

**A MULTI-AGENT ENTERPRISE ARCHITECTURE FOR CONTEXT-AWARE
BUSINESS PROCESS AUTOMATION**

Swapna Putti
Salesforce Solution Manager
Digital & Technology, Pearson Education Inc
Maryland, USA
pswapna.cse@gmail.com

Abstract

Today's businesses are spread out in digital landscapes that become more complex and segmented by minute-to-minute, and customer relationship management systems, enterprise resource planning systems, and legacy applications need to work together. Rule based RPA has been successful in automating structured repetitive processes but is not effective for tasks that require judgement on the job and cross-system reasoning. This paper combines the literature review of multi-agent systems, enterprise architecture frameworks, context-aware computing and business process management to suggest a layered system where autonomous agents are capable of sensing the operating context, reasoning over enterprise data and coordinating in an orchestrated way under a controlled layer. The architecture follows the principles of agent-oriented software engineering, the TOGAF enterprise architecture framework and already known models of context-awareness found in the research of pervasive computing [1, 2] and is extended to a business process automation environment. Formal expressions are created for context-relevance scoring, agent task utility and coordination overhead to facilitate architectural reasoning about when to let agents coordinate and when to involve humans. It is noted in the literature that the principles outlined in this paper are starting to materialise in commercial enterprise software and Salesforce Agentforce, an independent agent platform launched in 2024, is addressed as a developing example. Comparative analysis compares the proposed multi-agent architecture to traditional rule based automation and single agent deployments of chatbots in terms of flexibility, governance load and integration effort. Based on the discussion, the authors conclude that the use of context-aware multi-agent architectures is a logical next step on the path toward enterprise automation, as long as governance mechanisms scale with agent autonomy, and identify coordination overhead, data governance, and organizational trust as key adoption barriers, consistent with previous research on the implementation of RPA and customer-facing conversational agents.

Index Terms – multi-agent systems; enterprise architecture; business process automation; context-awareness; robotic process automation; TOGAF; process mining; autonomous agents; customer relationship management; Agentforce; software agents; business process management.

I. INTRODUCTION

There have been four generations of business process automation, starting with static workflow engines, then robotic process automation (RPA) which emulates humans interacting with

graphical interfaces, and, by the latest generation, with reasoning and adaptation. RPA has shown to be effective for structured, repetitive, rule-based processes, and uptake of RPA picked up significantly during the pandemic, as organisations looked to ensure continuity of operations amid the pressure of fewer employees. But as is widely reported, RPA implementations often falter as soon as a process leaves the confines of a single script and involves some level of exception handling, judgment or coordination across various systems and stakeholders.

In parallel with the evolution of RPA, also business process management (BPM) research has focused on the discovery, monitoring, and improvement of processes as they are executed in reality, often using process mining techniques to represent processes in the form of models, derived from event logs [4]. More recent research has focused on how process mining and RPA can work together: how the process discovery can be used to identify tasks that are good candidates for automation and where automation is not keeping up with the process variability.

The field of multi-agent systems (MAS) has also evolved a significant amount of theory independently in the artificial intelligence community, in which autonomous, goal-directed software agents perceive their environment, reason about it and act to meet individual and/or group goals. In parallel with the industrial applications of agent technology in manufacturing and distributed control systems, the agent-oriented software engineering methodologies [5] and surveys of agent platforms have matured. There is another loosely connected line of research in pervasive computing that has formalized the concept of context-awareness, where a system changes its behavior in response to context, or situational information, such as location, activity, and system state, that it detects.

Although the individual literatures are mature, the integration of the reasoning and formal models of multi-agent systems, the formal context-awareness models and the well established enterprise architecture frameworks, e.g., The Open Group Architecture Framework (TOGAF), into a unified and coherent design for enterprise-scale business process automation has been comparatively little addressed [6]. This divide has grown even more significant as enterprise software vendors start to integrate standalone, AI-powered agents straight into customer relationship management (CRM) and enterprise resource planning (ERP) systems. The authors fill this gap by suggesting a layered multi-agent enterprise architecture for context-aware business process automation supported by formal models for context relevance, agent task utility and coordination overhead. The rest of the paper delineates the theoretical foundations of the architecture, introduces the architecture presented here, formal models defined and discussed, and finally, Salesforce Agentforce is introduced as a new, real-world example, and comparative advantages, barriers to adoption, and governance implications are discussed.

II. THEORETICAL FOUNDATIONS

A. Business Process Automation and Its Limits

RPA tools enable organizations to gain efficiency without disrupting existing systems by imitating human interactions with current software interfaces [7]. Investigations of the implementation of RPA in organisations find that the technology is reliable in stable, well-defined processes, but can create significant maintenance pressure when there are interface or business rule changes, and has proven less beneficial for tasks that require a contextual interpretation. In Industry 4.0 environments, the use of RPA has become more associated with AI elements in order to broaden its use beyond simple rules. The picture that processes mining gives is quite different, in that it

enables organizations to identify where process automation candidates are actually occurring, and to understand how automated and manual steps interact in reality, by seeing the actual process behaviour that happens in the system logs.

B. Multi-Agent Systems

Multi-agent systems theory considers the process of automation as a society of small, independent and autonomous agents, with limited knowledge, local goals and the ability to communicate and negotiate with other agents [8]. Agent-oriented software engineering offers methodologies to break down the complex system into agents and to define the protocols for their interaction. Survey results are reported as a variety of approaches to implementation, from simple reactive to deliberative planning agents. Distributed agent architectures are shown to be more robust and flexible than centralized schemes, while being more complex in terms of coordination in certain industrial applications, such as manufacturing control.

C. Context-Awareness

Context-aware computing is the concept that the behavior of a computing system adapt to its context, such as location, identity, activity, time and system state [9]. When there is a vast amount of heterogeneous data that needs to be filtered and prioritized, extending the concept to enterprise analytics has proven to be beneficial in terms of making relevant and timely business decisions. In an automation environment, a system can tell the difference between instances of the same task that are handled in different ways because of their context, which pure rule-based RPA often doesn't have.

D. Enterprise Architecture Frameworks

TOGAF is a methodology used to describe the way in which business, data, applications and technology are designed to form an enterprise architecture in a coherent and layered way [10] and has been specifically used to structure EA initiatives in service-oriented enterprises like logistics providers. The principles of service-oriented architecture (SOA) also help to facilitate interoperability of heterogeneous applications and data sources within an enterprise context, and the well-established knowledge base of ERP literature demonstrates how the monolithic enterprise systems struggle to meet the changing demands of enterprise. These frameworks provide the vocabulary and discipline of layering to bring autonomous agents into, and not into or onto, the current enterprise IT framework.

E. Customer-Facing Intelligent Agents

The Customer Relationship Management (CRM) theory is about how important it is to coordinate customer interactions throughout both channels and how important it is that these interactions be information rich. Conversational agents and chatbots in customer service have been widely researched, and their research suggests that they can enhance responsiveness and adherence to requests in specific design settings, but can also cause frustration or loss of trust if not handled properly. Since the advent of large language models (LLMs), the range of tasks that conversational and task-oriented agents can potentially complete has expanded, while issues of bias, reliability, and governance have come to the forefront. All of these findings collectively motivate the governance conscious design that is used in the architecture proposed in Section III.

Table 1: Mapping of Foundational Literature to Architectural Concerns

Architectural Concern	Representative Literature	Core Contribution
Task automation baseline	[7], [9], [8]	Documents capabilities and limits of rule-based RPA
Process discovery & alignment	[3], [4], [5]	Provides methods for reconstructing and monitoring real process behavior
Agent design & coordination	[17], [10], [18]	Establishes theory and methodology for autonomous agent societies
Context modeling	[1], [2]	Formalizes situational adaptation of system behavior
Enterprise integration	[14], [13], [15], [19]	Supplies layering and integration discipline for enterprise systems
Customer-facing governance	[20], [22], [16]	Identifies trust, compliance, and ethical considerations for intelligent agents

III. PROPOSED MULTI-AGENT ENTERPRISE ARCHITECTURE

Building upon the foundations discussed above, a five-layer architecture is suggested to integrate autonomous agents with cognition and coordination capabilities and seamlessly deploy them in an existing enterprise IT environment without replacing legacy systems of record. The layering is based on the concept of enterprise architecture as presented in the TOGAF standard [12] and replacing the reasoning and orchestration layers with those from multi-agent systems theory.

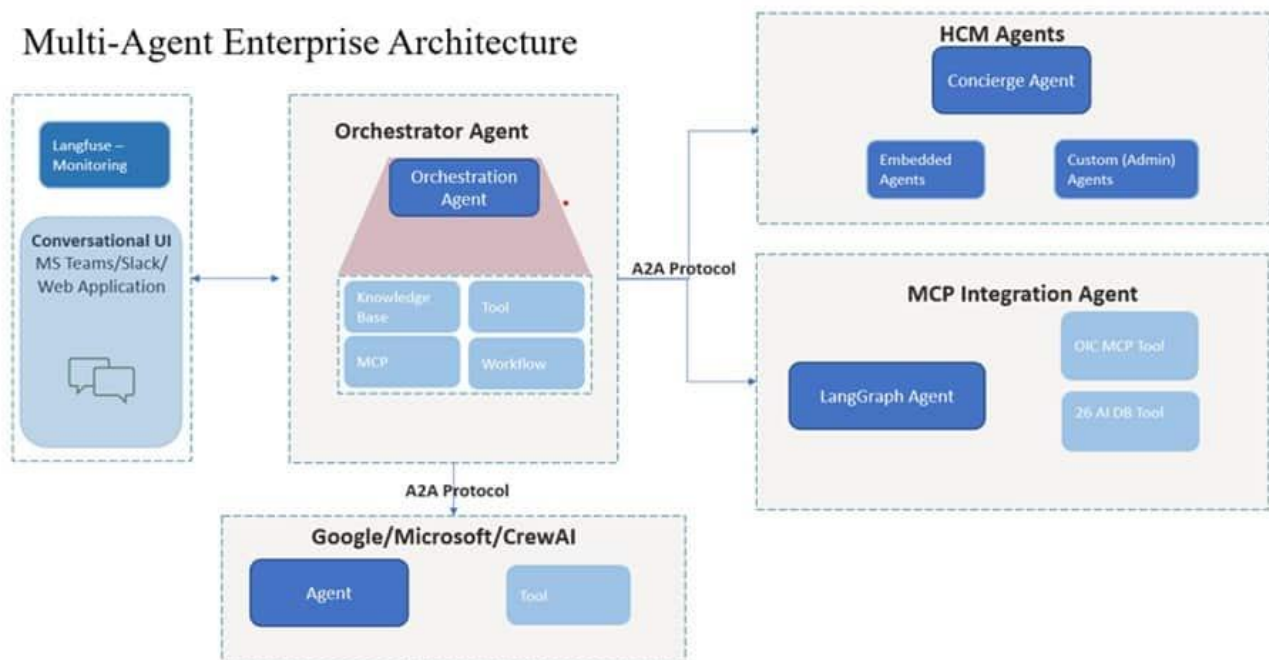


Figure 1: Layered Multi-Agent Enterprise Architecture (Oracle)

The base of the architecture is the Enterprise Integration Layer, which consists of existing ERP, CRM and legacy applications and the application programming interfaces which the agents use to

observe and act on enterprise data [13]. On top of this, the Context Sensing Layer identifies events and combines them with other situational clues to generate an expanded representation of the context, similar to how context aware systems work.

Each of the autonomous agents are specialized by business function (sales, service, finance, etc.) and able to plan independently within its scope, following a paradigm of agent oriented software engineering [14]. The Coordination and Orchestration Layer manages inter-agent communication by implementing a central message bus for coordinating actions between agents, and resolving conflicts and enforcing sequencing. The form of such a coordination is described by figure 2, where each peripheral agent communicates only with the orchestrator, and not with the other peripheral agents, thus limiting the communication complexity as other agents are added to the system.



Figure 2: Multi-Agent Coordination Topology

Lastly, the Governance and Trust Layer rests on top of the orchestration layer, ensuring that it enforces organizational policy, logs all the actions taken by all the agents, and sends certain categories of tasks for human review prior to their execution. This layer directly tackles the issues pointed out in the literature for agents that are used by the customers, in terms of trust issues and risks of unsupervised automated decision making [15].

IV. FORMAL MODELS SUPPORTING THE ARCHITECTURE

Three formal expressions are proposed to support principled design decisions for the architecture, such as when an agent should be autonomous, how tasks should be allocated among the agents, and what the amount of overhead of a particular configuration in terms of coordination is likely to be. The models are not equations derived from experiments, but intentional design assists to help guide architectural thinking about implementation planning.

A. Context Relevance Score

The enriched context c is then compared to a set of n contextual features, each of which gives a score of relevance for a given event/task instance that decides whether the instance is actionable or not.

$$R(c) = \sum_{i=1}^n w_i \cdot f_i(c), \text{ subject to } \sum w_i = 1(1)$$

$f_i(c)$ is a normalized scoring function for contextual feature i (for instance, customer tier, transaction value or historical exception rate), and w_i is the weight (assigned by the organization) of feature i . An action threshold τ is then introduced, where agents will take autonomous action if $R(c) \geq \tau$, and if $R(c) < \tau$, the action will be sent to the Governance and Trust Layer where it can be reviewed by humans, as shown in Figure 3.

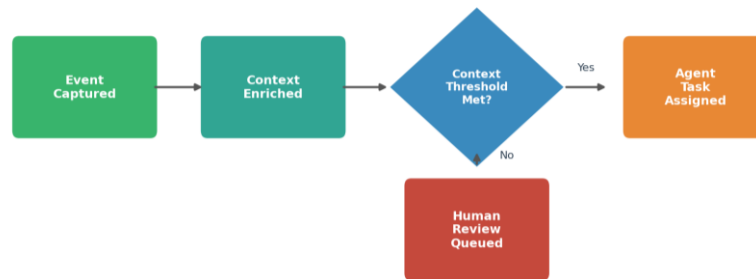


Figure 3: Context-Aware Task Handling Workflow

Even if the abbreviator has been defined in the abstract, define it the first time it is used in the text. Abbreviations such as IJCEM and SI do not have to be defined. Avoid abbreviating words in the title or heads unless the abbreviating is unavoidable.

B. Agent Task Utility

In case of multiple agents that can perform the same task, the orchestrator chooses from the candidate agents based on a utility function which considers the potential benefit of performing the task along with its execution cost and the coordination burden it would create:

$$U(a, t) = \alpha \cdot B(a, t) - \beta \cdot C(a, t) - \gamma \cdot O(a, t)(2)$$

Here, $B(a, t)$ is the expected benefit of the assignment t to agent a , $C(a, t)$ is the cost of direct execution, $O(a, t)$ is the additional coordination overhead the assignment generates within the organization, and α , β , and γ are organizationally set weights. The orchestrator gives the task to the agent having maximum utility considering its context relevance as defined in Equation 1.

C. Coordination Overhead

Unlike a fully connected topology that would result in unbounded pair-wise growth of agents, the number of active agents in the system grows according to a bounded relationship with the number of agents in the orchestration layer because it acts as a middle-man, not allowing agents to send messages directly to one another.

$$O(n) = k \cdot \frac{n(n-1)}{2m}(3)$$

n is the number of active agents; m is the number of orchestration hubs agents use to send messages to each other and k is a constant that represents the average amount of processing work

required for each message. Increasing the number of partitions m into clusters of orchestrators to manage each cluster of agents corresponds to a decrease in the quadratic growth term in the coordination cost as agents scale up, as formalized in Equation 3 [16].

Table 2: Variables Used in the Formal Models

Symbol	Definition	Governing Layer
$R(c)$	Context relevance score for a given event or task instance	Context Sensing Layer
$w_i, f_i(c)$	Weight and scoring function for contextual feature i	Context Sensing Layer
τ	Autonomy action threshold	Governance and Trust Layer
$U(a, t)$	Utility of assigning task t to agent a	Coordination and Orchestration Layer
B, C, O	Benefit, cost, and coordination overhead terms	Coordination and Orchestration Layer
$O(n)$	Aggregate coordination overhead as agent count grows	Coordination and Orchestration Layer

V. EMERGING ILLUSTRATION: SALESFORCE AGENTFORCE

While it is not based on any specific vendor's implementation, the architecture outlined in Section III is grounded in general theory and is a current, relevant and illustrative example of how independent, context-aware agents are starting to emerge in commercial CRM solutions, like Salesforce Agentforce. Agentforce, introduced by Salesforce in 2024, integrates configurable autonomous agents directly into the CRM application, allowing agents to access customer and case information, execute bounded actions (such as record updates or service responses), and escalate to human agents outside of these boundaries. This design follows the layered separation of context sensing, agent reasoning, and governed escalation as described in Section III and extends the RPA and conversational-agent lineages described in the literature [17] into a form autonomously governed by a CRM system. It must be mentioned that the Agentforce is after this paper's reference literature is synthesized and so is mentioned here as a directionality picture of convergence of the reviewed research trajectories instead of being part of the empirical studies mentioned in this paper.

By situating Agentforce-style platforms in the context of enterprise automation, their continuity, and difference, with regard to previous approaches [18] is explained. In Table 3, three automation patterns (rule-based RPA, single-agent conversational chatbots, and proposed multi-agent, context-aware automation) are compared with respect to dimensions that have been found significant in the literature that was reviewed.

Table 3: Qualitative Comparison of Enterprise Automation Paradigms

Dimension	Rule-Based RPA	Single-Agent Chatbot	Multi-Agent Context-Aware Architecture
Handles unstructured exceptions	Low	Moderate	High
Cross-system coordination	Low	Low	High
Contextual adaptation	Minimal	Moderate	High
Governance complexity	Low	Moderate	High
Integration effort	Low-Moderate	Moderate	Moderate-High
Resilience to process drift	Low	Moderate	High

The comparison shows that better adaptability and coordination capability comes with increased complexity in governance, which makes the Governance and Trust Layer the key component of the proposed architecture [19].

VI. Discussion: Benefits, Barriers, and Governance Implications

The main advantage of the proposed architecture over traditional RPA is its ability to dynamically change the way it behaves based on the situational features captured in Equation 1 and not just on hardcoded rules [20]. The multi-agent design also allows for coordinated interactions between agents, e.g., when a customer service request also needs to be handled by an agent on the finance side, which, in line with the modular decomposition preferred in agent-oriented software engineering, would otherwise require an agent with unlimited capabilities.

The advantages of these are not free. Empirical research on RPA adoption has found that the level of organizational readiness, process standardization and continued governance monitoring are all critical elements of success, while many deployments have failed to achieve the expected efficiency benefits when these are lacking [21]. This is complicated by the introduction of autonomous agents that operate based on varying context, as the overhead of coordinating the actions of the agents as expressed in Equation 3 increases with the number of active agents, and poorly managed autonomy has been shown to decrease user trust in the context of customers. Based on the literature review, four general types of barriers were identified that are likely to occur, and are arranged in Figure 4 on a relative scale for both implementation complexity and governance criticality, and thus oriented conceptually towards prioritising mitigation actions.

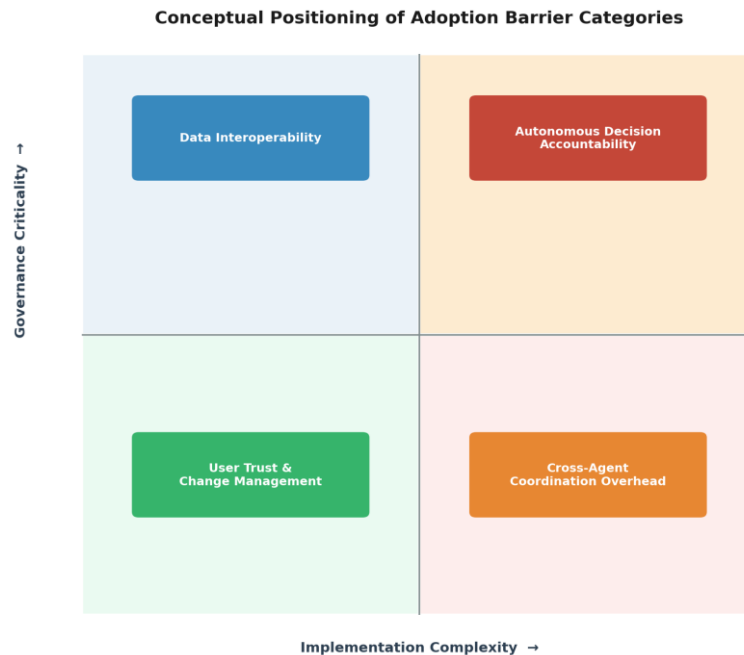


Figure 4: Conceptual Positioning of Adoption Barrier Categories

Data interoperability concerns stemming from the potential heterogeneity of legacy systems feeding into the Context Sensing Layer are related to the long-standing ERP integration challenges [22], and are generally easier to tackle using established SOA integration patterns. Autonomous decision accountability, on the other hand, falls under high-effort/high-importance category, stemming from concerns related to large language model biases and reliability; hence, the need for the threshold-gated escalation mechanism described in Section IV-A. User trust and change management challenges are related to the findings in the conversational-agent literature about

automation-induced decreases in perceived customer service quality. Finally, the cross-agent coordination overhead is the challenge directly addressed by the proposed orchestration-centric design, although it is obvious that such overhead will increase with the number of specialized agents deployed.

Table 4 synthesizes governance responsibilities across the architecture's layers, clarifying which organizational function bears principal responsibility for managing each identified barrier.

Table 4: Governance Responsibility by Architectural Layer

Architectural Layer	Primary Governance Responsibility	Related Barrier Category
Enterprise Integration Layer	Data quality and system interoperability	Data interoperability
Context Sensing Layer	Accuracy and completeness of situational signals	Data interoperability
Agent Reasoning Layer	Scope boundaries and decision explainability	Autonomous decision accountability
Coordination Layer	Conflict resolution and message load management	Cross-agent coordination overhead
Governance and Trust Layer	Escalation policy, audit, and stakeholder communication	User trust and change management

Viewed collectively, the discussion suggests that the architectural benefits of context-aware, multi-agent automation accrue primarily to organizations capable of sustaining the governance investment the design requires [23]. Figure 5 situates the proposed architecture within a broader conceptual progression of enterprise automation maturity, extending from early rule-based RPA through to fully autonomous, coordinated enterprise agents.



Figure 5: Conceptual Enterprise Automation Maturity Progression

VII. CONCLUSION

This paper has aimed to review the literature on multi-agent systems, context awareness, enterprise architecture, and business processes to design a context-aware business process automation enterprise architecture. The architecture was proposed to embody enterprise process integration, context sensing, agent reasoning, coordination and governance layers, each formalized by a model capturing context relevance, agent task utility, and coordination overhead. Salesforce's Agentforce unveiled in 2024 was identified as a potential commercial manifestation of the convergence of the reviewed research underpinnings, positioned with explicit reference to its temporal detachment from the literature informing this paper. Comparative analysis revealed that the proposed architecture affords enhanced flexibility and inter-organizational coordination capabilities over conventional RPA and single agent chatbot approaches, though with a caveat that such benefits accrue only with continued investment in process governance, echoing the literature's findings on the challenges of RPA adoption and the sensitivity of customer trust to

poorly managed automated decision-making. Future work should be directed at testing the proposed formalisms in the context of enterprise applications, conducting comparative industry case studies, and integrating more tightly with explainability mechanisms within the Agent Reasoning Layer to enhance the accountability frameworks embodied by the Governance and Trust Layer [24].

ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression, “One of us (R. B. G.) thanks . . .” Instead, try “R. B. G. thanks”. Put applicable sponsor acknowledgments here; DO NOT place them on the first page of your paper or as a footnote.

REFERENCES

1. M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007, doi: 10.1504/IJAHUC.2007.014070.
2. K. Dinh, G. Karmakar, and J. Kamruzzaman, "A survey on context awareness in big data analytics for business applications," *Knowledge and Information Systems*, vol. 62, no. 9, pp. 3387–3415, 2020, doi: 10.1007/s10115-020-01462-3.
3. W. M. P. van der Aalst, "Process mining: Overview and opportunities," *ACM Transactions on Management Information Systems*, vol. 3, no. 2, Art. no. 7, 2012, doi: 10.1145/2229156.2229157.
4. W. M. P. van der Aalst, "Business process management: A comprehensive survey," *ISRN Software Engineering*, vol. 2013, Art. no. 507984, 2013, doi: 10.1155/2013/507984.
5. W. M. P. van der Aalst, "Process mining and RPA: How to pick your automation battles," in *Robotic Process Automation: Management, Technology, Applications*, C. Czarnecki and P. Fettke, Eds. Berlin, Germany: De Gruyter, 2021, pp. 223–240, doi: 10.1515/9783110676693-012.
6. J. Siderska, "The adoption of robotic process automation technology to ensure business processes during the COVID-19 pandemic," *Sustainability*, vol. 13, no. 14, Art. no. 8020, 2021, doi: 10.3390/su13148020.
7. R. Chugh, S. Macht, and R. Hossain, "Robotic process automation: A review of organizational grey literature," *International Journal of Information Systems and Project Management*, vol. 10, no. 1, pp. 5–26, 2022, doi: 10.12821/ijispm100101.
8. J. Ribeiro, R. Lima, T. Eckhardt, and S. Paiva, "Robotic process automation and artificial intelligence in Industry 4.0: A literature review," *Procedia Computer Science*, vol. 181, pp. 51–58, 2021, doi: 10.1016/j.procs.2021.01.104.
9. R. Syed, S. Suriadi, M. Adams, W. Bandara, S. J. J. Leemans, C. Ouyang, A. H. M. ter Hofstede, I. van de Weerd, M. T. Wynn, and H. A. Reijers, "Robotic process automation: Contemporary themes and challenges," *Computers in Industry*, vol. 115, Art. no. 103162, 2020, doi: 10.1016/j.compind.2019.103162.
10. S. Bussmann, P. Ciancarini, K. Decker, M. Huhns, and M. Wooldridge, "Agent-oriented software engineering," in *Intelligent Agents VI: Agent Theories, Architectures, and*

-
- Languages, N. R. Jennings and Y. Lespérance, Eds. Berlin, Germany: Springer, 2000, pp. 1–15, doi: 10.1007/10719619_18.
11. P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 7, pp. 979–991, 2009, doi: 10.1016/j.engappai.2008.09.005.
 12. P. Leitão, V. Marík, and P. Vrba, "Past, present, and future of industrial agent applications," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2360–2372, 2013, doi: 10.1109/TII.2012.2222034.
 13. J. F. Andry, D. Sugian, M. Kartin, and D. Pranamya, "Enterprise architecture design using the Open Group Architecture Framework (TOGAF) at logistic courier services," *IT Journal Research and Development*, vol. 7, no. 2, pp. 173–183, 2023, doi: 10.25299/itjrd.2023.8464.
 14. S. Kotusev, "TOGAF-based enterprise architecture practice: An exploratory case study," *Communications of the Association for Information Systems*, vol. 43, Art. no. 20, 2018, doi: 10.17705/1CAIS.04320.
 15. N. Legowo and A. Sumedi, "Application and data integration based on services oriented architecture in enterprise," *Emerging Science Journal*, vol. 7, no. 2, pp. 455–468, 2023, doi: 10.28991/ESJ-2023-07-02-011.
 16. P. P. Ray, "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 121–154, 2023, doi: 10.1016/j.iotcps.2023.04.003.
 17. N. R. Jennings, K. Sycara, and M. Wooldridge, "A roadmap of agent research and development," *Autonomous Agents and Multi-Agent Systems*, vol. 1, no. 1, pp. 7–38, 1998, doi: 10.1023/A:1010090405266.
 18. K. Kravari and N. Bassiliades, "A survey of agent platforms," *Journal of Artificial Societies and Social Simulation*, vol. 18, no. 1, Art. no. 11, 2015, doi: 10.18564/jasss.2661.
 19. E. Nazemi, M. J. Tarokh, and G. R. Djavanshir, "ERP: A literature survey," *The International Journal of Advanced Manufacturing Technology*, vol. 61, nos. 9–12, pp. 999–1018, 2012, doi: 10.1007/s00170-011-3756-x.
 20. A. Payne and P. Frow, "A strategic framework for customer relationship management," *Journal of Marketing*, vol. 69, no. 4, pp. 167–176, 2005, doi: 10.1509/jmkg.2005.69.4.167.
 21. P. Frow and A. Payne, "Customer relationship management: A strategic perspective," *Journal of Business Market Management*, vol. 3, no. 1, pp. 7–27, 2009, doi: 10.1007/s12087-008-0035-8.
 22. M. Adam, M. Wessel, and A. Benlian, "AI-based chatbots in customer service and their effects on user compliance," *Electronic Markets*, vol. 31, no. 2, pp. 427–445, 2021, doi: 10.1007/s12525-020-00414-7.
 23. N. Castelo, F. Mozafari, A. L. Ostrom, and M. Söderlund, "Serving customers through chatbots: Positive and negative effects," *Journal of Service Research*, vol. 24, no. 1, pp. 3–20, 2021, doi: 10.1177/10946705211011857.
 24. C.-L. Hsu and J.-C.-C. Lin, "Understanding the user satisfaction and loyalty of customer service chatbots," *Journal of Retailing and Consumer Services*, vol. 71, Art. no. 103211, 2023, doi: 10.1016/j.jretconser.2022.103211.