Automatisation décisionnelle neurosymbolique à l'ère des modèles de raisonnement à grande échelle : médiation contextuelle et architecture multi-agents

Pierre Feillet[1]

[1] IBM France lab

feillet@fr.ibm.com

Résumé

L'automatisation décisionnelle évolue avec l'émergence des modèles de raisonnement à grande échelle (Large Reasoning Models, LRMs), qui restent insuffisants seuls pour répondre aux exigences des contextes industriels réglementés (explicabilité, auditabilité, déterminisme). L'apport central de cet article réside dans la séparation explicite entre raisonnement stochastique et exécution décisionnelle déterministe, rendue possible par une couche de médiation contextuelle basée sur MCP.
Appliquée à des déploiements industriels concrets, nous proposons une approche neurosymbolique de l'Intelligence Décisionnelle combinant LRMs, formalisation logique des politiques métier et architectures multi-agents. Le Model Context Protocol (MCP) étend le domaine d'action des LRMs via l'inférence symbolique et l'optimisation sous contraintes, vers des systèmes décisionnels explicables et de confiance. Nous présentons quatre contributions : (1) un pipeline neurosymbolique d'extraction d'ontologies et de règles depuis des documents métier, (2) une architecture d'exécution hybride basée sur MCP, (3) une approche agentique de supervision continue, et (4) une vision prospective d'architecture full-agentique inspirée des systèmes blackboard.

Mots-clés

AI, Décision, LRM, Agents, Règles, CPLEX, Optimisation, Confiance, Transparence

Abstract

Decision automation is evolving with the emergence of Large Reasoning Models (LRMs), which remain insufficient on their own for regulated industrial contexts requiring explainability, auditability, and determinism. The central contribution of this work is the explicit separation between stochastic reasoning and deterministic decision execution, enabled by a context mediation layer based on MCP.
Drawing on concrete industrial deployments, we propose a neuro-symbolic approach to Decision Intelligence combining LRMs, business policy logic, and multi-agent architectures. The Model Context Protocol (MCP) extends the operational scope of LRMs through symbolic inference and constraint optimization, enabling explainable and trustworthy decision systems. We present four contributions: (1) a neuro-symbolic pipeline for extracting ontologies and rules from business documents, (2) an MCP-based hybrid execution architecture, (3) an agentic approach for continuous supervision, and (4) a prospective full-agentic architecture inspired by blackboard systems.

Keywords

AI, Decision, LRM, Agents, Rules, CPLEX, Optimization, Trust, Transparency

1 Introduction

L'émergence des modèles de raisonnement à grande échelle (Large Reasoning Models, LRMs) transforme profondément les approches d'automatisation décisionnelle. Contrairement aux LLMs classiques orientés vers la génération de texte, les LRMs intègrent des capacités de raisonnement en chaîne (chain-of-thought), leur permettant de décomposer des problèmes complexes et de traiter des documents à haute valeur informationnelle. Ces modèles offrent ainsi des capacités inédites pour interpréter des documents réglementés et assister l'ingénierie des connaissances.

Cependant, leur nature probabiliste limite leur usage direct dans des contextes réglementés nécessitant explicabilité, déterminisme et auditabilité. Les approches symboliques conservent quant à elles un rôle central dans les systèmes décisionnels industriels, notamment via les règles métier et l'optimisation sous contraintes. La convergence de ces deux paradigmes ouvre la voie à une ingénierie décisionnelle neurosymbolique où les modèles stochastiques interagissent avec des mécanismes d'inférence déterministes au sein d'architectures hybrides.
À partir de déploiements industriels conduits par IBM France Lab, nous mettons en lumière cette convergence et proposons un cadre d'intégration structuré entre LRMs, services décisionnels symboliques et architectures agentiques.

1.1 Positionnement scientifique

L'approche neurosymbolique proposée s'inscrit dans un courant de recherche actif visant à combiner apprentissage profond et raisonnement formel [Garcez et al., 2022]. Par rapport aux approches existantes (telles que Neural Theorem Provers [Rocktäschel & Riedel, 2017] ou DeepProbLog [Manhaeve et al., 2018]), notre contribution se distingue par son orientation industrielle : nous ne cherchons pas à entraîner conjointement un modèle neuronal et un système symbolique, mais à extraire des modèles symboliques puis les orchestrer leur collaboration à l'exécution via un protocole standard (MCP). Cette approche permet de déployer des systèmes décisionnels hybrides sans requérir de modification des modèles (LRM) sous-jacents, ce qui la rend compatible avec des contraintes industrielles de gouvernance et de certification.

Par rapport aux approches RAG (Retrieval-Augmented Generation) et aux agents LLM classiques [Yao et al., 2023], notre architecture introduit une délégation explicite vers des moteurs d'inférence symbolique formellement vérifiables, garantissant des propriétés de déterminisme et de reproductibilité absentes dans les approches purement neuronales.

1.2 Contributions

Les principales contributions de cet article sont :

- Une approche neurosymbolique pour l'extraction d'ontologies et de règles logiques à partir de documents métier, implémentée sous forme d'un graphe LangGraph orchestrant des LRMs.
- Une architecture d'exécution hybride basée sur le Model Context Protocol (MCP), réalisant un découplage explicite entre raisonnement stochastique et exécution symbolique déterministe.
- Une perspective agentique pour la supervision continue de l'automatisation décisionnelle, intégrant détection de dérive et recommandation d'amélioration.
- Une vision prospective de plateformes décisionnelles full-agentique inspirée des systèmes blackboard, s'appuyant sur les protocoles A2A et MCP.

2 Extraction neurosymbolique de connaissances à partir de documents métier

2.1 Problématique et motivations

Les politiques métier constituent une source essentielle de connaissance décisionnelle, mais leur formalisation en modèles exécutables reste laborieuse. Elle nécessite typiquement l'intervention d'experts métier et d'analystes spécialisés, engendrant des délais et des coûts élevés. Les LRMs, grâce à leurs capacités de compréhension du langage naturel et de raisonnement structuré, permettent d'accélérer ce processus tout en maintenant un contrôle humain sur le résultat.

2.2 Pipeline proposé

Nous proposons un pipeline neurosymbolique composé de trois étapes principales, implémenté sous la forme d'un graphe LangGraph invoquant successivement un modèle de raisonnement pour effectuer les tâches suivantes :

- Extraction d'entités et de concepts métier depuis le document source.
- Structuration en ontologie décisionnelle (modèle de données typées, signature d'opération, graphe d'évaluation).
- Génération de règles logiques formalisées en Domain Specific Language (DSL) compilable.

Chaque étape permet à l'utilisateur d'intervenir sur le résultat intermédiaire, garantissant un contrôle humain continu. L'objectif est d'accélérer la capture d'un modèle décisionnel exécutable, composé d'un modèle de données typées, d'une signature d'opération, d'un graphe d'évaluation de nœuds logiques interconnectés dans un modèle Decision Model Notation (DMN), et d'un ensemble de règles logiques exprimées dans le DSL.

Dans nos expérimentations internes, ce pipeline permet de réduire significativement le temps de formalisation d'une politique métier (au moins divisé par 10), tout en maintenant un contrôle humain sur la qualité du modèle. Les principales corrections effectuées par les experts concernent la complétude des règles et la précision des conditions logiques, confirmant le rôle du pipeline comme accélérateur plutôt que substitut à l'expertise métier.

2.3 Illustration d'une demande de prêt

Le pipeline est illustré sur un cas de décision d'évaluation de demande de prêt bancaire. La politique métier, fournie en texte, est transformée en un modèle formel décrivant : une entité LoanApplication regroupant le montant, la durée et le revenu de l'emprunteur, et une entité LoanDecision formalisant l'approbation du dossier.

Chaque étape offre à l'utilisateur la possibilité d'intervenir sur les résultats intermédiaires, assurant ainsi un contrôle humain continu tout au long du processus. L'objectif est de faciliter et d'accélérer la formalisation d'un modèle décisionnel exécutable, structuré autour d'un modèle de données typé, d'une signature d'opération, d'un graphe d'évaluation composé de nœuds logiques interconnectés et modélisé selon le standard Decision Model and Notation (DMN), ainsi que d'un ensemble de règles logiques exprimées dans un langage spécifique au domaine (DSL).

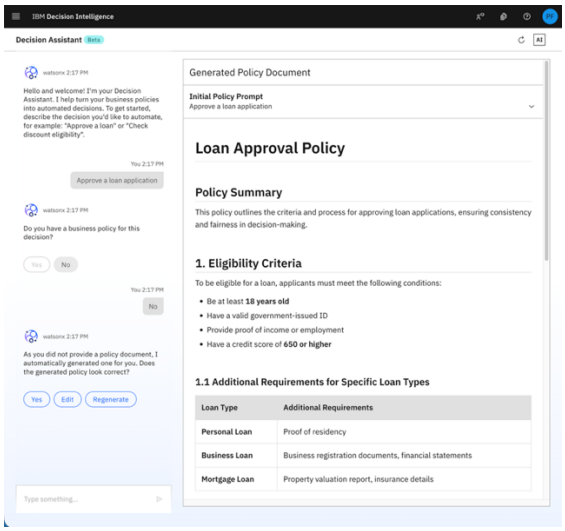

Figure 1: Démarrage d'extraction d'une politique métier

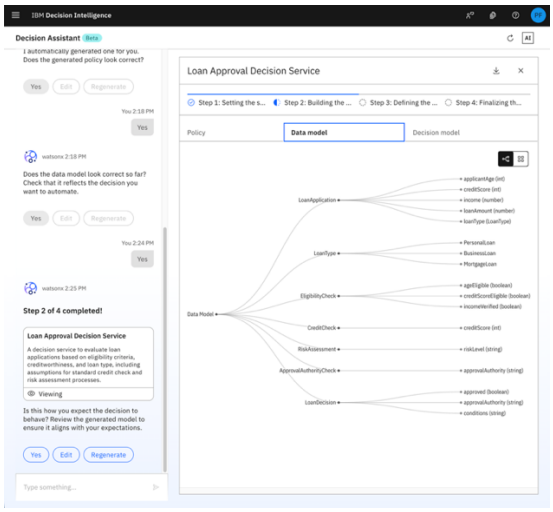

Figure 2 : Extraction d'un modèle de données

Les nœuds définissent les unités logiques et liens les relient pour faire transiter les évaluations vers les nœuds de résultats.

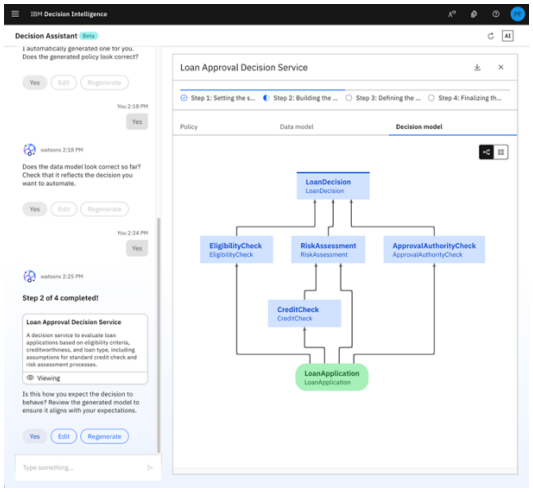

Figure 3 : Extraction d'un modèle de décision

2.4 Extraction de règles logiques

L'extraction focalise son attention sur les règles à créer pour assurer la mission de chaque nœud du graphe décisionnel. Les règles sont découvertes dans une forme naturelle, puis sont formalisées pour être compilables via un parser de Domain Specific Language définit sur le domaine métier et la langue choisie. Chaque règle est structurée sous la forme If condition then action avec des variations de syntaxe en fonction des types manipulés et du DSL.

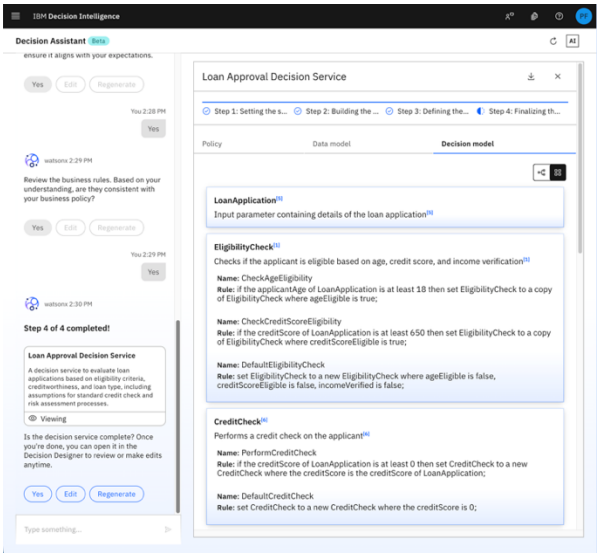

Figure 4 : Extraction des règles logiques

L'extraction est maintenant achevée. Le service décisionnel est maintenant prêt à l'emploi dans l'atelier de modélisation. La prochaine étape consiste à corriger les éventuelles erreurs de syntaxe, et de sémantique.

Cette revue humaine reste critique afin de vérifier l'exactitude, la couverture du modèle, et de le tester sur des cas métiers afin de le valider.

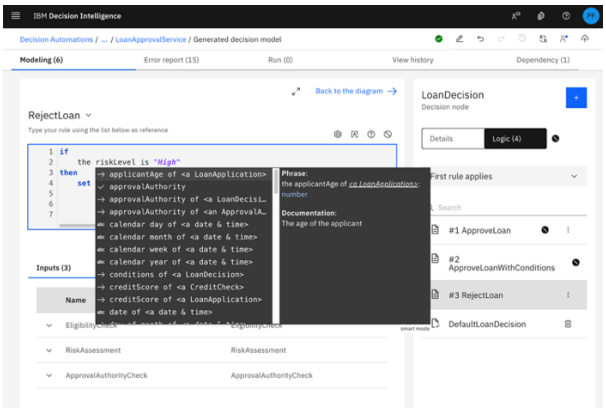

Figure 5 : Edition manuelle des règles en DSL du service décisionnel généré

Le modèle décisionnel obtenu en langue contrôlée permet une maintenance par les experts métiers, et autorise en même temps une compilation du modèle formel en un code exécutable déterministe. Le service décisionnel est alors prêt à être déployé en test puis en production.

2.5 Apport scientifique

Cette approche formalise un processus de *policy-to-ontology reasoning* permettant d'accélérer la conception de modèles décisionnels explicables.

3 Médiation contextuelle et exécution symbolique via MCP

3.1 Problématique : limites des LRMs pour les applications réglementées

Les LRMs présentent plusieurs limitations fondamentales pour les systèmes décisionnels industriels : absence de déterminisme fiable, non-reproductibilité des résultats, et difficulté d'interaction structurée avec des systèmes externes. Dans les environnements réglementés (finance, santé, assurance), ces limitations deviennent rédhibitoires : les organisations ont besoin de systèmes décisionnels auditables, traçables et gouvernables.
Nous faisons l'hypothèse qu'un découplage explicite entre raisonnement stochastique et exécution symbolique est nécessaire à l'élaboration de systèmes décisionnels fiables. Le Model Context Protocol (MCP) [MCP, 2024] est utilisé comme couche de médiation contextuelle afin d'étendre le LRM vers une délégation pour la prise de décisions logiques par des moteurs d'inférence symboliques.

3.2 Architecture proposée

L'architecture repose sur trois composants :

- Un agent conversationnel gérant les interactions utilisateur en langue naturelle, basé sur un LRM (raisonnement stochastique).
- Une couche d'intégration MCP, déclarant pour chaque service décisionnel un ensemble d'outils, ressources et prompts.
- Des moteurs d'inférence symbolique (règles de production via IBM ODM, solveur d'optimisation sous contraintes via IBM CPLEX) encapsulés comme des outils MCP.

Le LRM détecte l'intention décisionnelle depuis le contexte conversationnel, collecte les valeurs des paramètres requis, invoque le service d'inférence via MCP, et réinjecte les résultats dans la conversation. Cela permet une délégation vers des composants déterministes assurant auditabilité et reproductibilité.

3.3 Propriétés garanties

La séparation explicite des responsabilités entre LRM (orchestration conversationnelle) et moteur symbolique (exécution décisionnelle) permet de garantir :

- Déterminisme : pour des entrées identiques, le moteur symbolique produit toujours le même résultat.
- Auditabilité : chaque décision est traçable jusqu'aux règles activées.
- Reproductibilité : les décisions peuvent être rejouées et validées indépendamment.
- Gouvernance : les règles métier restent sous contrôle d'experts métier, indépendamment du modèle LRM.

3.4 Apport

Nous introduisons le concept de context-mediated neuro-symbolic execution, étendant le domaine d'action des LRMs tout en conservant des garanties symboliques formelles. Cette architecture est implémentée et disponible en open source [MCP-ODM, 2024].

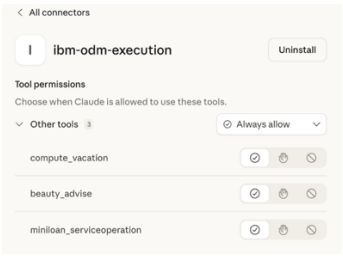

Figure 6 : Enregistrement d'un serveur MCP d'automatisation de décisions

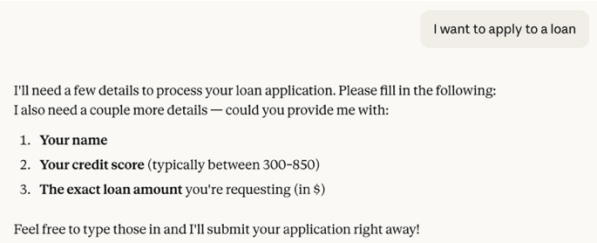

Figure 7 : Routage et collecte des valeurs d'appels vers un service décisionnel d'inférence de règles depuis un assistant conversationnel

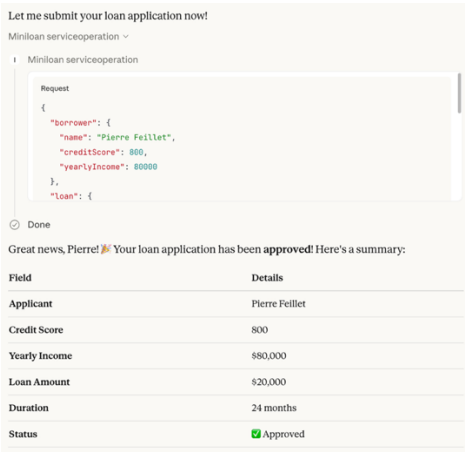

Figure 8 : Invocation d'un service décisionnel d'inférence de règles depuis un assistant conversationnel

4 Perspective d'agents réflexifs et réactifs pour une autonomie opérationnelle

4.1 Motivation

Les systèmes décisionnels opérationnels doivent évoluer constamment pour s'adapter aux changements réglementaires et concurrentiels. Comment tirer parti d'une approche agentique afin de faciliter leur maintenance et leur amélioration, tout en conservant un pilotage métier avec un niveau élevé de confiance dans l'automatisation ?

4.2 Architecture agentique de supervision

Nous augmentons le service d'inférence symbolique de quatre capacités agentiques :

- Observation de suivi opérationnel : collecte des métriques locales par décision et des KPIs métier agrégés.
- Détection de dérive décisionnelle : analyse des écarts par rapport à des objectifs techniques (temps de réponse) ou métier (total de remboursements acceptés).
- Recommandation d'amélioration : proposition de modifications des règles en connaissance de la distribution des valeurs observées.
- Synthèse métier et rapport d'alerte : production d'explications compréhensibles par les équipes métier.

Ces capacités collaborent dans une boucle de rétroaction continue entre usage opérationnel, analyse et proposition d'amélioration, assemblées dans un agent interagissant via une interface conversationnelle.

4.3 Rôle du MCP comme espace médiateur

Dans cette vision, le MCP joue le rôle de couche de médiation entre les différentes capacités agentiques. Il standardise les interactions entre raisonnement stochastique et inférence symbolique en encapsulant le contexte décisionnel partagé. Le MCP agit ainsi comme un espace d'échange structuré proche des architectures blackboard [Corkill, 1991], où les agents publient des observations, des hypothèses ou des résultats intermédiaires.

4.4 Apport

Nous proposons une approche d'agents neurosymboliques automatisant des décisions via une inférence logique invoquable depuis un LRM, et dotés de capacités d'analyse et de réaction pour une autonomie opérationnelle.

5 Vers une architecture « full-agentic »

Le périmètre est ici plus large afin de réinventer le mode opératoire d'opération de l'automatisation décisionnelle dans son ensemble de son cycle de vie. Les architectures blackboard historiques reposaient sur une coordination entre modules spécialisés via un espace partagé. Nous proposons une réinterprétation moderne adaptée aux LRMs et aux systèmes multi-agents, et centrée sur une expérience utilisateur en langue.

5.1 Vision

Le périmètre est ici plus large : réinventer le mode opératoire de l'automatisation décisionnelle dans l'ensemble de son cycle de vie. Les architectures blackboard historiques reposaient sur une coordination entre modules spécialisés via un espace partagé [Hayes-Roth, 1985]. Nous proposons une réinterprétation moderne adaptée aux LRMs et aux systèmes multi-agents.

5.2 Architecture multi-agents proposées

L'architecture couvre l'ensemble du cycle de vie d'automatisation décisionnelle via des agents spécialisés :

- Agent d'ingénierie des connaissances : extraction d'ontologies et de règles à partir de corpus documentaires (Section 2).
- Agent de détection d'écarts : comparaison entre politique textuelle et modèle formel généré.
- Agent de test : génération de suites de tests basées sur la politique et les cas utilisateur.
- Agent d'exécution symbolique : encapsulation des règles métier et solveurs d'optimisation via MCP.
- Agent d'observation : suivi des décisions, collecte de métriques opérationnelles et calcul des KPIs.
- Agent de détection de dérive : analyse évolutive des données et comportements décisionnels.
- Agent de recommandation : proposition d'ajustements de règles basée sur les observations terrain.
- Agent de supervision : orchestration des agents spécialisés via un plan d'exécution dynamique.

L'intégration repose sur des protocoles multi-agents standardisés (A2A [A2A, 2024]) et des frameworks agentiques émergents (LangGraph, BeeAI), chaque agent déclarant ses outils via MCP.

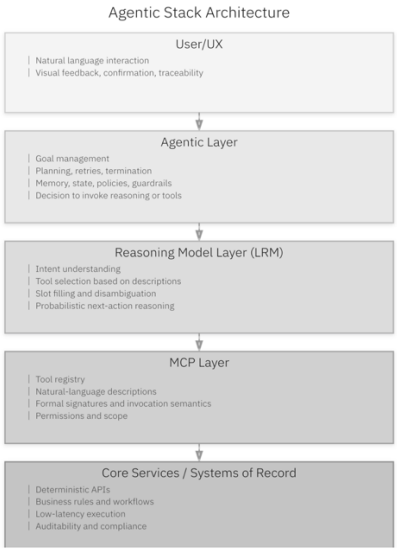

Figure 9: Pile agentique

5.3 DecisionOps neurosymbolique

Cette architecture vise à automatiser l'ensemble des compétences nécessaires à une automatisation décisionnelle de qualité : précision, robustesse, confiance et maîtrise des coûts. La recommandation d'amélioration et le réglage des seuils peuvent être réalisés par des simulations massives ou de l'optimisation pour identifier les valeurs optimales par rapport à une fonction de coût exprimée en termes métier.

6 Conclusion

Nous avons présenté une approche neurosymbolique pour l'automatisation décisionnelle industrielle, combinant extraction de connaissances par LRMs, médiation contextuelle via MCP, supervision agentique et architectures multi-agents inspirées des blackboards. L'expérimentation sur un cas concret d'évaluation de demandes de prêt démontre la faisabilité de l'approche et les bénéfices du découplage explicite entre raisonnement stochastique et exécution symbolique.

Les perspectives incluent : l'évaluation empirique quantitative des architectures agentiques en contexte industriel, la standardisation des interfaces entre LRMs et moteurs symboliques, et l'exploration de mécanismes de coordination distribuée pour des systèmes décisionnels autonomes.

En synthèse, l'ingénierie décisionnelle neurosymbolique s'impose comme une approche clé pour concevoir des systèmes décisionnels déterministes, explicables et gouvernables, capables de combiner la puissance des modèles de raisonnement à grande échelle avec les garanties offertes par l'inférence symbolique. Ces systèmes ouvrent la voie à une automatisation décisionnelle de confiance, accessible en langage naturel via des assistants conversationnels.

7 Références

[A2A, 2024] Google. *Agent-to-Agent (A2A) Protocol Specification*. 2024. Available at: https://a2a-protocol.org/latest/

[BeeAI, 2024] IBM Research. *BeeAI Framework*. 2024. Available at: https://beeai.dev/

[Corkill, 1991] Corkill, D. D. "Blackboard Systems." *AI Expert*, vol. 6, no. 9, pp. 40–47, 1991.

[Garcez et al., 2022] d'Avila Garcez, A., Lamb, L. C. "Neurosymbolic AI: The 3rd Wave." *Artificial Intelligence Review*, vol. 56, pp. 1255–1287, 2023.

[Hayes-Roth, 1985] Hayes-Roth, B. "A Blackboard Architecture for Control." *Artificial Intelligence*, vol. 26, no. 3, pp. 251–321, 1985.

[IBM-DI, 2024] IBM. *IBM Decision Intelligence*. 2024. Available at: https://www.ibm.com/products/decision-intelligence

[LangGraph, 2024] LangChain Inc. *LangGraph Framework*. 2024. Available at: https://www.langchain.com/langgraph

[Manhaeve et al., 2018] Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., De Raedt, L. "DeepProbLog: Neural Probabilistic Logic Programming." In: *Advances in Neural Information Processing Systems (NeurIPS 2018)*, 2018.

[MCP, 2024] Anthropic. *Model Context Protocol Specification*. 2024. Available at: https://modelcontextprotocol.io

[MCP-DI, 2024] Orsini, P. *MCP Server for IBM Decision Intelligence*. GitHub repository, 2025. Available at: https://github.com/DecisionsDev/ibm-decision-intelligence-mcp-server

[MCP-ODM, 2024] Grateau, P. *MCP Server for IBM ODM Execution*. GitHub repository, 2025. Available at: https://github.com/DecisionsDev/ibm-odm-decision-mcp-server

[OpenAgents, 2023] Wang, X., et al. OpenAgents: An Open Platform for Language Agents in the Wild. arXiv:2310.10634, 2023.

[Rocktäschel & Riedel, 2017] Rocktäschel, T., Riedel, S. "End-to-end Differentiable Proving." In: *Advances in Neural Information Processing Systems (NeurIPS 2017)*, 2017.

[Yao et al., 2023] Yao, S., Zhao, J., Yu, D., et al. "ReAct: Synergizing Reasoning and Acting in Language Models." In: *International Conference on Learning Representations (ICLR 2023)*, 2023.