

CONVERSATIONAL AGENTS FOR DECENTRALIZED B2B ORCHESTRATION IN MANUFACTURING SCM

Goldi Makhija

goldim1309@gmail.com

Abstract—Manufacturing supply chains demand coordination across autonomous partners while preserving local systems and policies; this paper presents a decentralized, agent-driven pipeline that transforms standard B2B interactions into executable conversation policies for automation at scale. The approach ingests UML sequence models and industry documents, derives conversation structure via relational constraints on message roles, and synthesizes role-specific state machines that enact procurement, logistics, and billing exchanges across firms. A prototype aligns with common standards, generates Dooley-style collaboration views, and augments agent behaviors with exception patterns (timeouts, partial commitments, late deliveries) to sustain continuity under operational variability. Case-based evaluation in an automotive-style scenario indicates feasibility for interoperable automation without centralized marketplaces, highlighting pathways for standards convergence and robust exception-aware execution in distributed SCM.

I. INTRODUCTION AND BACKGROUND

Supply chains play a pivotal role in ensuring the seamless flow of goods, services, and information between organizations. In highly competitive industries, disruptions in supply-chain management (SCM) can have significant consequences, including loss of revenue, reduced customer satisfaction, and long-term reputational damage. Studies indicate that firms may lose between 9% and 20% of their market value within six months of a major supply-chain disruption [1]. Such disruptions may arise from shortages of components, excessive inventory, underutilization of production facilities, or ineffective logistics management. These issues underscore the need for robust coordination mechanisms that allow independent actors in the supply chain to interact effectively while maintaining autonomy.

Traditional approaches to SCM often rely on centralized platforms that aggregate buyers and suppliers into a unified marketplace. While such platforms can streamline certain processes, they also introduce dependencies that discourage long-term cooperation and strategic alliances [2]. Centralized architecture concentrates control in a single entity, limiting flexibility and creating potential bottlenecks. In contrast, decentralized frameworks offer resilience and adaptability by distributing coordination tasks among autonomous participants. Decentralization is especially valuable in dynamic industrial ecosystems where firms are reluctant to share sensitive operational data on a common platform.

The advent of intelligent agent technology has enabled new opportunities for decentralized orchestration of business-to-business (B2B) interactions. Agents are autonomous computational entities capable of perceiving their environment,

reasoning, and acting in pursuit of defined objectives [3]. Within the context of SCM, conversational agents provide a natural abstraction for modeling structured interactions such as procurement, order acknowledgment, shipment notifications, and invoicing. These agents transform high-level business processes into executable dialogue policies, enabling automation across distributed organizations.

One of the main strengths of conversational agents lies in their ability to represent business interactions as structured dialogues. By capturing message exchanges and their associated roles, it becomes possible to formalize coordination patterns in a way that ensures consistency, correctness, and interoperability. This dialogue-driven approach extends beyond simple message passing, encompassing rules for sequencing,

exception handling, and conflict resolution [4]. Consequently, conversational agents address both operational and semantic dimensions of collaboration, providing a holistic framework for SCM automation.

Standardization efforts in B2B integration have traditionally focused on document formats and message semantics, with examples such as RosettaNet, ebXML, and the Open Applications Group (OAG) standards [5]. These initiatives ensure that business documents such as purchase orders or invoices follow consistent structures. However, they often fall short of specifying the behavioral logic governing the exchange of such documents. For example, an unacknowledged purchase order could result from disinterest, communication breakdown, or system malfunction. Without explicit behavioral conventions, participants may interpret the situation differently, leading to inefficiencies or disputes.

To overcome these limitations, agent-based approaches integrate linguistic theories of dialogue, formal logic, and computational models of interaction. This integration enables agents to interpret and act upon business processes with a level of flexibility not achievable by static message exchanges alone [6]. By embedding exception handling, adaptive composition, and dynamic role allocation, conversational agents align with the operational realities of modern supply chains. The outcome is a more resilient orchestration mechanism capable of handling variability in both partner behavior and system performance.

Decentralized conversational frameworks are also well-suited to the emerging trend of digital ecosystems, where multiple enterprises collaborate dynamically without fixed hierarchies. In such environments, participants demand direct connectivity

They also require mechanisms for adaptive composition, where new services and partners can be discovered and integrated during runtime [7]. Conversational agents satisfy these requirements by supporting runtime negotiation, flexible role binding, and exception-aware execution of workflows.

Another critical consideration in modern SCM is fault management. As distributed systems grow in complexity, the likelihood of failures—ranging from communication timeouts to incorrect deliveries—also increases. Conversational agents can mitigate these risks by embedding exception-resolution strategies directly into their state machines. By anticipating disruptions and offering predefined recovery behaviors, such agents enhance the robustness of decentralized orchestration frameworks [8]. This proactive approach reduces operational downtime and ensures continuity even under adverse conditions.

The integration of conversational agents into decentralized SCM aligns with broader technological shifts, including the rise of the semantic web and ontology-driven data exchange [9]. By combining semantic interoperability with agent-driven dialogue management, the proposed framework bridges the gap between standardized business artifacts and dynamic, context-aware coordination. This dual capability positions conversational agents as a promising enabler for next-generation supply-chain systems.

In summary, the move from centralized platforms to decentralized agent-driven orchestration addresses critical challenges in SCM. By combining structured dialogue modeling, semantic standardization, and exception handling, conversational agents provide a foundation for resilient, flexible, and interoperable B2B processes. The following sections build upon this foundation by presenting the theoretical underpinnings, framework architecture, exception-handling mechanisms, and evaluation of the proposed approach.

II. THEORETICAL FOUNDATIONS OF CONVERSATIONAL AGENTS IN SCM

The design of conversational agents for supply-chain management (SCM) draws from multiple theoretical perspectives, including distributed systems, computational linguistics, and agent-based modeling. At its core, an agent is defined as an autonomous software entity that perceives its environment, makes decisions, and performs actions to achieve predefined goals [3]. In SCM, these agents are tasked with representing organizational actors such as suppliers, manufacturers, and distributors. They operate as proxies that

negotiate, exchange messages, and enforce commitments, enabling decentralized coordination without reliance on centralized platforms.

One of the foundational concepts for conversational agents in B2B orchestration is *speech act theory*. Originating in linguistics, speech act theory posits that communication is not only about exchanging information but also about performing actions through language [10]. For instance, uttering a statement such as “I confirm the order” functions as an act of commitment. Translating this into SCM, a purchase order acknowledgment can be formalized as a performative act that

changes the state of the business interaction. This theoretical grounding provides a systematic framework for modeling agent dialogues as structured exchanges of commitments and obligations.

Building upon linguistic theories, computational dialogue models have been introduced to formalize business conversations. These models represent interactions as sequences of dialogue moves, each governed by rules of coherence and consistency [6]. In SCM, this translates into predictable sequences of messages such as order placement, acknowledgment, shipping notification, and invoicing. By encoding these dialogue structures into agents, it becomes possible to ensure that business processes are enacted consistently across different organizations, reducing ambiguity and errors.

Another critical theoretical pillar is the notion of *multi-agent systems* (MAS). Multi-agent systems emphasize the collective behavior of distributed agents that coordinate through interaction [11]. In supply chains, MAS provide a natural way to represent independent stakeholders who must collaborate while preserving autonomy. The decentralized nature of MAS aligns with the realities of global supply networks, where no single entity has full control. Coordination strategies such as negotiation, cooperation, and competition can be encoded into agent behaviors to reflect the diversity of industrial relationships.

Interoperability is a persistent challenge in SCM, and semantic technologies offer an essential foundation for addressing it. Ontologies provide shared vocabularies that define the meaning of business concepts, enabling consistent interpretation across organizations [12]. When integrated with conversational agents, ontologies ensure that messages exchanged in dialogues are semantically aligned, regardless of differences in local systems. This semantic foundation complements syntactic standards such as XML or ebXML by embedding meaning into communication, thereby enabling higher-level automation. Formal logic also plays a vital role in specifying and reasoning about agent behavior. Temporal logic, for example, allows the explicit representation of time-dependent constraints, such as deadlines for order acknowledgments or shipment arrivals [6]. Modal logics extend this reasoning by capturing notions such as obligation, permission, and prohibition, which are central to contractual commitments in B2B interactions. Together, these logical frameworks enable the precise specification of agent behaviors and verification of their compliance with business rules.

Theoretical advancements in service-oriented architectures (SOA) further contribute to the foundation of conversational agents. SOA emphasizes modularity, dynamic discovery, and composition of services [13]. Conversational agents extend these principles by adding dialogue-driven orchestration, where services are not only discovered and invoked but also coordinated through structured conversations. This theoretical alignment enables agents to interoperate seamlessly with existing web services while providing higher-level control over business interactions.

Another important influence is the conceptual interoperability model, which identifies different levels of integration across systems, from technical connectivity to full semantic and pragmatic interoperability [14]. Conversational agents operate at the higher levels of this model, addressing not just the technical exchange of data but also the alignment of meaning and intent. By operating at these levels, agents ensure that B2B exchanges are not only syntactically correct but also pragmatically valid, thereby reducing the risk of misinterpretation in distributed SCM.

The theoretical foundations also emphasize exception handling as a core element of robust interaction. Theories of coordination in distributed systems stress that failures are inevitable and must be anticipated [8]. In conversational frameworks, this translates into mechanisms for detecting deviations, diagnosing causes, and enacting recovery strategies. The integration of exception-aware models ensures that conversational agents can handle disruptions such as late deliveries or payment errors without compromising overall process continuity.

Finally, the convergence of semantic web technologies and agent frameworks highlights an emerging theoretical direction. By linking structured ontologies with dialogue models, agents can not only exchange messages but also reason about their meaning and context [15]. This convergence paves the way for intelligent orchestration, where agents dynamically adapt conversations based on contextual cues, historical data, and evolving business conditions. The theoretical integration of language, logic, and semantics thus provides the foundation for building advanced conversational agents capable of supporting decentralized B2B orchestration.

In summary, the theoretical underpinnings of conversational agents in SCM draw from multiple domains, including linguistics, multi-agent systems, semantic technologies, logic, and distributed computing. Together, these perspectives establish a comprehensive foundation that guides the design of decentralized, dialogue-driven frameworks for B2B orchestration. The next section builds on this foundation by presenting the architecture and operational details of the proposed decentralized orchestration framework.

III. PROPOSED DECENTRALIZED ORCHESTRATION FRAMEWORK

The decentralized orchestration framework proposed in this study leverages conversational agents as the primary coordination mechanism for supply-chain management (SCM). The framework transforms high-level business interactions into structured conversations that can be executed by autonomous agents, ensuring interoperability while preserving organizational independence. Unlike centralized marketplaces, the proposed architecture distributes orchestration responsibilities across participants, thereby reducing bottlenecks and enhancing resilience [2].

At the core of the framework is the use of *UML sequence diagrams* to capture the flow of interactions among supply-chain partners. UML provides a widely understood notation for representing message exchanges, making it a suitable foundation for modeling B2B scenarios. These diagrams serve as blueprints that are subsequently translated into formal representations, ensuring that the semantics of business processes are preserved during automation [16]. This approach bridges the gap between business analysts and system developers, enabling collaboration across organizational and technical domains.

The next step involves translating UML representations into ontology-based formats such as DAML-OIL. Ontologies provide a machine-interpretable representation of business concepts and their relationships, allowing agents to reason about messages in a semantically consistent way [5]. By encoding supply-chain interactions in ontological terms, the framework ensures semantic interoperability across heterogeneous systems. This semantic layer complements existing syntactic standards, enabling agents to align message meaning even when local terminologies differ.

From these ontological descriptions, the framework derives structured *conversation tables*. These tables define the sequence and dependencies of messages in a dialogue, capturing relationships such as responses, acknowledgments, and completions. For instance, a purchase order message may require acknowledgment and eventually trigger an invoice. The conversation table formalizes these relationships, ensuring that agents follow predefined conventions while also providing a foundation for automated validation [17]. This step is critical for preventing miscommunication and enforcing transactional correctness.

The conversation tables are then transformed into *bipartite conversation graphs*, which provide a visual and computational representation of the interactions among participants. These graphs highlight the roles of senders and receivers, the dependencies among messages, and the pathways through which exceptions may propagate. The graph-based representation enables systematic analysis of coordination patterns and provides a basis for generating collaboration diagrams, which serve as higher-level views of the orchestration process [1].

Based on the conversation graphs, the framework synthesizes *finite-state machines* (FSMs) that define the role-specific behaviors of conversational agents. Each FSM encode states such as *pending*, *approved*, or *complete*, and transitions triggered by message events. For example, upon sending a purchase order, an agent transitions from *start* to *pending*. Upon receiving acknowledgment, the state shifts to *approved*. This explicit representation of states and transitions ensures that agents can execute workflows in a deterministic and verifiable manner [8].

To increase robustness, FSMs are augmented with exception-handling mechanisms. These include timeout transitions, substitutions for unexpected responses, and recovery paths for failed transactions. By embedding exception handling at the state-machine level, the framework ensures that agents can respond dynamically to disruptions without requiring centralized intervention. This approach aligns with distributed systems theory, which emphasizes resilience through local recovery strategies [7].

The Orchestrian framework also incorporates Monitoring and traceability features. Agents log their interactions, enabling post-hoc analysis of conversations and providing evidence for auditing and compliance. This feature is particularly relevant in regulated industries such as automotive and aerospace, where accountability and transparency are essential [1]. By maintaining detailed records of interactions, the framework supports both operational decision-making and long-term strategic planning.

Scalability is achieved by allowing new partners to join the orchestration dynamically. Since the framework is based on ontologies and standardized dialogue structures, newly introduced agents can discover relevant roles and integrate seamlessly into existing workflows. This adaptive composition capability ensures that the system remains flexible in the face of changing supply-chain configurations, supporting real-time discovery and integration of partners [13].

In summary, the proposed framework operationalizes decentralized B2B orchestration through a structured pipeline: UML modeling, ontological encoding, conversation extraction, graph transformation, and state-machine synthesis. By distributing orchestration responsibilities to conversational agents and embedding exception-handling mechanisms, the framework provides a resilient, scalable, and semantically consistent solution for supply-chain automation. The next section extends this discussion by focusing on exception handling and robustness mechanisms in greater detail.

IV. EXCEPTION HANDLING AND ROBUSTNESS MECHANISMS

In distributed supply-chain management (SCM), exceptions are inevitable. Delays, system outages, communication failures, and incorrect transactions can disrupt the smooth execution of business processes. Without robust mechanisms to anticipate and handle such disruptions, decentralized orchestration would be

prone to breakdowns. Exception handling therefore becomes a central component of conversational agent frameworks, ensuring that processes remain reliable and resilient in the face of uncertainty [8].

The first step in building robustness is to systematically classify potential exceptions. Within B2B interactions, common categories include order issues (e.g., incorrect quantities or quality violations), delivery problems (e.g., late or misplaced shipments), payment anomalies (e.g., delays or inaccurate settlements), and unfulfilled goals (e.g., missed deadlines). By analyzing inverse goal states, exceptions can be anticipated before they occur. This proactive categorization allows agents to embed recovery strategies directly into their state machines, reducing the likelihood of system-wide failures [1].

Once exceptions are categorized, they are integrated into the formal specifications of agent behavior. In the proposed framework, finite-state machines (FSMs) are augmented with additional transitions that capture exception paths. For example, a pending state awaiting acknowledgment may transition to a timeout state if no response is received within a defined interval. Similarly, the receipt of a refusal message may trigger an alternate negotiation pathway rather than terminating

The intersection and These enriched FSMs ensure that agents can respond adaptively under adverse conditions [7].

Exception handling also involves substitution rules that broaden the interpretation of incoming messages. For instance, an expected acknowledgment may be substituted with either a refusal or a timeout, depending on the context. By incorporating substitution rules into the FSM, agents are prepared to recognize and handle deviations from ideal execution paths. This flexibility aligns with real-world practices, where businesses often need to renegotiate terms or find alternative suppliers when disruptions arise [4].

In addition to reactive measures, the framework supports predictive monitoring for early detection of anomalies. Agents maintain logs of their interactions and analyze them for patterns that may indicate emerging risks, such as repeated delays from a specific supplier. These logs provide valuable insights for both operational decision-making and long-term planning, allowing firms to assess partner reliability and adapt strategies accordingly [1]. Monitoring thus extends exception handling from a purely reactive function to a proactive resilience mechanism.

Interoperability plays an important role in robustness, as exceptions often arise at the interface between different systems. By employing ontology-driven representations, conversational agents can interpret exception-related information consistently across heterogeneous environments [5]. For example, a “late delivery” notification encoded in one firm’s system can be understood unambiguously by another firm’s agent, ensuring that exception responses are coordinated effectively. This semantic alignment reduces the risk of miscommunication during crisis scenarios.

The framework also emphasizes escalation strategies for critical failures. While local recovery is preferred, certain disruptions may exceed the capacity of individual agents to resolve. In such cases, escalation mechanisms trigger human intervention or cross-organizational arbitration. By balancing automated responses with escalation protocols, the framework ensures that exceptions do not escalate into systemic breakdowns while preserving accountability [6]. This hybrid approach blends machine autonomy with human oversight, providing flexibility in high-stakes scenarios.

From a systems perspective, exception-aware orchestration enhances fault tolerance and increases trust in decentralized coordination. Partners are more likely to adopt agent-driven frameworks when they are confident that disruptions will be handled transparently and fairly. Exception handling also fosters fairness by ensuring that failures are managed according to agreed-upon conventions rather than ad hoc decisions. This transparency contributes to long-term collaboration and strengthens strategic alliances in supply networks [2].

Another key benefit of exception handling is its contribution to continuous improvement. Each disruption and its resolution generate data that can be fed back into the system for refinement. Over time, agents can learn from these experiences, adapting their recovery strategies and improving prediction accuracy. This adaptive capability aligns with modern trends in

intelligent automation, where systems evolve through iterative learning and feedback loops [15]. Consequently, exception handling is not only about mitigating risks but also about enhancing long-term resilience.

In summary, exception handling transforms conversational agents from passive executors of workflows into proactive managers of uncertainty. By classifying exceptions, embedding recovery paths, supporting monitoring, and enabling semantic interoperability, the framework provides a comprehensive approach to robustness. These mechanisms ensure that decentralized orchestration remains resilient in practice, bridging the gap between theoretical designs and real-world industrial demands. The next section evaluates the performance of this approach through a case study in manufacturing SCM.

V. CASE STUDY AND EVALUATION

To demonstrate the applicability of the proposed decentralized orchestration framework, a case study was conducted in the context of the automotive supply chain. This industry was chosen due to its reliance on complex, multi-tier supplier networks and its susceptibility to disruptions caused by late deliveries, quality defects, or communication failures [1]. The case study focused on modeling procurement, order acknowledgment, shipment notification, and invoicing processes using conversational agents.

The scenario was initially encoded as a UML sequence diagram representing interactions between an automotive manufacturer and two key suppliers. This diagram served as the blueprint for formalization, which was subsequently transformed into DAML-OIL representations. By embedding business semantics into the ontology, the system ensured that the meaning of procurement-related documents was preserved across heterogeneous environments [5]. The use of UML provided a clear link between high-level business processes and executable system specifications.

From the ontology-based representations, structured conversation tables were derived. These tables specified message dependencies, including responses, acknowledgments, and completions. For example, each purchase order required a corresponding acknowledgment and eventually triggered either an invoice or a refusal. The conversation tables were then translated into bipartite graphs that revealed dependencies among agents and highlighted possible exception pathways [16]. The visual representation allowed analysts to validate correctness and completeness before agent deployment.

Finite-state machines (FSMs) were synthesized from the conversation graphs to capture the behavior of each participant. For the manufacturer's agent, states included *Start*, *Pending*, *Approved*, and *Complete*. Transition rules defined the conditions under which the system advanced, such as receiving an acknowledgment or encountering a timeout. Suppliers' agents exhibited similar role-specific FSMs, ensuring alignment of behaviors across participants [11]. The FSM formalization guaranteed deterministic and verifiable execution of business interactions.

Exception handling was explicitly incorporated into the case study. For instance, when a purchase order acknowledgment was not received within the deadline, the manufacturer's agent automatically triggered a substitution transition to a timeout state. Alternative negotiation pathways were also supported in case of refusals, enabling resilience against unexpected outcomes [8]. This embedded fault tolerance significantly reduced the risk of cascading failures across the supply chain. The evaluation of the case study

revealed several advantages of the agent-based approach. First, the framework improved robustness by ensuring continuity despite disruptions. Second, semantic interoperability across firms allowed agents to interpret messages consistently, even when local terminologies differed [12]. Third, scalability was demonstrated by introducing a new supplier agent into the orchestration without requiring redesign of the entire system. The ontology-driven discovery mechanisms supported dynamic integration of partners, addressing the need for adaptive composition in SCM.

Quantitative evaluation was based on metrics such as process completion rates, mean time to recover from disruptions, and communication overhead. Compared to a centralized orchestration baseline, the agent-driven approach showed lower recovery times and higher success rates in the presence of simulated disruptions. This finding supports the claim that decentralization reduces bottlenecks and enhances resilience [2]. Overall, the case study highlights the practical feasibility of the proposed methodology and its potential benefits for complex manufacturing ecosystems.

VI. FUTURE WORK AND CONCLUSIONS

The study presented a decentralized conversational-agent framework for automating B2B processes in supply-chain management. By formalizing UML diagrams into ontologies, deriving conversation structures, and synthesizing FSMs, the framework operationalized structured dialogues among distributed participants. The case study demonstrated its potential to improve interoperability, robustness, and adaptability in industrial contexts [4]. However, the research also revealed several areas that warrant further exploration.

One avenue for future work lies in extending semantic formalization. While DAML-OIL and ontologies were employed to encode business concepts, richer representations such as OWL-S and PSL could enhance reasoning about temporal and logical constraints [15]. These enhancements would allow agents to handle more sophisticated contractual obligations, including conditional payments, progressive shipments, and composite service dependencies.

Another research direction is the integration of blockchain and distributed ledger technologies with conversational agents. Blockchain can provide immutable records of business interactions, thereby enhancing trust and accountability across supply networks [17]. Coupling decentralized orchestration with tamper-proof transaction records would address concerns about opportunistic behavior and facilitate auditing in highly regulated industries.

The framework could also benefit from integration with digital twins and IoT-enabled systems. Digital twins provide real-time synchronization between physical assets and their virtual representations, enabling agents to make more informed decisions about logistics and production planning [7]. Similarly, IoT sensors could feed agents with operational data such as inventory levels or transportation conditions, further improving responsiveness to disruptions.

Scalability remains a central challenge for large-scale adoption. While the case study validated the framework in a limited automotive setting, applying it to global supply chains involving hundreds of stakeholders requires optimization of communication protocols and load-balancing strategies. Research into hierarchical agent organizations and federated coordination models may provide viable solutions [3].

In addition, the adoption of the proposed framework depends on alignment with international standardization efforts. Collaborations with OAG, RosettaNet, and ebXML initiatives will be necessary to ensure consistency with emerging specifications [18]. This alignment would increase confidence among industrial participants and accelerate the deployment of conversational-agent-based orchestration systems.

From a methodological perspective, future evaluations should include comparative studies across multiple industries, such as aerospace, pharmaceuticals, and logistics. These studies would test the generalizability of the approach and uncover domain-specific requirements. Furthermore, longitudinal studies measuring cost savings, reduction in disruptions, and improvement in collaboration could provide empirical evidence of long-term benefits.

In conclusion, conversational agents represent a promising paradigm for decentralized B2B orchestration in supply-chain management. By embedding exception handling, semantic interoperability, and scalable coordination, the framework addresses limitations of centralized systems and provides a pathway toward more resilient digital ecosystems. Continued research into advanced semantics, blockchain integration, IoT support, and standardization will further strengthen its potential to transform global industrial collaboration [6]. The proposed approach thus offers both theoretical contributions and practical implications, paving the way for next-generation supply-chain automation.

REFERENCES

- [1] C. S. Durugbo, "Collaborative supply-chain systems: An overview and future directions," *Computers in Industry*, vol. 63, no. 7, pp. 625–635, 2012.
- [2] C. Bussler, "The role of b2b integration in supply chain management," *IEEE Internet Computing*, vol. 8, no. 4, pp. 70–76, 2004.
- [3] M. Luck, P. McBurney, and C. Preist, "Agent technology: Enabling next generation computing," AgentLink Report, Tech. Rep., 2003.
- [4] J. L. Zhao, H. T. Ho, and S. C. Sethi, "Supply chain coordination with agent-based technologies," *Decision Support Systems*, vol. 46, no. 2, pp. 620–626, 2009.
- [5] A. Dogac, Y. Kabak, and G. Laleci, "Enriching ebxml registries with ontology-based semantics," *Distributed and Parallel Databases*, vol. 20, no. 3, pp. 225–248, 2006.
- [6] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.
- [7] F. Casati, S. Ilnicki, L. Jin, V. Krishnamoorthy, and M.-C. Shan, "Adaptive and dynamic service composition in eflow," in *International Conference on Advanced Information Systems Engineering*. Springer, 2000, pp. 13–31.
- [8] H. Garcia-Molina, M. J. Franklin, and D. Agrawal, "Coordinating transaction management in distributed systems," *ACM Computing Surveys*, vol. 23, no. 3, pp. 227–268, 1991.
- [9] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.
- [10] J. L. Austin, *How to Do Things with Words*. Oxford University Press, 1962.
- [11] M. Wooldridge, *An Introduction to MultiAgent Systems*. John Wiley & Sons, 2009.
- [12] M. Hepp, "Ontology for the semantic web: Recent advances and future directions," in *Handbook on Ontologies*. Springer, 2009, pp. 445–475.
- [13] M. P. Papazoglou and W.-J. van den Heuvel, "Service-oriented design and development methodology," *International Journal of Web Engineering and Technology*, vol. 2, no. 4, pp. 412–442, 2006.
- [14] A. Tolk and J. A. Mugira, "The levels of conceptual interoperability model (lcim)," in *Proc. IEEE Fall Simulation Interoperability Workshop*, 2003, pp. 1–10.
- [15] A. Sheth, C. Ramakrishnan, and C. Thomas, "Semantics for the semantic web: The implicit, the formal and the powerful," *International Journal on Semantic Web and Information Systems*, vol. 1, no. 1, pp. 1–18, 2005.
- [16] D. C. Schmidt, "Model-driven engineering," *IEEE Computer*, vol. 39, no. 2, pp. 25–31, 2006.
- [17] R. T. Fielding and R. N. Taylor, "Principled design of the modern web architecture," *ACM Transactions on Internet Technology*, vol. 2, no. 2,

pp. 115–150, 2002.

- [18] T. Berners-Lee, J. Hendler, and O. Lassila, “The semantic web,” *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.