that the NJ's paths taken are all straight, whereas the NP's paths are mixed with both straight and squiggly lines. This shows how the NJ senses for food, is able to find the largest cluster of food within its line of sight and makes a decision of where to go. The agent continues in a straight line till arriving at its destination. On the other hand, the NP re-evaluates its path at every step. Since moving forward may bring new information about the largest cluster, the old decision is no longer valid. The re-evaluation and continuous research is illustrated by the squiggly path. To think of it a little differently, the NP first tries to find the largest cluster that exists in its environment. The NJ looks for the local maximum, where "local" is defined by its line of sight.

Having explained the differences between each of the types, it is not only understandable but expected that each agent type should perform differently. According to our experiments, the SJs collected the most food with SPs in second place, NJs in third, NPs in fourth and the random agent coming in last. Regardless of which agent came in first and which last, their functions are consistent with their types. Assuming they can see food, both agents with judging preferences have regular intervals between each return trip to the pile. The SP's time between trips gets increasingly longer as it is forced to travel farther and sense the world more frequently. Since the NP is always looking for clusters the interval is dictated by how far it has to travel between each cluster and when it decides it has found the largest cluster.

Table 2 shows baseline results with the new system, modified from [9] to accommodate multi agent teams. These baseline results are, as above, for single agents; the data is averaged over 16 runs. The agents' world is smaller, 100×100 patches. Each agents' visual radius is also smaller, 30 patches long (as opposed to 50 patches in the original model). This allows the agents to collect nearly all the food in the environment within the alloted time and illustrates the Sensing type's memory and the Intuitive type's interest in undiscovered territory. Two starting conditions were simulated for each agent type: first, starting the agent at the origin and second, starting the agent at a random location in its world. The first line in each pair for each agent type is the first condition; the second line is the second condition. The data shown is averaged over 16 runs for each starting condition and each agent type. The differences between the two starting conditions are negligable.

Table 2. Single agent, 100x100 world

	food delivered		path length		path efficiency	team size	team efficiency
SJ	**28.25**	(1.53)	632.04	(83.44)	22.37	1	**28.25**
	26.81	(5.09)	601.12	(97.31)	22.42	1	26.81
SP	14.75	(0.86)	302.97	(31.91)	**20.54**	1	14.75
	13.56	(2.00)	294.43	(52.11)	21.71	1	13.56
NJ	18.25	(1.53)	**656.67**	(35.07)	35.98	1	18.25
	15.94	(6.38)	594.40	(127.86)	37.30	1	15.94
NP	3.25	(1.53)	311.44	(32.82)	95.83	1	3.25
	3.13	(1.09)	294.99	(30.76)	94.40	1	3.13
random	0.56	(0.51)	308.41	(23.59)	548.29	1	0.56
	0.69	(0.79)	295.93	(26.99)	430.44	1	0.69

The results are consistent with the original experiment: the SJs collected the most food, and the NP and random agents collected the least. The SP and NJ agents collect similar amounts of food, with the former slightly edging out the latter in the original experiments, and the order reversed in the replicated experiments. The difference in the size of the world and the smaller vision radius accounts for these differences. The NJ agent travels the furthest. The SP agent has the best path efficiency, while the SJ agent displays the best team efficiency.

3.2 Multiagent Experimental Results, Homogeneous teams

Table 3 contains results from multi agent simulations. Each simulation contains a homogeneous team of 5 members. Runs were conducted using the same two starting conditions, as above: starting at the origin and starting in a random location. The first line in each pair for each agent type is the first condition; the second line is the second condition. The data shown is averaged over 16 runs for each starting condition and each agent type. Other than the agent starting positions, the 16 runs also differed by the locations of food patches in the environment. The differences between the two starting conditions are negligible.

Table 3. Multi agent, homogeneous, starting at origin, 100x100 world

	food delivered		path length		path efficiency	team size	team efficiency
SJ	15.60	(1.65)	**649.13**	(39.73)	41.61	5	3.12
	15.71	(1.45)	599.43	(32.07)	38.15	5	**3.14**
SP	9.80	(0.53)	280.50	(12.57)	28.62	5	1.96
	10.18	(0.54)	277.35	(16.87)	**27.26**	5	2.04
NJ	12.90	(1.00)	614.08	(49.44)	47.60	5	2.58
	12.95	(1.32)	613.42	(53.39)	47.37	5	2.59
NP	2.00	(0.57)	294.48	(14.65)	147.24	5	0.40
	2.21	(0.44)	283.10	(16.78)	127.96	5	0.44
random	0.68	(0.23)	288.99	(11.58)	428.13	5	0.14
	0.64	(0.28)	304.21	(16.34)	477.20	5	0.13

Although the results between starting at the origin and random positions are negligible on average, the differences illustrate the competitiveness of the environment and agents. For example, by placing five SJ agents at the origin at the same time, they will all view the exact same starting world and focus on the same piece of food. As we explained in Section 2, the Judging preference means that each agent will set a plan and not sense again until it reaches its destination. In other words, each of the agents will target the same exact piece of food, but only one agent will actually get it. The second agent will realize there is nothing to pick up and will sense the world again.

Table 3 shows the results of five different experiments. Each experiment has 5 agents of the same personality type. Notice that the results are similar to our findings for the single agent environment. SJs still collect the most food, and SPs are the most efficient for each step they take. As we explained in Section 3.1, this makes sense even with the competition of a multiagent system. Since SJs are committed to their plan, they only sense the world and make decisions when they do not have a target. Although they compete with other like agents, once they are far enough away from the other agents in the space, they do not miss many opportunities. SPs remain the most efficient because as soon as the environment changes they are aware of those changes. This means that in a very competitive space, they immediately see the change and update their target accordingly.

Although homogenous sets of agents collect a lot of food, it is interesting to see how each agent type on average collects less, compared to the single agent environment; this difference is reflected in the disparate values for "team efficiency" in Table 3 as compared to Table 2. This is because a single SJ agent can focus on what it does best, without any distractions and without the environment changing. In a competitive space, like agents start by focusing on similar targets; e.g., all 5 SJs try to collect nearby food first. When the 5 agents deplete nearby food, they are forced to look farther out; and they now operate like an NJ does, traveling longer distances before picking up food. Although one might think that starting at random positions would lessen the impact on the mean, in practice, it does not. The reason is that our environment drives agents back to a nest once they have collected food, resetting the gains made by starting at a random position.

3.3 Multiagent Experimental Results, Heterogeneous teams

As we described in Section 3.2, homogenous sets of agents are extremely competitive, thereby diminishing the overall productivity of the team. In this section we explore different combinations of agent types working in teams, to try and maximize the amount of food collected by a team of agents and the team effectiveness. To demonstrate that not all agents are competitive, Table 4 shows that in an environment with only two agents, one SJ and one NJ, since their foci are different, their overall perfomance is similar to that of a single agent environment.

Table 4. Multi agent, heterogenous (1 SJ, 1 NJ)

	individual				*team*	
	food delivered		path length	path efficiency	size	efficiency
SJ	27.00	(1.93)	681.47 (39.54)	25.24	2	22.88
	25.75	(3.92)	618.50 (67.39)	24.02	2	21.16
NJ	18.75	(1.13)	659.33 (64.90)	35.16	2	22.88
	16.56	(2.53)	675.35 (86.45)	40.78	2	21.16

If we extend this idea, we see that diverse teams of agents are able to collect more food per agent than homogenous teams. In the above example, a single SJ and NJ on the board produces a total team efficiency of 22. Over the 231 different heterogenous experiments we ran, the five groupings that distributed the work of collecting food the most efficiently are shown in Table 5. Notice that the groupings with the highest team efficiencies do not always collect the most food.

Instead if we focus on the most food collected, Table 6 shows the top 5 heterogenous sets of agent populations. As discussed earlier, the competitiveness of each similar agent brings the team efficiency down. As is the ultimate question in many projects when deciding if more resources are necessary or more time, we show here that putting more agents to the task does not boost the

Table 5. Top 5 team efficiency

SJ	SP	NJ	NP	random	food collected	team size	path efficiency	team efficiency
1	0	1	0	0	45.75	2	29.31	22.88
1	1	0	0	0	37.50	2	24.64	18.75
1	1	1	0	0	49.75	3	32.09	16.58
0	1	1	0	0	33.00	2	29.81	16.50
1	0	1	1	0	48.00	3	33.73	16.00

overall performance. In this environment we might have simply collected more food by extending the time limit instead of adding more that one agent of each personality type.

Table 6. Top 5 food collection

SJ	SP	NJ	NP	random	food collected	team size	path efficiency	team efficiency
5	5	5	5	0	94.50	20	81.85	4.73
5	5	5	0	0	94.25	15	67.51	6.28
5	5	5	1	0	94.00	16	73.05	5.88
5	5	5	5	1	93.63	21	84.86	4.46
5	5	5	0	1	92.81	16	71.64	5.80

Finally, we examine path efficiency, to determine which groupings produce agents that explore the space effectively. Table 7 lists the five groupings with the best (lowest) path efficiency. It is interesting to see that these teams have reasonable team efficiency values, including the most efficient team with a value of 22.88; however, these teams are in the bottom third in terms of the amount of food collected. These results highlight the conclusion that agents with different personality types can be shown to behave differently in a simulated environment. Future work involves categorizing heterogeneous groupings of agents according to their ability to accomplish particular tasks.

4 Related work

There is a fair amount of research into the use of personality types in agent-based systems. Most approaches focus in one of two directions. The first, more prevalent focus is on creating personalities for agents that interact with human users in social environments. In these cases, the research involves encoding personality type or temperament to increase social acceptance. Dryer [10] explains that personality types can be used to enhance human-machine interaction. Lin and McLeod [11] introduce personality into their work, but instead of incorporating

Table 7. Top 5 path efficiency

SJ	SP	NJ	NP	random	food collected	team size	path efficiency	team efficiency
1	1	0	0	0	37.50	2	24.64	18.75
1	0	0	1	0	31.25	2	29.05	15.63
1	0	1	0	0	45.75	2	29.31	22.88
1	1	0	1	0	41.50	3	29.33	13.83
0	1	1	0	0	33.00	2	29.81	16.50

type as the part of the mechanism underlying agents' actions, they train their engine to recognize temperaments and information associated with each temperament. They use this training to filter results more effectively and provide better recommendations. Allbeck and Badler [12] use the "Big Five" theory to embody personality traits and make the motions of each agent flow more realistically and believably.

Lisetti [13] defines a taxonomy for socially intelligent agents, stressing *emotion* as a strong component of personality. She describes state machines that illustrate how an agent can shift from one emotion, such as "happy", to another emotion, such as "concerned". These shifts can occur for different reasons in agents with different personality types. For example, a "determined" agent that is "frustrated" may shift into an "angry" state and use that anger to work itself back into a "happy" state; whereas a "meek" agent may shift from "frustrated" to "discouraged" and never return to "happy".

The second focus is on modeling complex interactions between agents and their environment and describing variations in agent behaviors as personalities. Castelfranchi *et al.* [14] present a simulation framework called "GOLEM" in which agents of different personality traits are modeled. GOLEM provides an experimental framework for exploring the effect of personality traits on social actions, such as delegation. Agents develop models of each other, labeled as personality traits, and use these models to motivate their interactions. Talman *et al* [15] model personality along two axes: "cooperation" and "reliability". These different traits are implemented in a logical framework where agents play a game and reason about each others' "helpfulness", or lack thereof. Agents can recognize different personality types and respond effectively, customizing their actions appropriately for different personalities.

Both of these last two examples use the notion of personality as a means for agents to model each other and make decisions about how to effect (or not) cooperative activity with others. Another approach is given in [16] where personality is closely tied to emotion, as with the first type of focus listed above. In this work, agents' internal decision-making processes are guided by personality types. Agents are deployed in a simulated military combat scenario in which factors such as "cowardice" and "irritability" are modeled and act as motivators for certain types of actions. For example, an agent labeled as cowardly may be driven by fear and run away from threats when attacked; whereas an agent driven by anger might move forward and face the enemy.

All of the work discussed above is highly context dependent: personality traits are designed in tandem with the environment in which agents are simulated and the tasks that agents are addressing. The advantage of the MBTI model is that it is generic and can, in theory, be adapted to any environment and task. While the instantiation details of agents' personalities will necessarily be tailored to a particular environment, the abstract definition of the personality traits themselves is not specific.

Campos *et al.* [17] is the most closely aligned with our work, mainly because of their use of the MBTI model to leverage personality type and test agent performance in the same environment with different personalities. Similarly, the authors also started with two axes to illustrate personality, though they chose the S-N and T-F dichotomies. Even with our implementation of the S-N function we differ. Campos *et al.* implemented the dichotomy as a mechanism for developing a plan, a hybrid between the S-N and J-P dichotomies. We instead use the S-N function to weight inputs and allow the J-P function to develop the plan.

5 Summary

In the work presented here, we have shown how each personality type functions, illustrating the differences between them and explaining the factors that drive the differences. Since our goal was to see which personality type collected the most food within a given timeframe, we were able to conclude that the SJ personality type is the "winner". In proving that one personality type outshined the others, we are able to conclude that different personality types are in fact better for different tasks—at least in this highly simplified example.

Our next step is to enhance the agent model to include all four MBTI axes, producing a total of 16 personality types. As mentioned in Section 1, here we only looked at the two preferences that did not require other agents. Once we include the extroverted (E) and introverted (I) preferences, we aim to demonstrate how some personality types are better suited to working alone and others are better suited to working with others. Including the thinking (T) and feeling (F) preferences should illustrate how certain agents are more empathetic than others and may be better suited for missions that involve helping others, such as robot-assisted search and rescue (e.g., [18]).

Once the model is expanded to simulate all sixteen types, different and more complex environments and tasks will be explored in order to illustrate the differences between personality preferences. Group dynamics will also be examined, where the interactions of agents with different personality types can be shown to bring complexity to coordination even in groups that have previously been seen as homogeneous—because personality types were not implemented. We will then be able to test different combinations of heterogeneous agent groupings to see which groups work most efficiently together for which types of tasks.

References

1. Myers, I.B., Myers, P.B.: Gifts Differing. Consulting Psychologists Press (1980)
2. Jung, C.: Psychological types. In: The collected works of C. G. Jung. Volume 6., Princeton, NJ, Princeton University Press (1971, originally 1921)
3. Bratman, M.E., Israel, D.J., Pollack, M.E.: Plans and resource-bounded practical reasoning. Computational Intelligence **4**(4) (1988) 349–355
4. Kinny, D., Georgeff, M.: Modelling and design of multi-agent systems. Intelligent Agents III (LNAI) **1193** (1996) 1–20
5. Resnick, M.: Turtles, Termites and Traffic Jams: Explorations in Massively Parallel Microworlds. MIT Press, Cambridge, MA, USA (1994)
6. Nilsson, N.J.: Technical note no. 323. Technical report, SRI International, Menlo Park, CA (1984) This is a collection of papers and technical notes, some previously unpublished, from the late 1960s and early 1970s.
7. Brooks, R.A.: New approaches to robotics. Science **253**(5025) (13 September 1991) 1227–1232
8. Wilensky, U.: NetLogo. `http://ccl.northwestern.edu/netlogo/` (1999)
9. Salvit, J., Sklar, E.: Toward a Myers-Briggs Type Indicator Model of Agent Behavior. In: Multi-agent Based Simulation (MABS) Workshop at Autonomous Agents and MultiAgent Systems (AAMAS). (2010)
10. Dryer, D.C.: Getting personal with computers: how to design personalities for agents. Applied Artificial Intelligence **13** (1999) 273–295
11. Lin, C.H., McLeod, D.: Temperament-based information filtering: A human factors approach to information recommendation. In: Proceedings of the IEEE International Conference on Multimedia & Exposition, New York (2000)
12. Allbeck, J., Badler, N.: Toward representing agent behaviors modified by personality and emotion. In: Proceedings of the Workshop on Embodied Conversational Agents at the 1st International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Bologna (2002)
13. Lisetti, C.L.: Personality, Affect and Emotion Taxonomy for Socially Intelligent Agents. In: Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference (FLAIRS'02), AAAI Press (2002)
14. Castelfranchi, C., de Rosis, F., Falcone, R., Pizzutilo, S.: A Testbed for investigating personality-based multiagent cooperation. In: Proceedings of the Symposium on Logical Approaches to Agent Modeling and Design. (1997)
15. Talman, S., Gal, Y., Hadad, M., Kraus, S.: Adapting to Agents' Personalities in Negotiation. In: Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), ACM (2005)
16. Parunak, H.V.D., Bisson, R., Brueckner, S., Matthews, R., Sauter, J.: A Model of Emotions for Situated Agents. In: Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), ACM (2006)
17. Campos, A., Dignum, F., Dignum, V., Signoretti, A., Mag'aly, A., Fialho, S.: A process-oriented approach to model agent personality. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Budapest (2009) 1141–1142
18. Kitano, H., Tadokoro, S.: RoboCup Rescue: A Grand Challenge for Multiagent and Intelligent Systems. AI Magazine **22**(1) (2001)