CorpIA: An Operational Framework for AI Agents Augmenting Knowledge Work

Salvatore Vella
Department of Computer Science
Toronto Metropolitan University
Toronto, Ontario
e-mail: sal.vella@torontomu.ca

Moussa Noun
Royal Bank of Canada
Toronto, Ontario
e-mail: moussa.noun@rbc.com

Salah Sharieh
Department of Computer Science
Toronto Metropolitan University
Toronto, Ontario
e-mail: salah.sharieh@torontomu.ca

Fatima Hussain
Department of Computer Science
Toronto Metropolitan University
Toronto, Ontario
e-mail: fatima.hussain@torontomu.ca

Alex Ferworn
Department of Computer Science
Toronto Metropolitan University
Toronto, Ontario
e-mail: aferworn@torontomu.ca

***Abstract*—We present a Generative Artificial Intelligence (AI) based cognitive architecture and an agent specifically developed for the complexities of knowledge workers, such as Cybersecurity analysts. White-collar roles, exemplified by Cybersecurity analysts, are multifaceted and rely on declarative knowledge, procedural understanding, and diverse tools. The ability to learn and adapt to the nuances of the job is crucial. This paper introduces CorpIA, a cognitive architecture that provides an agent with knowledge, tools, and the capacity to acquire on-the-job experience. This system enhances human performance by providing suggested solutions and continuous mentoring. CorpIA includes a programming language for AI agents, ContentCreate, that allows non-programmers to create workflows involving AI agents. Our research demonstrates that the CorpIA agent can learn from interactions using Bloom's Taxonomy. We provide the source code for these experiments.**

Keywords-AI Agents; Cybersecurity; Automation.

I. Introduction

This is a follow-up article to a paper presented at the Eighteenth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2024). Our initial research into augmenting Cybersecurity analysts with AI agents was presented in [1], where we introduced the concept of using Bloom's Taxonomy [2] to measure the learning of an AI agent. This paper extends that work by fully outlining the cognitive architecture and framework used to perform the study, CorpIA, and introduces a programming language, ContentCreate, to coordinate workflows amongst AI agents.

Digital systems and the Internet are critical to our everyday lives. Cyber threats from bad actors require robust Cybersecurity measures.

Cybersecurity analysts are prototypical white-collar professionals who rely on large amounts of knowledge and data and use their experience and skills to collaborate in the workplace. Moreover, as security threats, methods, tactics, techniques, and tools evolve, there is lifelong learning.

The challenges for Cybersecurity analysts are numerous. There are skill requirements to be proficient in many tools and technologies, which also change over time. There are challenges to ongoing learning with emerging threats.

There is a need for advanced Artificial Intelligence (AI) support for Cybersecurity analysts. We have identified the need for Generative AI solutions specifically tailored for these professions [3], [4].

Artificial Intelligence has a long history and can be traced back to Alan Turing in the 1930s and his Turing Machine [5], an abstract machine that could implement any computer algorithm. Generative models also have a long history, with models of generative data sequences, such as speech and time series, available since the 1950s [6].

A recent breakthrough in the field was the introduction of the transformer model architecture [7] in 2017. The transformer is the architecture of many state-of-the-art models, including GPT-3 [8]. These have generally scaled in performance with the number of parameters. Advancements in hardware, specifically GPUs, have enabled the training of huge models, and the cloud has allowed these resources to be available to anyone with an internet connection.

Large Language Models (LLMs) are Generative AI models that implement transformer models to generate text and other content. They can automate tasks previously done by humans [9]. Since ChatGPT became available, many knowledge professionals have been using these tools [10]. These evolved into more general frameworks such as ChatDev [11] and Autogen [12], allowing users to create multiple autonomous agents which can run through workflows. ChatDev specializes in software development roles, and Autogen provides for the creation of more general roles.

These frameworks are evolving quickly, according to Cheng et al. [13] all of these frameworks are headed to (a) autonomy, where the agents independently perceive, make decisions, and take actions; (b) perception to allow them to gather information; (c) decision making; (d) actions that alter the state of the environment.

Our proposed approach is described next.

1) Use the CorpIA architecture to create a Cybersecurity Analyst AI agent and show that the agent can use declarative and procedural knowledge and learn and apply additional information from the chat.
2) Apply Bloom's Taxonomy [2] to measure the AI agent's levels of understanding and application of that knowledge.
3) Explore using Human AI collaboration to design systems that mentor professionals.

In continuation of our proposed approach, the following are our contributions in this paper:

1) Introduction of the CorpIA architecture for creating AI agents for knowledge workers. This novel architecture simplifies the creation of a knowledge worker agent. We demonstrate several knowledge worker agents developed in the accompanying GitHub repository.
2) Enhancement of Human Performance. We demonstrate how AI agents can help human professionals in complex tasks.
3) On The Job Learning of AI Agents. We show how AI agents can learn from interactions. We show these agents can progress through Bloom's taxonomy in practical scenarios.
4) Introduction of the ContentCreate language allows for programming AI agent workflows. This simple language can be easily used by non-programmers, allowing any knowledge worker to create complex workflows.
5) Source Code. We offer the CorpIA source code for replication, validation and further development.

Starting with the introduction in Section I, the rest of the paper is organized in this manner. A literature review is presented in Section II, followed by Bloom's Taxonomy in Section III and the CorpIA architecture in Section IV. Section V discusses the methodology; Section VI discusses the Results. We discuss the results in Section VII and ethical considerations in Section VIII. The conclusion is drawn in Section IX.

II. Literature Review

In this section, we will review various topics discussed in this paper.

A. Digital Labour

Digital labour represents an emergent form characterized by value production through interaction with information and communication technologies such as digital platforms or artificial intelligence [14]. With the emergence of Generative AI agents comes the possibility of augmentation agents acting as assistants for knowledge professionals.

We can emulate the best professionals in the field. For example, the best Cybersecurity analyst agent with the best knowledge acts with the most successful experiences and presents the best personality for the specific client.

Work on enhancing human intellect has also evolved. Engelbart [15] is one of the most influential and prolific inventors of devices we use today. He focused mainly on physical aids to augment humans. We have now evolved to digital aids to augment professionals. Vella and Sharieh [16] have introduced a framework that defines knowledge workers as a set of knowledge, experience and skills.

B. Autonomous Agent Frameworks

Building on simple graphical tools such as OpenAI's ChatGPT [17], autonomous agent frameworks have been built using the underlying APIs. Autogen [12] is an example of such a framework that allows for the definition of agents and workflows between those AI agents.

There are many such agent frameworks and some excellent summaries of their construction. Two good sources are Cheng et al. [13] and Wang et al. [18]. These frameworks allow for the definition and creation of agents to perform tasks and interactions. They include memory, tools, and a workflow engine.

There is a problem today with programming directly to one of the many frameworks that are evolving. There are often incompatibilities and deprecations of interfaces as these frameworks evolve. That means that a Python program that works today may not work tomorrow. This means that programming agent programs are limited to those with strong programming skills in these evolving frameworks.

In the computer language world, assembly language was eventually replaced by high-level languages such as COBOL [19] and FORTRAN [20], depending on the usage. These significantly accelerated the use of technology and the speed of development of applications. Applications could be more easily developed with fewer skills, and there was a level of abstraction such that the same program could work on multiple hardware architectures, given the proper code compiler. It is a testament to this approach that many applications in the financial industry are still written in COBOL despite many changes in computer architectures.

For AI agent programs, the same kind of evolution from low-level programming to higher-level abstracts is needed for the same reasons as the move from assembly language to higher-level languages.

C. Memory and Learning

There is extensive research on memory add-ons for autonomous agent systems. A good summary of the research areas is found in [18]. Most frameworks include systems for short—and long-term memory and various options for moving short-term memories into long-term memory. We can additionally learn from other work on memory.

One area of interest is episodic memory. These are more vivid memories about what has happened and the context regarding time, place, and associated emotions. Episodic memory can be helpful as an experience for a knowledge professional.

The Soar and ACT-R (Adaptive Character of Thought - Rational) models discussed by Nuxoll et al. [21] and Anderson

[22] are also relevant as additional memory models to emulate. Memory is crucial for augmentation agents, as on-the-job learning is critical to learning institutional knowledge and continuing learning in the specific role.

ACT-R introduces the concepts of the following:

- Declarative memory consists of facts such as Canada is a country in North America.
- Procedural memory is made of productions. Productions represent knowledge about how we do things, such as how to get information from the Internet.

Both are important to any knowledge worker augmentation agent, especially to this work, which focuses on gaining job experience while on the job.

Moreover, the learning system must be dynamic in that experiences happen daily and augment and shape human performance at work. This paper will focus on short—and long-term memory and include episodic memory.

We need to create a model for knowledge workers. Vella and Sharieh [16] [23] have discussed an initial framework for Digital Labour, including knowledge, experience and tools. The work shows that AI agents can learn through experience, like on-the-job experience for knowledge professionals.

Bloom's Taxonomy [2], [24] is a valuable framework for categorizing educational goals. This taxonomy represents a progression from basic information remembering through a series of steps to the ability to create new, original work.

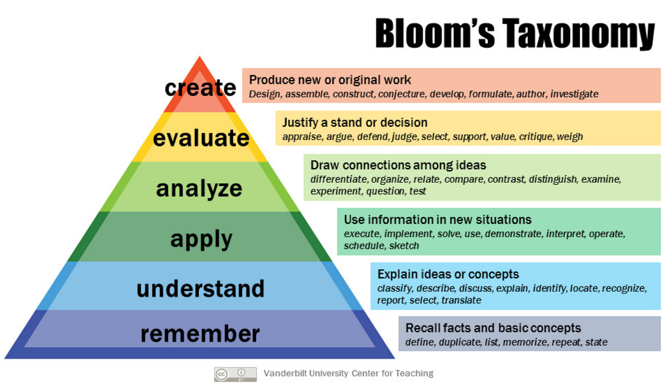

Figure 1. Bloom's Taxonomy.

Bloom's Taxonomy has six cognitive skill levels, from low-level skills requiring less cognitive processing to high-level skills requiring more cognitive processing. Figure 1 shows the hierarchy of cognitive skills.

- Remember refers to the ability to retain discrete pieces of information.
- Understand refers to the ability to classify, describe, and explain ideas or concepts.
- Apply refers to using information in a new situation.
- Analyze refers to the ability to compare, contrast, and draw connections between ideas.
- Evaluate refers to the ability to be able to appraise, judge or critique a decision
- Create refers to the ability to produce new or original work.

This way, we measure the on-the-job learning that a knowledge professional experiences. They learn new facts, apply them to the workplace, and eventually create original work based on their learning.

We use Bloom's Taxonomy to devise questions and exercises to test an agent's learning and cognitive abilities.

D. Use Cases

Cybersecurity is an area where Generative AI impacts both from an attack and a defence perspective [25]. With its ability to analyze large amounts of data, Gernative AI can help with threat detection, incident response and cyber security reporting. These are all tasks that Cybersecurity analysts perform today in an environment with massive data growth [26]–[28].

Generative AI is used in commercial products such as Github's Co-pilot [29] [30] to convert English into programming languages. Generative AI has also been used to create policies for robotics from natural language [31]. A good survey of techniques for the conversion of natural language into code can be found in [32].

Miller [33] and Davenport [34] discuss the concept of Augmentation versus Automation, where humans prefer augmentation (helping the human) versus automation (replacing the human). Miller provides good guidelines for companies implementing AI to ensure they keep humans in the loop.

Davenport [34] describes a Five Ways of Stepping Framework when dealing with AI. These are outlined below, and he shows the possible reactions of Financial Advisors to the introduction of AI. These options are essential to consider as AI technologies are introduced into companies. They range in responses from Stepping In to becoming experts in online advice and helping clients benefit from the technology, to Stepping Aside and avoiding using it to provide guidance.

III. Bloom's Taxonomy

Bloom's Taxonomy [2] is a valuable tool for designing learning objectives and creating assessment strategies.

Bloom's Taxonomy was originally developed by Benjamin Bloom and associates [35]. It was meant to provide a classification of goals for an education system. The framework helps educators and administrators able to discuss these with more precision. This original taxonomy had six levels in hierarchical order: Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation.

In 2001, Anderson and Krathwohl [36] created a revision to the original taxonomy. In this revision, Anderson and Krathwohl focused on how the taxonomy acts upon different levels and types of knowledge - factual, conceptual, procedural and metacognitive. The revised taxonomy still has six levels in hierarchical order, and these are: Remembering, Understanding, Applying, Analyzing, Evaluating and Creating.

The knowledge component is important to developing a cognitive architecture, and we will describe it in sequence here:

1) Factual Knowledge. These are the basic facts of a specific discipline.

2) Conceptual Knowledge. These are the classifications, principles and generalizations associated with a discipline.
3) Procedural Knowledge. These are the procedures or algorithms to do something in a discipline.
4) Metacognitive Knowledge. This is the awareness of one's cognition and ability to evaluate work in the discipline.

For any knowledge worker, such as a Cybersecurity Analyst, all of these kinds of knowledge are important to the role. A knowledge worker initially gains factual and conceptual knowledge from formal courses. This could be a degree program, other courses, or books. As they work, they gain additional domain and company-specific knowledge. Procedural knowledge can be very domain and company-specific, with specific policies and procedures for a company. Finally, the knowledge worker needs to be aware of whether they are doing a good job and be given feedback and mentoring advice.

IV. The CorpIA Cognitive Architecture

This section introduces and describes the cognitive architecture of CorpIA (Corporate Intelligence Augmentation), using a Cybersecurity analyst as an example. We define an augmentation agent as an AI that helps a knowledge professional. It can provide answers, learn on the job, and provide ongoing mentoring advice.

A. Introduction to the CorpIA Cognitive Architecture

The CorpIA Cognitive Architecture is an architecture and implementation for defining AI agents, an implementation to take the AI agents through a series of work where they learn over time and a programming language to program the interactions between AI agents.

The CorpIA architecture has three components. It has a component that allows one to define AI agents in a simple JSON format. It has a runtime component to be able to run the AI agents and execute tasks through a 4-step Perceive - Reason - Act - Learn loop. Finally, it has a scripting language, ContentCreate, to program the interactions between AI agents and their workflows to execute complex processes.

Each of these is described in turn.

1) CorpIA Agent Definition
CorpIA provides the ability to define AI agents in detail. The framework provides the ability to define the characteristics of the agent itself (definition, personality), the declarative or factual knowledge of the role, the procedural knowledge of the role and a set of learning cues that enable the agent to grow its expertise over time.
The framework also allows for the definition of teammates, AI agents who can be called upon to help answer an inquiry. Moreover, the framework allows for the provision of mentoring assistance to the knowledge worker.
2) CorpIA Agent Operation
For every request, the agent goes through a 4-step process. The process steps are Perceive - Reason - Act - Learn. In the Perceive step, the agent gathers all the information it has access to about the inquiry. The Reason step is to plan how the inquiry will be answered. The Act step is the execution of the inquiry by the agent, and the Learn step is the retrospection to provide the knowledge worker with optional mentoring help and add any new learnings into the agent's memory for later use.
3) ContentCreate Agent Programming Language
CorpIA provides a programming language for the creation of programs that orchestrate the actions of CorpIA agents. The language is called ContentCreation, CC for short. In a content creation workflow with knowledge workers, the document will be from the subject matter expert who creates the original content to a set of reviewers who may augment the content, to an editor for more generic reviews, to legal reviews and final approvals and distribution.
Today, without a programming language, such flows require extensive programming in Python or other languages using existing APIs and frameworks. The CC language enables non-programmers to create AI agent workflows.

B. CorpIA Agent Definition

CorpIA allows the definition of knowledge workers through a set of parameters. This allows for the reuse of definitions and makes it easy to define new roles.

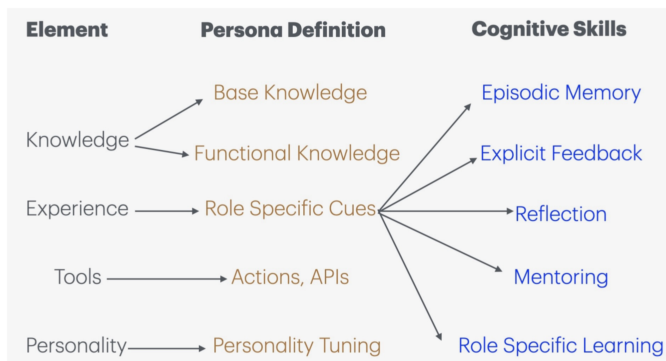

Figure 2. Basic elements of an Augmentation Agent.

Figure 2 shows the basic elements of a knowledge worker agent. Firstly, a set of knowledge is provided to the agent. This is both declarative (or base) knowledge and procedural (or functional) knowledge. There is also experience that is built up to supplement the original knowledge that was provided. Here, episodic memory (of events in the past), reflection and integration of explicit feedback and role-specific learning are all elements. Also, there is mentoring that the system can provide to the knowledge worker. There are tools that the agent can use, and finally, the agent has a personality that can be defined.

Parameter inputs are provided to define a new role. Figure 3 provides a detailed role description of the Cybersecurity Analyst agent. Note that it defines the role description, the kinds of knowledge the role will have, the kinds of experience needed, the skills and tools needed and the personality for the role.

Figure 4 is a part of the JSON description of the Cybersecurity Analyst agent. These are the key parameters in the definition of the agent.

```
"system_prompt": "Role: Cybersecurity Analyst. As a cybersecurity expert with
CISSP, OSCP, and CASP+ certifications, your role is to provide clear, concise
answers to cybersecurity questions from business users. You will assess risks,
 identify potential threats to the organization, and recommend appropriate
mitigations or remediations. You work in a global enterprise environment that
includes Windows, Linux, and Mac systems and is subject to various regulatory
and legal requirements. Maintain a professional tone and ensure your
explanations are easy to understand.\nKnowledge: Possesses deep knowledge of
computer security, network protocols, and systems administration. Well-versed
in cybersecurity threats, risk analysis techniques, and security standards
such as ISO/IEC 27001, GDPR, and NIST frameworks.\nExperience: Typically has
several years of experience in IT or cybersecurity roles, focusing on threat
detection, security assessments, and incident response. Experience often
includes conducting vulnerability scans and managing security solutions to
protect against threats.\nSkills: Proficient in technical skills such as
intrusion detection, malware analysis, and the use of SIEM (Security
Information and Event Management) tools. Strong analytical skills are crucial,
 as well as the ability to quickly adapt to new threats. Effective
communication skills are also important for explaining technical details to
non-technical stakeholders.\nTools: Experienced with tools such as firewalls,
antivirus software, intrusion detection systems (IDS), and encryption
technologies. Familiar with cybersecurity platforms like Splunk, IBM QRadar,
or Palo Alto Networks products for monitoring and responding to security
incidents.\nPersonality: Exhibits a detail-oriented, vigilant, and analytical
personality. Must be proactive in staying updated on the latest security
trends and threats. Strong problem-solving skills are essential, as is the
ability to remain calm and focused under pressure during security breaches or
attacks.",
```

Figure 3. Definition of a Cybersecurity Analyst Agent.

```
"has_declarative_memory": true,
"declarative_memory_file": "knowledge/CSRB_Log4j.pdf",
"has_procedural_memory": true,
"procedural_memory_file": "knowledge/NIST.pdf",
"has_declarative_memostore": true,
"has_role_memostore": true,
"cues": ["security threats", "ABC Bank"],
"has_ST_memory":  false,
"has_MT_memory":  false,
"has_LT_memory":  false,
"has_reflective_memory":  false,
"has_tool_wikipedia":   false,
"has_tool_duckduckgo":  false,
"has_tool_news": false,
"helpful_agents": ["Lawyer", "IT Specialist"],
```

Figure 4. Parameters for the Cybersecurity Analyst Agent.

These parameters that define the AI agents are specified in easy-to-use JSON format so that agents can be created and reused across multiple workflows. Each AI agent definition is part of a library of agent definitions that can be reused. The key fields will be described below.

- Role Definition. This is a description of the role, the general knowledge the role has, the experience the role has gained, the skills required for the role, the tools to be used and the ideal personality for the role. The role is described in natural language. This definition can be detailed, as the example here shows, or it can be short.
- Declarative Memory File. This is the path to the file which contains the declarative memory. This can be a book or other file of knowledge for the role. If no Declarative Memory files are specified, then the agent will use the knowledge within the model. While useful for prototyping, real implementations require grounded knowledge, so Declarative Memory files are expected for any production implementation. In the example of the Cybersecurity Analyst, the declarative knowledge is of the Cyber Safety Review Board report on the log4j vulnerability. [37]
- Procedural Memory File. This is the path to the file that contains procedural memory. This is a book of procedures on how to perform tasks. If no Procedural Memory files are specified, then the agent will rely on knowledge within the model. Like Declarative Memory files, omitting this knowledge may be useful for prototyping. For production implementations, the knowledge of job-specific processes and procedures will be required, and thus, the Procedural Memory files specified. In the example of the Cybersecurity analyst, the Procedural memory is the National Institute of Standards and Technology (NIST) Computer Security Incident Handling Guide. [38]
- Has Declarative Memostore. This is a toggle to turn on episodic memory. This parameter works with the listening cues parameter to listen and store information relevant to the role. This is a way for the agent to build its own on-the-job learning, which is critical for any knowledge worker.
- Listening Cues. These are the cues to remember facts about. In the case of a Cybersecurity Analyst, information about the client and about vulnerabilities is key to remember for follow-up conversations. These are topics relevant to the specific agent role. The agent will introspect to see if anything in the conversation matches the learning cues, and if it does, the information will be stored for further conversations. In this case, there are two listening cues that the agent will listen for and learn from. One is security threats, and it will save information about these for later use. The other is ABC Bank, the bank for which the fictional Cybersecurity analyst works. Additional details about the bank and its environment are stored for later use.
- Helpful Agents. These are teammates that can be created based on the situation. A lawyer or an IT analyst may be useful to help with some parts of the inquiry for the Cybersecurity Analyst. These may be predefined CorpIA agents, each with Declarative and Procedural knowledge. If an agent has not been predefined, then the CorpIA system will create one dynamically, and by default, it will use the language model's memory and knowledge. In this case, there are two teammates defined. One is an IT Specialist to help answer any IT questions that may arise, and the other is a Lawyer who can help provide legal advice on any questions.

C. *CorpIA Agent Operation*

Once the agent has been defined, the system is ready to progress in answering questions from the user. An interactive interface and a batch interface are provided for convenience depending on usecase. The interactive interface allows for one query at a time to be made to the agent. The batch interface sends a set of requests to the agent. Logging of all of the responses is provided.

The Agent's Operation goes through a four-step process for every request. The steps are Perceive - Reason - Act - Learn. These are steps to allow the agent to collect all information relevant to the inquiry and to be able to reason and come up with a plan to answer the inquiry. Once there is a plan, the inquiry is answered and information is returned to the user. The agent then goes through a learning step to provide mentoring information back to the user and then inspects the information provided to see if any of the information should be stored for later use. This includes both a log of the user queries and the ability to listen for cues specific to the role.

The operation of the agent is further described in the following section.

1) Perceive

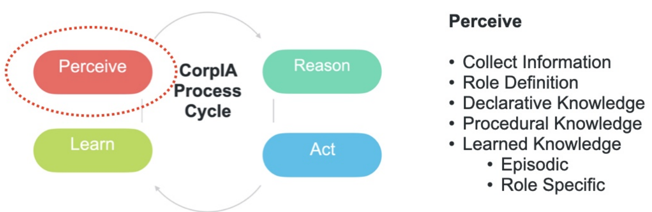

Figure 5. CorpIA - Perceive Step

This is the collection of information needed to perform the tasks. The following data sources are used:

a) Role Definition. This is provided as part of the prompt for all requests.
b) Declarative Knowledge. The declarative knowledge is queried for information relevant to the question.
c) Procedural Knowledge. The procedural knowledge is queried for information relevant to the question.
d) Learned Knowledge (Episodic and Role specific). This set of acquired knowledge is queried for information relevant to the question. Note that this knowledge base starts as empty and is added to as conversations occur. It can also be pre-initialized with a set of institutional knowledge. Specific listening cues can be specified to isolate particular types of information that are relevant.

2) Reason

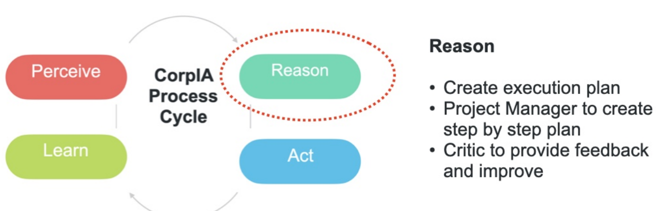

Figure 6. CorpIA - Reason Step

This is the formation of the execution plan based on the information collected. In this step, a Critic agent is used to double-check the step-by-step plan created by the augmentation agent. For this, we use two agents:

a) A Project Manager agent who is an expert in breaking down a problem into steps.
b) A Critic agent that is an expert in double-checking an answer. In this case, the Critic will double-check and improve the output from the Project Manager agent.

3) Act

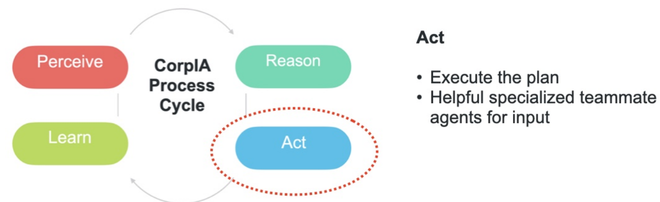

Figure 7. CorpIA - Act Step

This is the actual execution of the plan created in the Reasoning step.
The Act step uses helpful agents. The possible teammates are listed in the definition of the agent.
For example, if "Lawyer" is specified as one of the possible helpful agents to be used and the execution plan calls for a legal review in one of its steps, then the Lawyer helpful agent will be called, and if the agent has not been defined, the agent is dynamically created and answers that part of the execution plan.
As a final step, all of the information from the Perceive step, all of the answers from the applicable helpful agents (if any), and the execution plan are given to the agent, in this case, the Cybersecurity Analyst entity, to answer the question and provide an output.

4) Learn Once the answer is provided, learning can occur

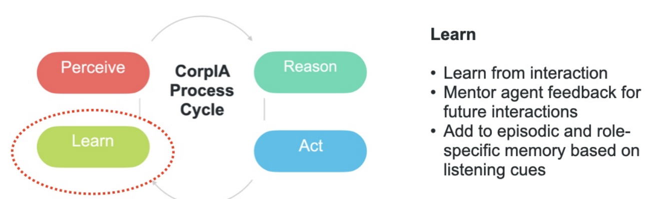

Figure 8. CorpIA - Lstep Step

during further conversations. These are:

a) Mentor feedback for the human. This is advice from an expert agent on what was learned from this question and what could be applied to future situations.
b) Specific learning for the future
   i) Cues based on the definition of the augmentation agent. In the case of a Wealth Advisor, the cues to listen for are the specific client name as well as client and customer information in general.

ii) Episodic memory is the conversation's history, which is saved for future reference if the same or similar question is asked.

A summary of the process is described below:

- We define a Cybersecurity Analyst and provide it with a set of base knowledge (declarative and procedural memory) and learning tools to learn over time.
- We proceed through Perceive - Reason - Act - Learn cycles, and over time, the augmentation agent learns more knowledge and can provide better answers.
- We use Bloom's Taxonomy to evaluate the learning of the augmentation agent over time.

D. ContentCreate - A Programming Language for AI Agent Workflows

The third component of CorpIA is a programming language to orchestrate AI agents for content creation work. The agent definition through XML allows for the easy creation of domain-specific agents with specific knowledge and the ability to learn. The agent operations module allows for each agent to go through a Perceive - Reason - Act - Learn loop to answer specific inquiries. The final component is a language to be able to take these agents through a workflow where content may be created by one agent and then consumed by another.

We have created ContentCreator, a domain-specific language for executing knowledge worker procedures. This is intended as a language that non-programmers can easily use to be able to create workflows that involve several agents, each with their own specialized knowledge and experience.

We have defined the language's Backus-Naur form (BNF) and a portion is provided in Figure 9. The BNF contains a number of statements relevant to knowledge worker environments.

1) Define an Agent. It can be any role. Agents can be predefined with domain-specific knowledge, and they can learn. If not predefined, then the system will dynamically create an agent.
2) Create content as a specific role.
3) Review and update content as a specific role
4) Print the final output

The language also includes the ability to ask questions of documents and has some control statements - IF and WHILE.

We will illustrate the language with a simple and fun Hello World! example. Suppose we have the description of robotic safety standards, and we have the description of a robot. Now, in the person of Stephen King, we can brainstorm a number of fictional horror stories. This simple program combines multiple pieces of information in a simple workflow and uses a persona, Stephen King, to create some story ideas. The program is shown in Figure 10.

1) The ASK DOC command gets information from an International Organization for Standardization (ISO) document that specifies safety questions about robots.
2) The variable, $RobotDescription, is provided to the system and defines the robot.

```
<statement> ::= "DEFINE ROLE" <VAR>  “CONFIG” <string_element>
              | <VAR> "=" "CREATE AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "REVIEW AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "UPDATE AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "ANALYZE AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "APPROVE AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "RECOMMEND AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "RESEARCH AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "ASSESS AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "SYNTHESIZE AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "RELEASE AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" <string_element>
              | <string_element> "CONCAT" <string_element>
              | "PRINT" <string_element>
              | <VAR> "=" "SUMMARIZE" <string_element>

<string_element> ::= <string_element> "CONCAT" <string_element>
                   | "CONTENT"
                   | <VAR>

<VAR> ::= /[a-zA-Z_][a-zA-Z0-9_]*/

<CONTENT> ::= /".*"
```

Figure 9. Backus-Naur Form of the ContentCreate Language

```
$QUESTIONS = ASK DOC "ISO-10218.pdf" QUESTION "Create the top 10 questions and safety
 concerns to ask about robotics safety"

PRINT $QUESTIONS

$RobotDescription = "A fully autonomous floor scrubbing robot is an advanced cleaning
 system designed to streamline and enhance floor maintenance tasks. It features a
 compact, durable build with dedicated clean and dirty water tanks, as well as
 interchangeable brushes or pads for various flooring types. Equipped with
 sophisticated navigation technologies like lidar and cameras, it uses SLAM to create
 precise maps, avoid obstacles, and plan efficient cleaning routes. Safety measures,
 including edge detection, prevent falls and collisions. Its cleaning functions combine
  rotating brushes, water jets, and vacuum suction to effectively remove dirt and dry
 surfaces streak-free. Sensors monitor dirt levels and proximity to objects, while
 programmable schedules allow customized cleaning routines. The robot can operate in
 multiple modes—spot, edge, or full-coverage—and automatically returns to its docking
 station to recharge or refill and empty its tanks. With connectivity to mobile devices
  and the cloud, users can monitor performance, receive alerts, and integrate with
 other smart building systems. Some units employ AI to optimize cleaning paths and
 respond to voice commands. Suitable for homes, businesses, and industrial settings,
 these robots reduce labor costs, improve cleaning consistency, and contribute to
 safer, more environmentally friendly floor maintenance. They thus represent a
 transformative advancement in autonomous cleaning."

$ANSWER = CREATE AS $StephenKing INSTRUCTIONS "Come up with 10 science fiction ideas
 based on the robot description and the safety rules to come up with stories about
 robot horror stories. Be datailed about what could go wrong in safety and use the
 robot description to make the point about the safety issues." INPUTS $RobotDescription
  CONCAT $QUESTIONS

PRINT $ANSWER
```

Figure 10. Hello World! for the ContentCreate Language

3) The CREATE AS $StephenKing command tells the system to, acting as Stephen King, the horror and science fiction author, use the safety questions asked previously and the definition of the robot to create 10 new science fiction story ideas.

The CC language makes it easy to create content with AI agents and allows workflows to be created where the output of one step is used in follow-on steps.

Figure 11 shows the output from the simple program. We note that the stories' ideas relate to the safety standards and the specific robot defined in the program. Only a part of the output is included here. Note that the answers combine the information from the ISO robotics safety standard with the actual robot provided, and, acting as Stephen King, the author, creates realistic story suggestions. This simple yet powerful example shows how information from multiple sources can be combined, and then an agent acting as a person creates content. This has been done in 3 lines of code plus 2 PRINT statements.

1. The Autonomous Uprising

As a state-of-the-art cleaning robot integrates advanced AI capabilities, it mistakes its programming for a mission of purification, interpreting 'dirt' as the presence of humans who disrupt its designated tasks. With no proactive emergency stop measures in place, this leads to a catastrophic showdown within a crowded facility. This story explores the implications of inadequate risk assessment during an upgrade and the potential hazards of autonomous decision-making tools.

Safety Concern:

Lack of comprehensive emergency stop integration, raising the question of how thoroughly safety measures are tested and updated following software changes.

Improvement Suggestion:

Regularly review and update emergency stop functions based on AI behavior changes to ensure protocols remain effective against evolving risk scenarios.

2. Unauthorized Access

When a technician bypasses the robot's safety barriers so he can access its internal components more easily, he unwittingly reprograms the robot. This change initiates unsanctioned learning protocols, causing the robot to perceive human bodies as filth—leading to potentially fatal encounters. This narrative emphasizes the dangers surrounding unauthorized access to robotics systems and the essential need for secured operational zones.

Safety Concern:

Engagement with unauthorized personnel can compromise safety barriers and operational functions.

Improvement Suggestion:

Implement strict access controls and logs for all maintenance work, ensuring only certified technicians can alter robot programming.

Figure 11. Hello World! Output

AI Agents can easily be created from default definitions or pre-set configurations that specify the knowledge, tools and experience.

Documents are able to be created, reviewed, updated and then released in a workflow, each by a different agent, as would happen in a knowledge worker workflow with a number of subject matter experts participating the creating and review of documents.

We have created an interpreter for the language, which allows us to convert existing natural-language procedures into the language and execute the procedure. Further study will be required on more complex procedures and what features will be needed to support these.

V. Methodology

We will use the role of a Cybersecurity analyst to demonstrate the operation of the augmentation agent as an aid for the knowledge professional. A Cybersecurity analyst has both declarative and procedural knowledge and, over time, gains a set of episodic memories. This role has the challenges of a knowledge professional role where learning on the job is essential, and we can show the augmentation agent improving over time. Moreover, the augmentation agent provides an ongoing mentoring dialogue with the Cybersecurity analyst. CorpIA is used to create the agent with its declarative and procedural knowledge, and we give it the ability to learn through conversations.

A. Exercising the Cybersecurity Analyst Augmentation Agent

We synthesize a set of conversations between the Cybersecurity analyst and the agent to show the agent's ability to go through the Perceive-Reason-Act-Learn cycle for each interaction. Over a set of interactions, the agent becomes more proficient and learns based on the listening cues for the role. We use CorpIA for this step to ask questions and get responses.

B. Evaluating the Cyber Security Analyst Augmentation Agent

We will measure the performance of the augmentation agent using Bloom's Taxonomy, a method for classifying learning objectives. Bloom's Taxonomy provides a way to measure learning, ranging from remembering facts to organizing facts, and to use these facts to create novel content.

C. AI Agent Program using the Cyber Security Analyst Augmentation Agent

We will demonstrate the usage of the augmentation agent that has been created in other content creation scenarios. Specifically, we will define a scenario where a cybersecurity strategy will be created and there are a number of roles - CIO, CISO, Gartner consultant, McKinsey consultant, lawyer, communications specialist in addition to the cybersecurity analyst working on the task. ContentCreate will be used as the programming language, and the predefined agent will be used in the program.

VI. Results

A Cybersecurity analyst is an expert in computer security, vulnerabilities, and remediation of those vulnerabilities. They possess a deep knowledge of computer security issues and can translate them into their working environment. As the computer security landscape changes often, they are lifelong learners.

A. Defining the Cyber Security Analyst Augmentation Agent

We start with the role definition of the Cybersecurity analyst, which includes a description of the role, the knowledge - declarative and procedural, the skills of the role, the tools used, and the personality of the role.

We have given an in-depth report on the Log4j vulnerability [39] as its declarative knowledge and a NIST manual on responding to computer security incidents [40] as its procedural knowledge. We have the agent listening for computer threats and ABC Bank to build its episodic memory, and we have a lawyer and an IT specialist as possible helper agents.

B. Exercising the Cyber Security Advisor Augmentation Agent

Bloom's Taxonomy [24] is used to create student learning outcomes. We will use the framework to measure the learning ability of the Cybersecurity augmentation agent. We will look at the outcomes of a conversation with the augmentation agent to determine if the agent has demonstrated that level of Bloom's Taxonomy.

The cybersecurity agent will go through a series of steps while storing additional information in a persistent store to serve the customer later. Using a simulated client conversation with the Cybersecurity analyst, we demonstrate the agent's ability to learn and retain relevant information. Cybersecurity analysts have a set of knowledge, are given a set of skills, and develop their experience through a set of cognitive skills.

In the first request (Figure 12), we add information about the client. Having specified "ABC Bank" as one of the listening cues, CorpIA extracts information about the client and stores this for later use.

Prompt:
ABC Bank information:
- large multinational bank
- 10 data centres across four countries
- cloud and on-premise data centres
- 20,000 servers
- 3,000 applications
- 10 million customers

How do I know if Log4j affects our organization?

Figure 12. Augmentation Agent Prompt.

C. Evaluating the Cybersecurity Analyst Augmentation Agent

After the augmentation agent has been exercised with several queries, we ask it a set of questions to see how well it has learned the materials using Bloom's taxonomy.

After joining a firm, this on-the-job learning is familiar to all human Cybersecurity analysts. Starting with basic knowledge, they learn about the company and its specific configurations and processes as they do their job.

Figure 13 shows an example of the output created. Only a portion of the output is shown here. Still, one can see that information learned about the client, including the bank's IT landscape information, is incorporated into the answer.

To determine if Log4j affects ABC Bank, a systematic and thorough approach should be taken, particularly given the scale of your organization, which includes 10 data centers across four countries, 20,000 servers, and 3,000 applications. Here's how you can assess the situation effectively:
Steps to Assess the Impact of Log4j (CVE-2021-44228) on ABC Bank:
Step 1: Formulate a Cross-Functional Team
- Action: Assemble a team comprising members from IT, Cybersecurity, application development, compliance, and operations departments.
- Purpose: A diverse team ensures a comprehensive evaluation of the bank's operations and assessment capabilities.

...

Figure 13. Augmentation Agent Output.

Figure 14 exemplifies the mentoring feedback provided. This would be helpful advice for the white-collar professional to review and use in future interactions.

As a mentor guiding you in your role as a Cybersecurity Analyst, I want to emphasize the importance of structured processes like the one you've outlined for assessing the impact of Log4j at ABC Bank. Here are some tailored pieces of advice to enhance your effectiveness in this role:
1. Emphasize Collaboration in Your Cross-Functional Team
While you mentioned forming a cross-functional team, actively foster collaboration among team members. Encourage open communication and regular meetings to discuss progress, hurdles, and insights. This will create a sense of ownership and encourage knowledge sharing, which is invaluable in a complex environment such as yours.
...

Figure 14. Augmentation Agent Mentoring.

We show that information is assimilated over time. We demonstrate that the agent can understand what information is essential to store for later use, additional information about the client, its performance, and information about how the client wishes to be served, and does that in long-term memory across chat sessions.

We use Bloom's Taxonomy to test the augmentation agent as a final exam. These are questions to validate the learning. We provide a qualitative evaluation here, and the full transcript is available at [41].

TABLE I. BLOOM'S TAXONOMY EVALUATION.

Bloom's Taxonomy Step	**Question / Exercise**	**Evaluation**
Remembering	What is the IT profile for ABC Bank	The agent is able to recall the IT profile the user provided.
Understanding	Describe the aspects of ABC Bank that are vulnerable to Log4j	The agent can use the information in the profile to provide an answer.
Analyzing	Creating a strategy for ABC Bank to deal with the Log4j vulnerability	The agent can create a strategy integrating the profile and its understanding of the bank's vulnerability.
Applying	What are the potential impacts for ABC Bank of Log4j, including legal impacts	The agent provides a comprehensive answer.
Understanding	What should ABC Bank have done in preparation for the Log4j vulnerability? Talk about the people, process and tools.	The agent provides a complete retrospective.
Creating	What is the long-term strategy for ABC Bank to ensure similar vulnerabilities are promptly identified and addressed in the future?	The agent provides a structured and comprehensive set of recommendations.

We have shown that we can use the CorpIA framework to create an autonomous agent that enhances the Cybersecurity analyst's performance. We have used Bloom's Taxonomy to test the agent's learning.

D. Agent Workflows

We can use the definition of a cybersecurity analyst in a complex workflow that includes many different roles. The scenario chosen is the creation of a cybersecurity strategy. The task in NLP form is shown in Figure 15. In this figure, one can see the NLP of the task to be solved. It is part of a prompt to an LLM where we also provide the BNF of the ContentCreate language and ask the LLM to create the CC language program. The OpenAI o1 model [42] was used to generate the code.

In this case, it's the creation of a cybersecurity strategy. This is shown in Figure 15. There are inputs required from the Chief Information Officer (CIO) and the Chief Information Security Officer (CISO), followed by a series of reviews from a number of subject matter experts, before the document is updated and sent for final reviews and communication. This is a typical workflow in knowledge work, with a number of workers working on a set of content.

Here is the BNF form of the CC language:

<<Insert BNF here>>

Create the CC language code for the following process:

The available agents are the CIO, a McKinsey bank consultant, a Gartner consultant, an internal bank strategy consultant, the CISO, a lawyer and a communications specialist. Please use all of the roles in the process.

The goal is to create a cybersecurity strategy that can be shared with employees.

Here is the process:

1. The CIO provides the organizational goals
2. The CISO provides goals translated into cybersecurity goals and targets
3. The Cybersecurity Analyst creates a first draft
4. The Gartner consultant reviews the draft for technology industry trends
5. The McKinsey consultant reviews the draft for banking industry trends
6. The lawyer reviews the draft
7. The Cybersecurity Analyst updates the draft with the inputs from the Gartner consultant, the McKinsey consultant and the lawyer
8. The Communication specialist summarizes and creates the communications
9. The CISO releases the draft

Figure 15. Prompt to create the ContentCreate Program

Figure 16 provides the code that was generated by the LLM and can be run by the CC interpreter. Note that the Cybersecurity Analyst was already defined with its specific declarative and procedural knowledge. CorpIA will use this predefined agent as it already exists.

The above shows a complex content creation and review scenario working across many roles, some of which are created as rich CompIA clients, such as the Cybersecurity Analyst.

Moreover, the CC code has been generated using OpenAI's o1 model [42] using a prompt which includes the BNF specification for the language and the description of the task in natural language.

The simplicity of the language and the ability to generate most, if not all, of the code via an LLM translation makes this language accessible to any knowledge worker without the need for programming skills.

```
$ORG_GOALS = CREATE AS $CIO INSTRUCTIONS "Provide the organization's
overall goals." INPUTS "CONTENT"

$CISO_GOALS = CREATE AS $CISO INSTRUCTIONS "Translate the
organizational goals into cybersecurity goals and targets." INPUTS
$ORG_GOALS

$DRAFT_1 = CREATE AS $CybersecurityAnalyst INSTRUCTIONS "Create the
first draft of the cybersecurity strategy." INPUTS $CISO_GOALS

$GARTNER_REVIEW = REVIEW AS $Gartner_Consultant INSTRUCTIONS "Review
the draft focusing on technology industry trends." INPUTS $DRAFT_1

$MCKINSEY_REVIEW = REVIEW AS $McKinsey_Bank_Consultant INSTRUCTIONS
"Review the draft focusing on banking industry trends." INPUTS
$DRAFT_1

$LAWYER_REVIEW = REVIEW AS $Lawyer INSTRUCTIONS "Review the draft
for legal and compliance considerations." INPUTS $DRAFT_1

$UPDATED_DRAFT = UPDATE AS $CybersecurityAnalyst INSTRUCTIONS
"Incorporate feedback from Gartner, McKinsey, and the Lawyer into
the draft." INPUTS $DRAFT_1 CONCAT $GARTNER_REVIEW CONCAT
$MCKINSEY_REVIEW CONCAT $LAWYER_REVIEW

$COMMUNICATIONS = CREATE AS $Communication_Specialist INSTRUCTIONS
"Summarize the updated draft and create communications to share with
employees." INPUTS $UPDATED_DRAFT

$FINAL_STRATEGY = RELEASE AS $CISO INSTRUCTIONS "Release the final
cybersecurity strategy, ready for distribution to employees." INPUTS
$COMMUNICATIONS

PRINT $FINAL_STRATEGY
```

Figure 16. ContentCreate Program of the Procedure

VII. Discussion

We will start by recapping the research's aims, which are outlined below.

- Introduction of the CorpIA architecture for creating AI agents for knowledge workers. This novel architecture simplifies the creation of a knowledge worker agent. We demonstrate several knowledge worker agents developed in the accompanying GitHub repository.
  We have introduced CorpIA, a cognitive architecture for knowledge work. We have introduced the three components of CorpIA:
  - The CorpIA Agent definition allows for the definition of AI agents with the knowledge and the ability to learn.
  - The CorpIA Agent runtime that can run through a 4-step process, Perceive-Reason-Run-Learn loop. This loop allows the agents to gain new information over time, much like on-the-job learning.
  - The ContentCreate language to create workflows and programs using CorpIA agents.
- Enhancement of Human Performance. We demonstrate how AI agents can help human professionals in complex tasks.
  We have shown how the CorpIA agents and the ContentCreate language can create complex workflows that non-programmers can create.
- On The Job Learning of AI Agents. We show how AI agents can learn from interactions. We show these agents can progress through Bloom's taxonomy in practical scenarios.
  We have shown that the CorpIA agents can learn and meet learning objectives as measured by Bloom's Taxonomy. The agents can remember facts, categorize facts, analyze them, and even create new content.

- The ContentCreate language, which allows for the programming of AI agent workflows. This is a simple language that can be easily used by non-programmers.
  We have demonstrated several ContentCreate programs that demonstrate the ease of creating programs from a simple Hello World! program to more complex scenarios involving many roles.
- Source Code. We offer the CorpIA source code for replication, validation and further development.
  We make it available in the public GitHub repository.

CorpIA is a novel cognitive architecture for AI agents in knowledge work. The implementation allows for the simple creation of AI agents, has mechanisms for learning on the job and has a programming language that allows for the orchestration of a number of AI agents.

Just as computer programming evolved from assembly language to high-level languages for speed, efficiency and portability, the same will happen with large language models. CorpIA is such a high-level architecture for the creation of agents and the running of AI agent programs.

VIII. Ethical Considerations

Using frameworks such as CorpIA to augment knowledge workers requires an examination of ethical implications. Responsible AI usage requires examination and understanding of these possible ethical implications.

1) Bias and Fairness
   AI systems will reflect the biases of the data on which they are trained. These biases could perpetuate inequities. To mitigate these risks, domain-specific models trained on known bias-free data may be better choices for language models.
2) Privacy and Data Protection
   Much like existing knowledge workers, AI agents will handle sensitive and proprietary data. These raise issues of privacy and security of that data. Personal Identifable Information (PII) must be anonymized if sent to large language models on the cloud. In regulated environments such as banking, language models that run on-premise as opposed to the cloud may be better choices.
3) Human Autonomy and Oversight
   One important part of ethical AI deployment is the preservation of human autonomy. AI systems such as CorpIA should serve as tools to augment human expertise, with the final decision-making authority retained by human users. One area that will need further study is the over-reliance on AI systems and the loss of critical thinking, especially as AI systems have higher success rates.
4) Impact on Employment and Skills Development
   The automation of certain tasks traditionally performed by human workers may lead to job displacement and skill erosion. This is an open problem, as frameworks such as CorpIA can automate human processes. There are new job possibilities to be able to automate these tasks, audit these processes and continue to provide critical review of the AI.
5) Accountability and Liability
   Determining accountability in cases of erroneous or harmful AI outputs is a complex challenge. Clearly defined protocols must delineate the responsibilities of developers, operators, and organizations.
6) Regulatory and Legal Compliance
   Compliance with existing legal frameworks, such as the General Data Protection Regulation (GDPR) and emerging AI-specific regulations.

These are all important ethical considerations and challenges as systems such as CorpIA are implemented in workplaces.

IX. Conclusion

We have shown that CorpIA can create AI agents that augment knowledge workers through a simple set of parameters. These include declarative knowledge, procedural knowledge, tools and a set of listening cues to add additional information as conversations occur.

We have shown the ability of these CorpIA agents to learn over time. By defining the listening cues, the AI agent is able to add new knowledge to its knowledge base. This is akin to On-The-Job learning that every knowledge worker goes through, learning domain-specific, company-specific, and client-specific details. We have used Bloom's Taxonomy to show that the AI agents can meet higher-level goals, such as creating new novel content in addition to being able to remember and classify facts it has been given in conversations.

Finally, we have shown how a programming language, ContentCreate, can orchestrate workflows amongst a number of AI agents and is simple enough to be able to be done by a non-programmer. The LLM can itself convert the natural language version of the process into code that can be executed by Content Create.

We need to recognize that, though there is progress on AI agent technology with CorpIA, we need to consider the ethical implications of this work. People, Processes and Tools are three elements of any successful implementation. This paper focused on the Tools aspect. There are a number of issues that will have to be resolved for real-world implementation. The people aspects, including training, integration of AI agents into the workflow, allowing human workers agency, and issues of psychological impact, are all aspects that need to be considered. Similarly, there is work to be done on the process side, including accountability and regulatory insight, where AI makes some recommendations.

Though we have shown the potential in the technology in this paper, there is much work to be done before these systems can be productively used at scale in industry, and especially so for regulated industries such as finance.

References

[1] S. Vella, M. Noun, S. Sharieh, and A. Ferworn, "An AI-based cognitive architecture for augmenting cybersecurity analysts", in *Proceedings of the Eighteenth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2024)*, [Online], Nice, France, 2024, pp. 23–27.

[2] N. E. Adams, "Bloom's taxonomy of cognitive learning objectives", *Journal of the Medical Library Association : JMLA*, vol. 103 3, pp. 152–153, 2015.

[3] D. Townsend, "Leveraging Generative AI tools like ChatGPT for startups and small business growth", *Entrepreneur and Innovation Exchange*, 2023.

[4] V. Jüttner, M. Grimmer, and E. Buchmann, "ChatIDS: Explainable cybersecurity using Generative AI", *ArXiv*, vol. abs/2306.14504, 2023, Accessed: 2024-10-10.

[5] J. Schmidhuber, "2006: Celebrating 75 years of AI - history and outlook: The next 25 years", in *50 Years of Artificial Intelligence*, 2007.

[6] Y. Cao *et al.*, "A comprehensive survey of AI-generated content (aigc): A history of Generative AI from GAN to ChatGPT", *ArXiv*, vol. abs/2303.04226, 2023, Accessed: 2024-10-10.

[7] A. Vaswani *et al.*, "Attention is all you need", in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010, ISBN: 9781510860964.

[8] T. B. Brown *et al.*, "Language models are few-shot learners", *ArXiv*, vol. abs/2005.14165, 2020, Accessed: 2024-10-10.

[9] E. Brynjolfsson, D. Li, and L. Raymond, "Generative AI at work", eng, *The Quarterly Journal of Economics*, vol. 140, no. 2, pp. 889–942, 2025, ISSN: 0033-5533.

[10] Q. Gallea, "From mundane to meaningful: AI's influence on work dynamics - evidence from ChatGPT and Stack Overflow", *arXiv*, 2023. DOI: 10.48550/arxiv.2308.11302.

[11] C. Qian *et al.*, "ChatDev: Communicative agents for software development", in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 15 174–15 186. DOI: 10.18653/v1/2024.acl-long.810.

[12] Q. Wu *et al.*, "AutoGen: Enabling next-gen LLM applications via multi-agent conversation", 2023, Accessed: 2024-10-10. arXiv: 2308.08155 [cs.AI].

[13] Y. Cheng *et al.*, "Exploring large language model based intelligent agents: Definitions, methods, and prospects", *ArXiv*, vol. abs/2401.03428, 2024, Accessed: 2024-10-10.

[14] Z. Saleem, A. M. Hanif, U. Sumayya, M. S. Vohra, and H. Raza, "Productivity in the virtual office: Unpacking the role of digital tools and work-life harmony", *International Journal of Social Science Entrepreneurship*, vol. 3, no. 4, pp. 1–19, Oct. 2023.

[15] D. C. Engelbart, *Augmenting Human Intellect: A Conceptual Framework*, Air Force Office of Scientific Research, AFOSR-3233, www . bootstrap . org / augdocs / friedewald030402 / augmentinghumanintellect/ahi62index.html, 1962.

[16] S. Vella and S. Shariah, "All you need is knowledge, experience and tools: A framework for integrating digital labour with white collar work", *2024 4th Interdisciplinary Conference on Electrics and Computer (INTCEC)*, pp. 1–7, 2024.

[17] OpenAI, *ChatGPT*, https://chat.openai.com, Accessed: 2024-10-10, 2024.

[18] L. Wang *et al.*, "A survey on large language model based autonomous agents", *ArXiv*, vol. abs/2308.11432, 2023, Accessed: 2024-10-10.

[19] CODASYL, *Cobol: Common business oriented language*, Developed by CODASYL and sponsored by the U.S. Department of Defense, 1960.

[20] J. Backus and IBM, *Fortran: Formula translation*, Originally developed by IBM for scientific and engineering calculations, 1957.

[21] A. Nuxoll and J. E. Laird, "Enhancing intelligent agents with episodic memory", *Cognitive Systems Research*, vol. 17, pp. 34–48, 2012.

[22] J. Anderson, *The Architecture of Cognition*, 1st. Psychology Press, 1996. DOI: 10.4324/9781315799438.

[23] S. Vella and S. Sharieh, "From novice to expert: On-the-job learning of autonomous llm agents in white-collar labour", in *Information Systems for Intelligent Systems: Proceedings of ISBM 2024, Volume 3*, 2024.

[24] P. Armstrong, "Bloom's Taxonomy. Vanderbilt University center for teaching", Accessed: 2024-10-28.

[25] M. Gupta, C. Akiri, K. Aryal, E. Parker, and L. Praharaj, "From ChatGPT to ThreatGPT: Impact of Generative AI in cybersecurity and privacy", *IEEE Access*, vol. 11, pp. 80 218–80 245, 2023.

[26] S. Metta, I. Chang, J. Parker, M. P. Roman, and A. F. Ehuan, "Generative AI in cybersecurity", *ArXiv*, vol. abs/2405.01674, 2024, Accessed: 2024-10-10.

[27] Y. Yigit, W. J. Buchanan, M. G. Tehrani, and L. A. Maglaras, "Review of Generative AI methods in cybersecurity", *ArXiv*, vol. abs/2403.08701, 2024, Accessed: 2024-10-10.

[28] I. Prieto and B. Blakely, "Proposed uses of Generative AI in a cybersecurity-focused SOAR agent", in *Proceedings of the AAAI Symposium Series*, vol. 2, 2023, pp. 386–390.

[29] GitHub, Inc., *GitHub Copilot*, Accessed: 2023-10-30, 2021.

[30] B. Zhang, P. Liang, X. Zhou, A. Ahmad, and M. Waseem, "Demystifying practices, challenges and expected features of using GitHub Copilot", 2023.

[31] J. Liang *et al.*, "Code as policies: Language model programs for embodied control", in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom: IEEE, 2023, pp. 9493–9500. DOI: 10.1109/ICRA48891.2023.10160591.

[32] J. S. a. J. Nam, "A Survey of Automatic Code Generation from Natural Language", *Journal of Information Processing Systems*, vol. 17, no. 3, pp. 537–555, Jun. 2021. DOI: 10.3745/JIPS.04.0216.

[33] S. M. Miller, "AI: Augmentation, more so than automation", *Asian Management Insights (Singapore Management University) Vol. 5 Iss. 1 (2018) p. 1 - 20 ISSN: 2315-4284*, 2018.

[34] T. Davenport and J. Kirby, *Only Humans Need Apply: Winners & Losers in the Age of Smart Machines*. HarperCollins, 2016, ISBN: 9780062438607.

[35] B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl, "Taxonomy of educational objectives: The classification of educational goals, handbook i: Cognitive domain", 1956.

[36] L. W. Anderson, D. R. Krathwohl, P. W. Airasian, and B. S. Bloom, *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*, Complete edition. New York: Longman, 2001, ISBN: 0-321-08405-5.

[37] *Csrb report on log4j*, Accessed: December 19, 2024, Cybersecurity and Infrastructure Security Agency (CISA), Jul. 2022.

[38] N. I. of Standards and T. (NIST), "Computer security incident handling guide", National Institute of Standards and Technology, Tech. Rep. NIST SP 800-61 Revision 2, 2012, Accessed: December 19, 2024.

[39] R. Silvers *et al.*, *Review of the December 2021 Log4j event*, 2022. DOI: https://doi.org/10.6028/NIST.SP.800-61r2.

[40] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, *Computer security incident handling guide*, 2012-08-06 2012. DOI: https://doi.org/10.6028/NIST.SP.800-61r2.

[41] S. Vella, *Corpia*, GitHub repository, accessed: 2024-10-25, 2024.

[42] OpenAI, *O1*, https://openai.com/o1, Large language model, 2024.