

# A Strategy Video-Game for Collaborative Agents with a Personality and Humoristic Dialogues

Context Survey

Jordan Mackie  
*Student*

Alice Toniolo  
*Supervisor*

Christopher Stone  
*Supervisor*

**Abstract**—In this document, we provide a background for the areas of research related to the project, and some discussion of the tools and technologies that will be used. After describing the goals and results of previous research, we identify the components that could be reused or built upon for the purposes of building a system of collaborative agents with personality driven decisions and humoristic dialogues, and why the results of this project could be useful for areas such as human-computer interaction, modelling, and entertainment.

## I. INTRODUCTION

Multiagent systems are the next step to increase the level of autonomy that can be provided by technology. Agents that are able to learn, adapt, and negotiate with other agents to achieve their goals allow for complex problems to be solved or modelled, such as monitoring and maintaining national power grids [1].

Often, developers and users will anthropomorphise these agents when describing their behaviour. This project aims to encourage this by implementing agents with a model of personality that affects their choice of actions, by rendering the negotiations between agents in a natural language, and by using models of humour to make the interactions between agents entertaining.

## II. SURVEY

### A. Multiagent Systems in Entertainment

Many modern video games involve the user managing multiple characters to achieve some goal, such as producing in-game resources or defending against an opponent. This sort of problem lends itself easily to multiagent systems. Instead of having one artificial intelligence engine driving the actions of all the characters, developers can create agents with a limited set of actions and some concept of progress towards their goals and allow emergent behaviour find a solution to the problem, sometimes in surprising ways.

[2] discuss how multiagent systems can be applied to create more realistic worlds in sandbox games. By implementing the environment, objects, and non-playable characters (NPCs) as agents, developers can create a world that reacts and adapts the player, but also operates in isolation from the character to provide a realistic setting. The concept of personalities is also mentioned as a way of allowing similar NPCs to

exhibit different slightly different behaviours, such as having aggressive or relaxed driving styles.

Even in games with very simple rules and logic, multiagent systems can find interesting and complex solutions. OpenAI implemented hide-and-seek using agents, where hiders avoid the line-of-sight of the seekers [3]. The game was played in a world with randomly generated walls and objects such as ramps (for climbing over walls) and blocks (for forming barricades). What made this fascinating is that the agents were not incentivised to use these objects, but after repeatedly playing and learning, both the hiders and seekers created strategies such as blocking each other in a safe area, and even removing the objects from the other team before hiding.

Knowing that multi-agent autocurricula can realise strategies not considered by humans, [4] discuss how they can be applied to strategy board games such as Diplomacy and Risk. They describe a generic framework for supporting agent-based competitive bots for board games and then implement bots for the aforementioned games. Their research suggests that for games with a large number of units and a large action space, a multi-agent approach can identify effective strategies quicker than the exhaustive methods used for chess engines.

### B. Models of Personality and Emotion

Creating a truly immersive video game requires characters that the player can empathise with. Robots have been shown to be able to influence human behaviour as an authority figure [5] and when begging not to be turned off [6] by expressing emotions. [7] created and demonstrated a model of personality and emotion that would allow agents to react differently to the same stimulation. For example, when a agent with an 'introverted' personality is offered help, they are less likely to accept it due to the prolonged interaction it would entail.

[8] also created a model that would use personality, emotion, and social relationships to determine the behaviour of NPCs in a video game. The frequency and tone of interactions between NPCs as well as the NPC and the player were accounted for when choosing facial expressions and tone of voice during conversations. Test subjects described feeling especially attached to the NPCs that utilised this model showing.

A useful aspect of multiagent systems is that agents can be developed in isolation but still interact (e.g. an agent that searches for cheap transport options and an agent responsible

for auctioning train tickets could be developed separately with no knowledge of the logic being used by the other). [9] discuss how developing heterogeneous agent systems using personalities and social structures could help when dealing with third-party agents that have been constructed to lie and exhibit selfish or uncooperative behaviour.

### C. Collaborative Argumentation

Knowledge is distributed in a multiagent system therefore specialist agents need to be able to alter the beliefs of others by appealing to their individual goals. [10] implemented a framework that achieves this based on a scheme given by [11]:

In the current circumstance R, we should perform action A, which will result in new circumstances S, which will realise goal G, which will promote some value V.

Specifically, they were able to create a framework for multiple parties to discuss and collaborate which could greatly affect the design of multiagent systems that utilise it. By producing an ontology which any agents involved in the discussion understand, goals and circumstances can be conveyed through the use of predicates and concepts.

The overhead of argumentation required in multiagent systems can be a quick filter to determine which problems it is a suitable solution for. For example, [12] investigated how quickly agents that used collaborative argumentation to achieve global consistency in a time-constrained task (i.e. escaping a burning building) performed. Their results suggest that optimisations or different approaches to the protocol design would likely be necessary for a real-time strategy game, but it could depend on many factors such as the number of agents, the distribution of knowledge, and more.

### D. Natural Language Generation

In order to make the discussions between agents more entertaining and user-friendly to observe, our project involves using natural language generation (NLG) methods to translate the messages to plain English. [13] built an NLG system for a particular ontology language (W3C Ontology Language a.k.a. OWL), which is able to construct texts corresponding to objects and their properties in the ontology in English and Greek. [14] achieved similar results for one of the other Semantic Web Formalisms (RDF) and were able to produce instructions for cooking recipes from a corpus of data in a far less human-friendly format. They were even able to adjust the level of technical jargon in the resulting text to accommodate for the readers familiarity with cooking. The ontology allowed for optimised searching of the parse trees but they also required a domain corpus (i.e. existing textual recipes) in order to properly train their natural language generator.

Generic frameworks have also been created that are not based strictly on ontologies which may also be useful. For example, SimpleNLG [15] is an NLG engine that allows for valid English phrases to be constructed through a very simple Java API.

### E. Computational Humour

Simply reading the discussion between agents would certainly not make for an entertaining game, and so adjusting the dialogue or agent behaviour is required. However, artificial humour also has benefits for everyday computing: a smartphone that is able to sympathise with the user when it is unable to connect to a weak WiFi network or cheer them up with a joke when they miss their bus is one that the user is more likely to enjoy using.

Unfortunately, given humour is an incredibly contextual and culture driven concept, it is no surprise that computational humour is well researched yet still not 'solved'. Jason Rutter describes why using artificial intelligence for humour is particularly challenging:

"Humour is a very interesting way to look at artificial intelligence because at some point something has to have two meanings, which is not easy to do with a computer." [16]

There are three main theories of humour that are used for computational humour: superiority theory (laugh about the misfortune of others), relief theory (using taboo subjects to release tension), and incongruity theory (using lexical and structural ambiguity). The last theory is currently the most popular, and was used during the development of JAPE [17]. JAPE identifies features such as homophones to produce jokes such as:

"What is the difference between leaves and a car?  
One you brush and rake, the other you rush and brake."

However, JAPE does not allow for much user interaction, and most textual humour generation takes the form of simple puns and riddles. Our agents could instead generate situational humour by surprising the user by diverting unexpectedly from expected behaviour. The Suslov Model of humour accounts for the fact that we subconsciously predict expected conclusions to situations and phrases, and that contradictions in what is the most likely direction for a conversation or situation to what was previously predicted can create a humour response [18].

These probabilistic models currently appear more hopeful than the rule based models used by older implementations such as JAPE. Instead, [19] used an existing corpus of jokes and a recurrent neural network to produce a system capable of generating jokes and even anti-jokes (jokes without an actual punchline).

## REFERENCES

- [1] Thies Wittig, editor. *ARCHON: An Architecture for Multi-agent Systems*. Ellis Horwood, Upper Saddle River, NJ, USA, 1992.
- [2] Sergio Ocio and José Antonio López Brugos. Multi-agent systems and sandbox games. 2009.
- [3] Anonymous. Emergent tool use from multi-agent autocurricula. In *Submitted to International Conference on Learning Representations*, 2020. under review.
- [4] Stefan J. Johansson. On using multi-agent systems in playing board games. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, pages 569–576, New York, NY, USA, 2006. ACM.

- [5] Derek Cormier, Gem Newman, Masayuki Nakane, James Everett Young, and Stephane Durocher. Would you do as a robot commands ? an obedience study for human-robot interaction. 2013.
- [6] Aike C Horstmann, Nikolai Bock, Eva Linhuber, Jessica M. Szczuka, Carolin Straßmann, and Nicole C. Krämer. Do a robots social skills and its objection discourage interactants from switching the robot off? In *PloS one*, 2018.
- [7] Arjan Egges, S Kshirsagar, and Nadia Thalmann. A model for personality and emotion simulation. pages 453–461, 01 2003.
- [8] Andry Chowanda, Peter Blanchfield, Martin Flintham, and Michel Valstar. Computational models of emotion, personality, and social relationships for interactions in games. 05 2016.
- [9] Cristiano Castelfranchi, Fiorella de Rosis, Rino Falcone, and Sebastiano Pizzutilo. Personality traits and social attitudes in multiagent cooperation. *Applied Artificial Intelligence*, 12:649–675, 1998.
- [10] Elizabeth Black and Katie Atkinson. Dialogues that account for different perspectives in collaborative argumentation. pages 867–874, 01 2009.
- [11] Katie Atkinson and Trevor Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence*, 171(10):855 – 874, 2007. Argumentation in Artificial Intelligence.
- [12] Gael Hette, Gauvain Bourgne, Nicolas Maudet, and Suzanne Pinson. Hypotheses refinement under topological communication constraints. 01 2007.
- [13] Ion Androutsopoulos, Gerasimos Lampouras, and Dimitrios Galanis. Generating natural language descriptions from owl ontologies: The natural owl system. *J. Artif. Int. Res.*, 48(1):671–715, October 2013.
- [14] Philipp Cimiano, Janna Lüker, David Nagel, and Christina Unger. Exploiting ontology lexica for generating natural language texts from RDF data. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 10–19, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.
- [15] Albert Gatt and Ehud Reiter. SimpleNLG: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 90–93, Athens, Greece, March 2009. Association for Computational Linguistics.
- [16] Will Knight. Computer crack funnier than many human jokes. <https://www.newscientist.com/article/dn1719-computer-crack-funnier-than-many-human-joke/>, 2001.
- [17] Graeme Ritchie, Ruli Manurung, Helen Pain, Annalu Waller, Rolf Black, and Dave Mara. A practical application of computational humour. *Proceedings of the 4th International Joint Workshop on Computational Creativity*, 01 2007.
- [18] I. M. Suslov. Computer model of a "sense of humour". i. general algorithm, 2007.
- [19] Abhinav Moudgil. Humor generation with recurrent neural networks. <https://amoudgl.github.io/blog/funnybot/>, 2017.