A Mediator Agent based on Multi-Context System and Information Retrieval

Rodrigo Rodrigues, Ricardo Azambuja Silveira and Rafael De Santiago
Federal University of Santa Catarina - PPGCC, Trindade, Florianopolis, Brazil

Keywords: Intelligent Agent, Multi-Context Systems, Symbolic, Connectionist, Negotiation, Information Retrieval.

Abstract: Nowadays, decisions derived from intelligent systems frequently affect human lives (e.g., medicine, robotics, or finance). Traditionally, these systems can be implemented using symbolic or connectionist methods. Since both methods have crucial limitations in different aspects, integrating these methods represents a relevant step to deploying intelligent systems in real-world scenarios. We start tackling the integration of both methods by exploring how to use different types of information during the agent's decision-making. We modeled and implemented an intelligent agent based on a Multi-Context System (MCS). MCSs allow the representation of information exchange among heterogeneous sources. We use a framework called Sigon to implement the proposed agent. Sigon is a novel framework that enables the development of MCS agents at a programming language level. As a case study, we present a mediator agent for conflict resolution during negotiation. The mediator agent creates advice by retrieving information from the web and employing different data types (e.g., text and image) during its decision-making. This work provides a promising and flexible way of integrating different information and resources using MCS as the main result.

1 INTRODUCTION

Nowadays, decisions derived from intelligent systems frequently affect human's lives (e.g., medicine, education, or legal). There is an emerging need for understanding how AI methods execute these decisions (Goodman and Flaxman, 2017; Arrieta et al., 2019). Traditionally, two categories can separate AI methods: symbolic and connectionist. Symbolic AI works by carrying on a sequence of logic-like reasoning steps over a set of symbols consisting of language-like representations (Garnelo et al., 2016). On the other hand, connectionist AI refers to embodying knowledge by assigning numerical conductivities or weights to the connections inside a network of nodes (Minsky, 1991).

Even though connectionist techniques have helped AI achieve impressive results in many different fields, most of the criticism about this method revolves around data inefficiency, poor generalization, and lack of interpretability (Garnelo and Shanahan, 2019; Chollet et al., 2018). In a symbolic approach, we have an easily understandable and transparent system. However, they are known as less efficient (Arrieta et al., 2019; Anjomshoae et al., 2019). The question of how to conciliate the statistical nature of learning with the logical nature of reasoning, aiming to build such robust computational models integrating concept acquisition and manipulation, has been identified as a key research challenge and fundamental problem in computer science (Besold et al., 2017; Valiant, 2003).

Considering both methods' benefits to AI, many studies have focused on combining connectionist and symbolic approaches. The main goal is to increase intelligent systems' expressiveness, trust, and efficiency (Arrieta et al., 2019; Bennetot et al., 2019; Garnelo et al., 2016; Marra et al., 2019; Garcez et al., 2019). Some works focus on object representation and compositionality and how they can be accommodated in a deep learning framework (Garnelo and Shanahan, 2019). While others present a survey about how to employ reinforcement learning, dynamic programming, evolutionary computing, and neural networks to design algorithms for MAS decision-making (Rizk et al., 2018). Even though different surveys explore the integration of machine learning and agents (Jedrzejowicz, 2011), we notice that two crucial points were not fully covered: (i) - the usage of a neural network as part of the agent's reasoning cycle; (ii) - the integration of different types of information during the agent's decision-making.

Rodrigues, R., Silveira, R. and De Santiago, R.
A Mediator Agent based on Multi-Context System and Information Retrieval.
DOI: 10.5220/0010781400003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 2*, pages 78-87
ISBN: 978-989-758-547-0; ISSN: 2184-433X

We believe that the major challenges when deploying these systems in real-world environments are: (i) - the presence of different types of information (i.e text, audio, video, and image), where most of these different data are unstructured; and (ii) - in some cases only connectionist or symbolic methods could not suffice to produce robust Intelligent Systems to assist during problem resolution. Taking that into consideration, in this paper, our primary goal is to propose, model, and implement an intelligent agent that can reason under the presence of different data types and employ connectionist and symbolic methods during its reasoning cycle.

To achieve this goal, we propose an intelligent agent based on Multi-Context Systems (MCS). MCSs allow the representation of information exchange among heterogeneous sources (Cabalar et al., 2019; Brewka and Eiter, 2007; Brewka et al., 2011; Brewka et al., 2014). In MCSs, contexts describe different sources that interact with other contexts via special rules called bridge-rules (Cabalar et al., 2019). More precisely, we propose adding two custom contexts into a BDI-like agent. These contexts model resources responsible for reasoning under the presence of different types of information. We propose two new bridge-rules and changes to the agents' planning preconditions verification to integrate these custom contexts into the agent's decision-making. We modeled an agent's actuator by developing a web-scraper for information retrieval. Generally, Web data scraping can be defined as the process of extracting and combining contents of interest from the Web in a systematic way (Glez-Peña et al., 2014).

To develop our agent, we use a framework called Sigon. According to the best of our knowledge, Sigon is the first framework that enables the development of MCS agents in a programming language level (Gelaim et al., 2019). We present a case study in which our proposed agent acts as a mediator, responsible for solving conflict during a buyer and seller negotiation. To construct such solutions, a mediator brings more information and knowledge and, if possible, resources to the negotiation table (Trescak et al., 2014). The mediator agent's strategy revolves around retrieving information via its actuator and detecting emotions based on facial expression. In this scenario, we also show how these strategies can be modeled in an MCS.

This paper is organized as follows: Section 2 presents an overview about the related works. Section 3 presents the topics investigated in this research. Section 4 presents our agent model's initial proposal and how an actuator can be implemented as a web-scraper. Section 5 shows a case study and how this agent can be implemented. Finally, in section 6 a conclusion and future works is showed.

2 RELATED WORKS

Rodrigues et al. present in (Rodrigues et al., 2021) a Systematic Literature Mapping (SLM), reporting an overview about the integration of neural network and intelligent agents. From 2015 to 2020, 1019 papers were analyzed. One of the most important findings is that most studies use neural networks to define learning agent's reward policies, leaving uncovered the integration of neural networks as part of the agent's decision-making.

We start exploring how to integrate symbolic and connectionist methods by modeling an agent as a Multi-Context System (MCS). Many different approaches of MCS have been employed for interlinking heterogeneous knowledge sources (Cabalar et al., 2019; Dao-Tran and Eiter, 2017; Brewka and Eiter, 2007). MCS also were employed for modeling negotiating agents (Trescak et al., 2014; Parsons et al., 1998; de Mello et al., 2018). However, none of them explored the integration of different data types during the agent's reasoning cycle.

A framework called Sigon was created to fill the gap between theory and practice. According to the best of our knowledge, Sigon is the first programming language for developing agents as MCS (Gelaim et al., 2019). Sigon was already employed for modeling agents' situational awareness in urban environment (Gelaim, 2021), and the development of perception policies (De Freitas et al., 2019). Even though Sigon was used in those scenarios, the available version does not support modeling custom sensors for processing different data types (e.g., images, videos, and audio). Sigon also does not support referencing different contexts than the belief context during planning precondition verification. In our work, we started addressing these two limitations by changing the Sigon grammar and integrating custom sensors into the agent's reasoning cycle.

3 BACKGROUND

This section briefly introduces the topics used in our paper. Subsection 3.1 introduces the concept of neural networks. Subsection 3.2 presents the definition of intelligent agents. In subsections 3.3 and 3.4 we present the definitions of MCS and Sigon.

3.1 Neural Networks

Neural networks are models inspired by the structure of the brain (Ozaki, 2020; McCulloch and Pitts, 1990), which provides a mechanism for learning, memorization and generalization. These models can differ not only by their weights and activation function but also in their structures, such as the feedforward NN that are known for being acyclic, while recurrent NN has cycles (Ozaki, 2020). An artificial neural network consists of different neuron layer, where input layers form the NN, one or more hidden layers, and an output layer (Wang, 2003). Definition 1 is presented in (Kriesel, 2007) and models a simple neural network.

Definition 1. *An NN is a sorted triple (N,V,w) with two sets N, V and a function w, where N is the set of neurons and V a set $\{(i,j)|i,j \in \mathbb{N}\}$ whose elements are called connections between neuron i and neuron j. The function $w : V \to \mathbb{R}$ defines the weights, where $w((i,j))$, the weight of the connection between neuron i and neuron j, is shortened to w_{ij}.*

3.2 Intelligent Agents

Despite the existence of different definitions of intelligent agents, we assume that an agent has certain properties. An agent definition can have the following properties: autonomy, social skills, reactive, and proactive (Wooldridge et al., 1995). The agent's behaviour and properties can be determined by modelling its mental attitudes. In the *Belief-Desire-Intention* (BDI) architecture proposed by (Bratman, 1987), the three mental attitudes represent, respectively, the information, motivational, and deliberative states of the agents (Rao et al., 1995).

3.3 Multi-Context Systems (MCS)

A MCS specification of an agent contains three basic components: units or contexts, logics, and bridge rules (Casali et al., 2005). Thus, an agent is defined as a group of inter-connected units: $\langle \{C_i\}_{i \in I}, \Delta_{br} \rangle$, in which a context $C_i \in \{C_i\}_{i \in I}$ is a tuple $C_i = \langle L_i, A_i, \Delta_i \rangle$, where L_i, A_i, and Δ_i are the language, axioms, and inference rules respectively. A bridge rule can be understood as a rule of inference with premises and conclusions in different contexts, for instance:

$$\frac{C_1 : \psi, C_2 : \varphi}{C_3 : \theta}$$

means that if formula ψ is deduced in context C_1 and formula φ is deduced in context C_2 then formula θ is added to context C_3 (Casali et al., 2005). The information flows between contexts via bridge-rules. In section 3.4, we present how a BDI-agent can be modelled using Sigon framework.

3.4 Sigon: A Framework for Agent's Development

According to the best of our knowledge, Sigon is the first programming language for developing agents based on MCS. Sigon framework enables the development of agents components as contexts and defines its integration via bridge-rules (Gelaim et al., 2019). The definitions of a Sigon agent are presented in 2.

Definition 2 (Sigon BDI-agent)**.**

$$AG = \langle \{BC, DC, IC, PC, CC\}, \Delta_{br} \rangle, \tag{1}$$

where BC, DC, IC, PC, CC are the beliefs, desires, intentions, planning, and communication contexts; and Δ_{br} are the bridge rules for information exchange between contexts defined in 4, 5, and 6.

The beliefs, desires, and intentions context are modeled as a logical context, following the previously presented definition. The communication and planning contexts are modeled as functional contexts. The communication context consists of a set of sensors and actuators. A Communication context is defined as:

Definition 3. $CC = \langle \bigcup_{i=1}^{n} S_i, \bigcup_{j=1}^{m} A_j, \rangle$

where S_i with $1 \leq i \leq n$ are agent sensors, and A_j with $1 \leq j \leq m$ its actuators (Gelaim et al., 2019). Plans and actions form the planning context. Sigon's plans and actions are based on Casali et al. (2005) work. An action is defined as:

$$action(\alpha, Pre, Post, c_a) \tag{2}$$

where α is the name of the action, *Pre* is the set of pre-conditions for α execution, *Post* is the set of post-conditions, and c_a is the α cost (Gelaim et al., 2019). A plan is defined as:

$$plan(\varphi, \beta, Pre, Post, c_a) \tag{3}$$

where φ is what the agent wants to achieve, β is the action or the set of actions the agent must execute to achieve φ, *Pre* is the set of pre-conditions, *Post* is the set of post conditions, and c_a is the cost (Gelaim et al., 2019). Bridge-rules Δ_{br} are defined as follows:

$$\frac{CC:\ sense(\varphi)}{BC:\ \varphi} \tag{4}$$

$$\frac{DC:\ \varphi \textit{ and } BC:\ \textit{not } \varphi \textit{ and } IC:\ \textit{not } \varphi}{IC:\ \varphi} \tag{5}$$

$$\frac{\begin{array}{c} p = \ plan(\varphi, \beta, Pre, Pos, c_a) \\ PC:\ plan(\phi, \beta, Pre, Pos, c_a) \textit{ and } IC:\ \phi \textit{ and } \ BC:\ Pre \end{array}}{CC:\ \beta} \tag{6}$$

Sigon framework provides a BDI algorithm that can be used during the agent's development. Initially, an agent perceives data from the environment and then executes the bridge-rule presented in definition 4. This first bridge-rule adds the perception captured by the sensors of the communication context (CC) to the beliefs context (BC). According to definition 5, the second bridge-rule is responsible for choosing an intention that the agent wants to achieve. An intention is added when the agent does not believe it, does not have it as an intention, and desires it. The third bridge-rule presented in 6 selects an action to be executed. An action β is selected when the plan's precondition *Pre* is satisfied in the beliefs context (BC), and *phi* is true or can be inferred in the intentions context (IC) (Gelaim et al., 2019). For more details about Sigon implementation, we encourage the reader to access (Gelaim et al., 2019).

4 PROPOSAL

In this section, we present the proposed agent. This agent can process different types of perceptions during its reasoning cycle. The agent is modeled as a Multi-Context System, in which different contexts can represent heterogeneous knowledge sources. Definition 4 shows the agent's modeling as a Multi-Context System (MCS).

Definition 4 (Proposed agent as an MCS)**.**

$$AG = \langle \{BC, DC, IC, PC, NNC, AC, CC\}, \Delta_{br} \rangle, \tag{7}$$

where BC, DC, IC, PC, NNC, AC, CC are the beliefs, desires, intentions, planning, neural network, auxiliary, and communication contexts; and Δ_{br} are the bridge-rules for exchanging information between contexts.

We extended a BDI-like agent and added two new custom contexts. The neural network context is responsible for processing perceptions representing images. Since the neural network does not always provide an output with high accuracy or that can be used during the agent's reasoning cycle, we defined an auxiliary context responsible for mitigating the imprecision generated from the neural network's output. Figure 1 presents the initial version of the agent proposed in this work. In subsections 4.1 and 4.2 we present in more details how we implemented the agent's contexts and bridge-rules.

4.1 Modeling a Custom Communication Context

In Sigon, a communication context is responsible for creating an interface between the agent and its environment (Gelaim et al., 2019). Since one of the primary goals of our work is to provide flexible ways of processing different data types, the communication context should be able to process these data and create new perceptions to be used during the reasoning cycle. To achieve this goal, we present a method of integrating different data types by defining custom sensors. Each sensor defines how the data is processed and how the data is passed to the communication context. This method is based on software engineering design patterns, more precisely, the decorator pattern. The decorator pattern attaches additional responsibilities to an object dynamically (Kassab et al., 2018).

After defining which plan should be executed, an agent must perform a set of actions. In this work, we model the agent's actuator as a web-scraper application. In this process, a software agent, also known as a Web robot, mimics the browsing interaction between the Web servers and the human in a conventional Web traversal (Glez-Peña et al., 2014). This web-scraper main goal is to extract the required information and generate new perceptions that the agent should process. This strategy permits the agent to improve its decision-making by expanding its knowledge about the environment.

Listing 1 presents the Sigon syntax for defining an agent's sensors and actuators. The modules *Image* and *WebScraper* are responsible for mapping an observation to perception and modeling an action, respectively. The web-scraper actuator modeled in this work can extract new information and provide new perceptions to the agent. In subsection 4.2, we present how these different perceptions can be integrated into the agent's reasoning cycle.

```
communication:
    sensor("imageData", "perception.Image").
    actuator("findData", "actuator.WebScraper").
```

Code 1: Sigon syntax for defining actuators and sensors.

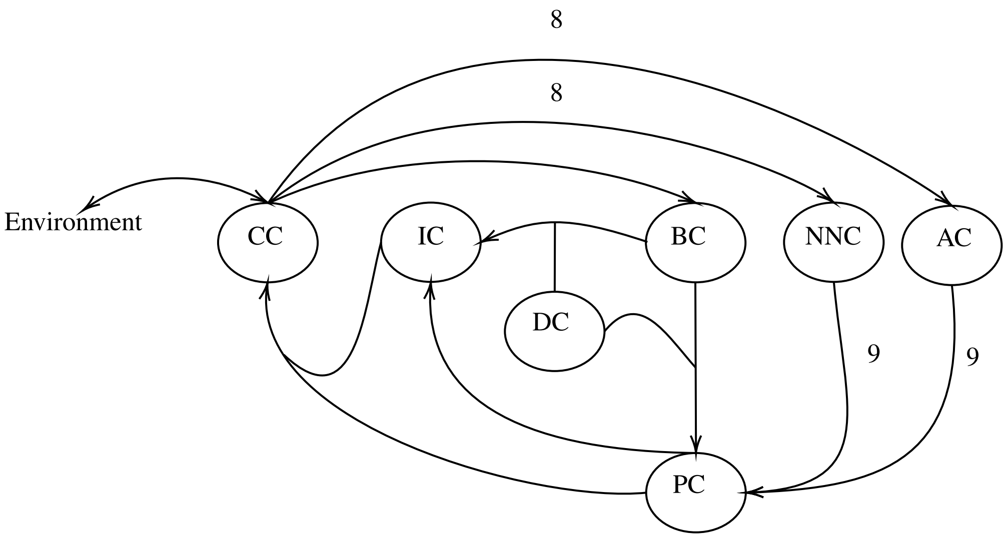

Figure 1: Contexts and bridge-rules of the proposed agent.

4.2 Integrating Neural Network and Auxiliary Contexts into the Agent's Reasoning Cycle

Since the intelligent agent can perceive different data, it is necessary to integrate these perceptions with the custom contexts (i.e. neural network and auxiliary contexts). Bridge-rules presented in 8 are similar to the existing ones that add perceptions into the Beliefs context. However, the main difference is that these new bridge-rules 8 route the perceptions to the responsible custom context. It is worth mentioning that this new bridge-rule provides a generic way of dealing with several data types. For instance, one can define a sensor to perceive audio data that different custom contexts can use.

$$\begin{array}{c} \dfrac{CC : sensor_i(\beta)}{NNC : \beta} \\ \dfrac{CC : sensor_j(\gamma)}{AC : \gamma} \end{array} \tag{8}$$

The final step of this initial integration is achieved by using a neural network's output or the auxiliary context information as a precondition of the agent's plan. The existing version of the Sigon framework does not support verifying whether a certain part of a precondition is satisfied in other contexts. We noticed that this approach did not take advantage of Multi-Context System's main goal to consider different knowledge sources. We changed the Sigon grammar to enable modeling preconditions that can reference different contexts. For each context and term of a precondition, the planning context will execute a bridge-rule to verify whether a precondition is satisfied or not by the referenced contexts. This approach enables us to model the interaction between the planning context with several custom contexts. This bridge-rule is presented in 9.

$$\frac{PC:\ plan(\phi,\beta,Pre,Pos,c_a)\ and\ IC:\ \phi\ and\ C_i : Pre}{CC:\ \beta} \tag{9}$$

where C_i is in the set of existing contexts of the Agent AG, in which for logical contexts Pre is true or can be inferred, or for functional contexts, it exists. In subsection 5 we present a case study that shows how this proposed agent can be implemented in the Sigon framework.

5 CASE STUDY

In this section, we present a mediator agent that is responsible for solving conflicts during negotiation. The mediation process simulates a real-world case in which the mediator is trustworthy that can employ different resources, and provides new information (Trescak et al., 2014). Our main goal is to explore how different types of information (i.e., text and image) can be employed during the agent's decision-making. The mediator agent uses two main strategies during conflict resolution: facial expression recognition and information retrieval. This scenario allows us to explore how different custom contexts interact with other agents' contexts during the reasoning cycle.

Facial expression recognition is a relevant tool for the study of Emotion Recognition Accuracy (ERA). Its usage enables to estimate the impact on objective outcomes in negotiation, a setting that can be highly emotional and in which real-life stakes can be high (Elfenbein et al., 2007). Information retrieval will be employed to expand the agent's knowledge about the negotiation item. During this work, the information added into the agent's knowledge base will be used during the planning phase, more precisely, in situations where facial expressions recognition does not provide an output with the required precision or matches the precondition of an existent plan.

5.1 Negotiation Scenario Definition

In this case study, we modeled a scenario where a person tries to sell an item to another person. At the beginning of the negotiation, the seller proposes an initial price, and the buyer can accept or propose a new value. Our scenario is inspired in the home improvements negotiation scenario from (Parsons et al., 1998) and (Trescak et al., 2014), in which agents must solve conflicts to reach its design goals. We use a mediator agent to provide a fair negotiation, in which the mediator can advise about the price of the item, trying to satisfy both parties. Since the main objective of this case study is to explore the integration of different information types, the negotiation protocol employed in this scenario is simplified. The following subsection shows how we implemented the mediator agent with its two main negotiation strategies.

Subsection 5.2 presents how we modeled a web-scraper and added it to the agent's actuators. We also provide tests regarding similarities functions and several approaches to clean the data that could affect the agent's negotiation strategy. Subsection 5.3 presents the details of the mediator agent implemented in the Sigon framework.

5.2 Web-scraper Implementation

We focused on extracting information from an e-commerce platform called MercadoLivre. This approach enables us to create new perceptions about the information gathered during this process, improving the agent's decision-making by retrieving new information about an item that is being negotiated. The following listing 2 shows an example of the output generated by the web-scraper. After this step, the agent can use text sensors to process these perceptions and generate new information about a specific item.

```

[
 {
  "Title": "Celular 16gb 2gb Ram LG K7i
       Mosquito Away 4g Igual K10 11",
  "Price": 698.58,
  "User": "J.F.IMPORTACAO",
  "Amount": 91,
  "New": false
 }
]
```

Code 2: Perception example.

During this implementation, we faced a few challenges during the information extraction. The first one is that in some cases, the details about an item are presented in different sections in the platform, affecting the quality of the retrieved data. The second one was related to defining strategies to remove entries that did not represent the item. For instance, we tried retrieving information about a specific smartphone, and the platform returns information about this smartphone's accessories, such as charger and screen protection. In this sense, the value of this entry did not represent an accurate value for this item, affecting the agent's new perceptions.

To mitigate these two limitations, we employed the following strategies: remove the fields **amount** and **new** from an entry. The main reason is that, in some cases, the information was not filled on the platform. In the second limitation, we modeled the following strategies in the agent's auxiliary context:

1. Executing similarity functions: for this strategy, we executed the following similarity functions: Hamming, Levenshtein, Jaro, Jaro-Winkler, Smith-Waterman-Gotoh, Sorensen-Dice, Jaccard Overlap Coefficient. To execute these functions, we used a library called **strutil**. This library implementation can be accessed in GitHub repository strutil. We noticed that these functions could not be detected whether a certain entry was not related to the item. Taking that into consideration, we removed this strategy from this implementation;

2. Removing the mild and extreme quartiles: we tried to compute these quartiles, however in some experiments, we noticed that some obvious items were not removed, or no mild and extreme were detected, even though it was clear that some entries did not represent the searched item;

3. Removing outliers based on the standard deviation: we removed the items that did not match the following criteria:
   $price > mean - deviation$ and $price < mean + deviation$. Using this strategy enabled us to remove the entries related to the item accessories, such as screen protectors and chargers. After this step, the agent calculates the mean value of the remaining items, which provides more accurate results about the searched item.

5.3 Designing a Mediator Agent in the Sigon Framework

This subsection presents how to integrate the web-scraper and the trained neural network into an agent developed in Sigon. First, we start by showing how to use the web-scraper as an actuator. Second, we modeled custom context and added the trained neural network into it. Moreover, third, we present the mediator agent modeled as a Multi-Context System. We also provide some reasoning cycles of the mediator during conflict resolution.

In this work we used a framework called Deepface for facial expression recognition. Deepface is a lightweight hybrid high performance face recognition framework, which wraps the most popular face recognition models: VGG-Face (Parkhi et al., 2015), FaceNet (Schroff et al., 2015), OpenFace (Baltrušaitis et al., 2016), DeepFace (Taigman et al., 2014), DeepID (Sun et al., 2014; Sun, 2015) and Dlib (King, 2009) (Serengil and Ozpinar, 2020). Those models already reached and passed the human level accuracy of 97.53% (Serafim et al., 2017; Taigman et al., 2014).

In listening 3, we present an initial version of the mediator agent. This agent has three sensors and two actuators. We define the agent's sensors and actuators as follows:

1. Sensor *textSensor* handles perception about the negotiation item and the information exchange between involved parties;
2. Sensor *negotiationPerception* is responsible for handling perceptions about the information retrieved by the actuator;
3. Sensor *imageSensor* handles images containing pictures of seller or buyer;
4. Actuator *defineNextValue* can inform the advice created in the current reasoning cycle, which consists of increasing, decreasing, or keeping the same value during the negotiation phase;
5. Actuator *findInformation* can retrieve information about the negotiation item, such as price, amount, whether it is a brand new product or not.

```
communication:
    sensor(" textSensor ", " integration .TextSensor").
    sensor(" negotiationPerception ", " integration .WebScraperPerception").
    sensor("imageSensor", " integration .ImageSensor").
    actuator("defineNextValue", " integration .TextActuator").
    actuator(" findInformation ", " integration .WebScraper").

beliefs :
 assistHuman.
 item(LgK10). //smartphone Lgk10
 negotiating .

_neuralNetwork:
 detecEmotion.
 currentEmotion( seller , neutral ).
 currentEmotion(buyer, neutral ).

_auxiliary :
  retrievedPrice (X). //proposes a new value based on the information retrieved

desires :
 updateDecision.

intentions :
 updateDecision.

planner:
    plan(
    updateDecision,
    [action( findInformation () ) ],
    [_neuralNetworks: currentEmotion(buyer, neutral ),
    _neuralNetworks: currentEmotion( seller , neutral ) ], _).
    plan(
    updateDecision,
    [action(defineNextValue(decrease ) ],
    [_neuralNetworks: currentEmotion(buyer, happy),
    neuralNetworks: currentEmotion( seller , sad) ], _).
    plan(
    updateDecision,
    [action(defineNextValue( increase ) ],
    [_neuralNetworks: currentEmotion(buyer, happy) ], _).

! _neuralNetwork X :- communication imageSensor(X).
! _auxiliary X :- communication negotiationPerception (X).
! beliefs X :- communication textSensor(X).
```

Code 3: The initial mental state of the mediator agent implemented in the Sigon framework.

In listing 3, the beliefs context has information about the current state of the negotiation. The desires and intentions contexts define which goal the agent will try to achieve in the current reasoning. It is worth mentioning that an agent can have different desires and intentions. However, for the sake of simplicity, we omit this process. The neural network context has a strategy that can detect emotions based on facial expressions pictures. In this case study, we focused on the phase when the negotiation is stalled, and the mediator agent should provide a new proposal. Taking that into consideration, the mediator agent detects that both parties have neutral emotions. To provide a new proposal, the agent uses its actuator to retrieve information about the item.

Listing 4 presents the next cycle of the mediator's reasoning. In this cycle, the agent will process the information retrieved by its web-scraper actuator and update the knowledge of the auxiliary context. Firstly, the bridge-rule in line 25 will be activated, adding the perception processed by the *negotiationPerception* sensor into the auxiliary context. The auxiliary context uses the strategies presented in subsection 5.2, where the agent removes the outliers and uses the mean value of the retrieved item. Based on the current state of the contexts, the agent then executes the plan of proposing a new deal, presented in line 18.

```
1  communication:
2      sensor(" textSensor ",
          " integration .TextSensor").
3      sensor(" negotiationPerception ",
          " integration .WebScraperPerception").
4      sensor("imageSensor",
          " integration .ImageSensor").
5      actuator("defineNextValue",
          " integration .TextActuator").
6      actuator(" findInformation ",
          " integration .WebScraper").
7
8  _auxiliary :
9   retrievedPrice (659). // agent proposes a new
        value based on the retrieved
10
11 desires :
12  updateDecision.
13
14 intentions :
15  updateDecision.
16
17 planner:
18     plan(
19     updateDecision,
20     [action(defineNextValue( create ) ],
21     [_neuralNetworks: currentEmotion(buyer, sad),
22     _neuralNetworks: currentEmotion( seller , sad),
23      _auxiliary : retrievedPrice (_) ], _).
24
25 ! _neuralNetwork X :- communication
       imageSensor(X).
26 ! _auxiliary X :- communication
        negotiationPerception (X).
27 ! beliefs X :- communication textSensor(X).
```

Code 4: The mental state of the mediator agent during the next reasoning cycle.

In the next cycle, the agent's neural network context detects that both parties are happy with the previous proposed value. The mediator sends a message offering to keep the current value and ends the negotiation. Since the primary goal of this case study is to present how an agent can use a different type of information during decision-making, we decided to omit some steps of negotiation protocol and information about the item.

In this case study, we showed some relevant steps of the mediator's reasoning cycle. We employed information retrieval, how to process different data types, integration with other existing contexts, and how advice can be created and proposed to the involved parties. We also investigated situations in which using only a single resource or reasoning, such as NN for facial expression recognition, is insufficient. Our agent can use a different mechanism, such as information retrieval, to acquire new knowledge about the environment to mitigate this limitation.

6 CONCLUSIONS AND FUTURE WORK

In this work, our primary goal was to explore how to combine connectionist and symbolic methods during an agent's decision-making. To achieve this goal, we employed Multi-Context Systems to model different resources, where its integration with the existent agent's implementation occurs via bridge-rules. Each resource can model a custom context, where a custom context is responsible for processing different types of information. We also provide a flexible way of verifying the plan's precondition with other custom contexts, enabling us to integrate custom contexts with the agent's planning algorithm. One of the main results of our work is that it provides flexible ways of integrating different kinds of resources during the design of intelligent agents.

As a case study, we presented how to build an intelligent agent that can mediate conflict resolution. The developed mediator can use two different strategies: (i)- facial expression recognition and (ii) - retrieve information about a particular item during negotiation. The facial expression recognition was achieved using neural networks, which detects emotions such as happiness, sadness, anger, fear, disgust, surprise, and neutrality. Since we deal with imprecise outputs of the detected emotion, we decided to create an auxiliary custom context. The auxiliary context models a strategy based on information retrieval, where a web-scraper gathers data about the negotiated item, such as price, availability, whether the returned item is brand new or not. The auxiliary custom context also models a strategy for removing data that does not represent the searched item, affecting the agent's decision-making.

We intend to pursue the following paths as future works: (i) - Develop a new version of Sigon framework in Python. This decision enables us to use the most popular Machine Learning libraries without the necessity of using third party libraries to integrate with Sigon agents; (ii) - explore more robust approaches of connectionist and symbolic integration, such as the ones provided in neural-symbolic field; (iii) - deploy this agent in a real-world scenario and compare it with other similar works.

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

REFERENCES

Anjomshoae, S., Najjar, A., Calvaresi, D., and Främling, K. (2019). Explainable agents and robots: Results from a systematic literature review. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1078–1088. International Foundation for Autonomous Agents and Multiagent Systems.

Arrieta, A. B., Díaz-Rodríguez, N., Ser, J. D., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2019). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai.

Baltrušaitis, T., Robinson, P., and Morency, L.-P. (2016). Openface: an open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10. IEEE.

Bennetot, A., Laurent, J.-L., Chatila, R., and Díaz-Rodríguez, N. (2019). Towards explainable neural-symbolic visual reasoning. *arXiv preprint arXiv:1909.09065*.

Besold, T. R., Garcez, A. d., Bader, S., Bowman, H., Domingos, P., Hitzler, P., Kühnberger, K.-U., Lamb, L. C., Lowd, D., Lima, P. M. V., et al. (2017). Neural-symbolic learning and reasoning: A survey and interpretation. *arXiv preprint arXiv:1711.03902*.

Bratman, M. (1987). Intention, plans, and practical reason.

Brewka, G. and Eiter, T. (2007). Equilibria in heterogeneous nonmonotonic multi-context systems. In *AAAI*, volume 7, pages 385–390.

Brewka, G., Eiter, T., and Fink, M. (2011). Nonmonotonic multi-context systems: A flexible approach for integrating heterogeneous knowledge sources. In *Logic programming, knowledge representation, and nonmonotonic reasoning*, pages 233–258. Springer.

Brewka, G., Ellmauthaler, S., and Pührer, J. (2014). Multi-context systems for reactive reasoning in dynamic environments. In *ECAI*, pages 159–164.

Cabalar, P., Costantini, S., De Gasperis, G., and Formisano, A. (2019). Multi-context systems in dynamic environments. *Annals of Mathematics and Artificial Intelligence*, 86(1-3):87–120.

Casali, A., Godo, L., and Sierra, C. (2005). Graded BDI models for agent architectures. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3487 LNAI:126–143.

Chollet, F. et al. (2018). *Deep learning with Python*, volume 361. Manning New York.

Dao-Tran, M. and Eiter, T. (2017). Streaming multi-context systems. In *IJCAI*, volume 2017, pages 1000–1007.

De Freitas, G. S., Gelaim, T. Â., De Mello, R. R. P., and Silveira, R. A. (2019). Perception policies for intelligent virtual agents. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 8(2):87–95.

de Mello, R. R. P., Gelaim, T. Â., and Silveira, R. A. (2018). Negotiating agents: A model based on bdi architecture and multi-context systems using aspiration adaptation theory as a negotiation strategy. In *Conference on Complex, Intelligent, and Software Intensive Systems*, pages 351–362. Springer.

Elfenbein, H. A., Der Foo, M., White, J., Tan, H. H., and Aik, V. C. (2007). Reading your counterpart: The benefit of emotion recognition accuracy for effectiveness in negotiation. *Journal of Nonverbal Behavior*, 31(4):205–223.

Garcez, A. d., Gori, M., Lamb, L. C., Serafini, L., Spranger, M., and Tran, S. N. (2019). Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *arXiv preprint arXiv:1905.06088*.

Garnelo, M., Arulkumaran, K., and Shanahan, M. (2016). Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*.

Garnelo, M. and Shanahan, M. (2019). Reconciling deep learning with symbolic artificial intelligence: representing objects and relations. *Current Opinion in Behavioral Sciences*, 29:17–23.

Gelaim, T. (2021). *Situation awareness and practical reasoning in dynamic environments*. PhD thesis, Federal University of Santa Catarina.

Gelaim, T. Â., Hofer, V. L., Marchi, J., and Silveira, R. A. (2019). Sigon: A multi-context system framework for intelligent agents. *Expert Systems with Applications*, 119:51–60.

Glez-Peña, D., Lourenço, A., López-Fernández, H., Reboiro-Jato, M., and Fdez-Riverola, F. (2014). Web scraping technologies in an api world. *Briefings in bioinformatics*, 15(5):788–797.

Goodman, B. and Flaxman, S. (2017). European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57.

Jedrzejowicz, P. (2011). Machine learning and agents. In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, pages 2–15. Springer.

Kassab, M., Mazzara, M., Lee, J., and Succi, G. (2018). Software architectural patterns in practice: an empirical study. *Innovations in Systems and Software Engineering*, 14(4):263–271.

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758.

Kriesel, D. (2007). A brief introduction on neural networks.

Marra, G., Giannini, F., Diligenti, M., and Gori, M. (2019). Integrating learning and reasoning with deep logic models. *arXiv preprint arXiv:1901.04195*.

McCulloch, W. S. and Pitts, W. (1990). A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biology*, 52(1-2):99–115.

Minsky, M. L. (1991). Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI magazine*, 12(2):34–34.

Ozaki, A. (2020). Learning description logic ontologies: Five approaches. where do they stand? *KI-Künstliche Intelligenz*, 34(3):317–327.

Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition.

Parsons, S., Sierra, C., and Jennings, N. (1998). Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292.

Rao, A. S., Georgeff, M. P., et al. (1995). Bdi agents: From theory to practice. In *ICMAS*, volume 95, pages 312–319.

Rizk, Y., Awad, M., and Tunstel, E. (2018). Decision making in multiagent systems: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):514–529. cited By 10.

Rodrigues, R., Silveira, R. A., and de Santiago, R. (2021). Integrating neural networks into the agent's decision-making: A systematic literature mapping. In *Proceedings for 15th Workshop-School on Agents, Environments, and Applications: 2021*.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Serafim, P., Nogueira, Y., Vidal, C., and Cavalcante-Neto, J. (2017). On the development of an autonomous agent for a 3d first-person shooter game using deep reinforcement learning. In *2017 16th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, pages 155–163. IEEE.

Serengil, S. I. and Ozpinar, A. (2020). Lightface: A hybrid deep face recognition framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 23–27. IEEE.

Sun, Y. (2015). *Deep learning face representation by joint identification-verification*. The Chinese University of Hong Kong (Hong Kong).

Sun, Y., Wang, X., and Tang, X. (2014). Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1891–1898.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708.

Trescak, T., Sierra, C., Simoff, S., and De Mantaras, R. L. (2014). Dispute resolution using argumentation-based mediation. *arXiv preprint arXiv:1409.4164*.

Valiant, L. G. (2003). Three problems in computer science. *J. ACM*, 50(1):96–99.

Wang, S.-C. (2003). *Artificial Neural Network*, pages 81–100. Springer US, Boston, MA.

Wooldridge, M., Jennings, N. R., et al. (1995). Intelligent agents: Theory and practice. *Knowledge engineering review*, 10(2):115–152.