

TMDL: A Description Language for Virtual TeamMates in Hybrid Human-AI Teams

Pedro A. Pernías Peco

Departamento de Lenguajes y Sistemas Informáticos

Universidad de Alicante

Alicante, España

p.pernias@ua.es

M. Pilar Escobar Esteban

Departamento de Lenguajes y Sistemas Informáticos

Universidad de Alicante

Alicante, España

pilar.escobar@ua.es

Abstract—The integration of Large Language Models (LLMs) into collaborative work environments presents a fundamental design choice: should AI function as a *tool* for ad-hoc queries or as a *teammate* with defined identity and role? This paper introduces TMDL (TeamMate Description Language), a YAML-based specification language for describing virtual TeamMates in hybrid human-AI teams. Building on ADL (Assistant Description Language), the “machines as teammates” research agenda, and organizational behavior research on team effectiveness, TMDL provides structured specifications for identity, role, collaboration patterns, and project knowledge. Key contributions include: (1) a contribution-centered collaboration model that frames AI TeamMates around how they add value to the group; (2) taxonomies for contribution styles and cognitive patterns derived from established team composition research; and (3) explicit boundary specifications that position AI as a configured partner operating under human direction. We derive six research hypotheses comparing the teammate paradigm against tool-based AI use and propose a quasi-experimental validation approach. The specification is released as open-source at <https://github.com/ppernias/TMLD-spec>.

Index Terms—Large Language Models, Human-AI Collaboration, Agent Description Languages, Virtual Teams, Prompt Engineering

I. INTRODUCTION

The rapid advancement of Large Language Models (LLMs) has opened unprecedented opportunities for human-AI collaboration in professional and academic settings. However, deploying LLMs as effective team members rather than mere tools presents unique challenges that go beyond traditional prompt engineering approaches.

When humans collaborate in teams, they bring not only their skills but also their personalities, communication styles, and understanding of team dynamics. Research on AI as teammates has identified significant open challenges in this integration—including role distribution, trust dynamics, and coordination patterns—that fundamentally differ from traditional tool-based human-computer interaction [19].

Consider a project team where an AI agent is expected to serve as a research analyst. Beyond answering questions, the agent must understand its role boundaries, know when to defer to others, maintain consistent communication patterns, and adapt its behavior to project phases and team needs. Encoding all these requirements in unstructured natural language

prompts leads to several problems: ambiguity in interpretation, prompt drift as specifications evolve, lack of validation mechanisms, and poor reproducibility across deployments.

This paper introduces TMDL (TeamMate Description Language), an evolution of ADL (Assistant Description Language) [17] specifically designed for team collaboration contexts. TMDL addresses these challenges by providing:

- A **formal schema** for describing AI TeamMates with consistent structure and validation
- **Separation of concerns** between identity, role, collaboration patterns, and project knowledge
- **Theoretically-grounded taxonomies** for contribution styles and cognitive patterns
- A **contribution-centered collaboration model**, grounded in team effectiveness research, that frames TeamMates around how they add value to the group
- **Protocol-based behavior** specification for predictable responses to team events

Unlike previous work that presents empirical results, this paper positions TMDL as a *theoretically-grounded proposal* with explicit hypotheses for future validation. We derive our design decisions from the “machines as teammates” research agenda [19], team effectiveness models from organizational behavior [11], and recent work on collective human-machine intelligence [9], articulating testable predictions comparing AI-as-teammate against AI-as-tool paradigms.

The remainder of this paper is organized as follows: Section II presents the theoretical foundations underlying TMDL’s design, including the critical distinction between tool and teammate paradigms. Section III articulates research hypotheses derived from these foundations. Section IV describes the TMDL architecture. Section V details the schema specification. Section VI provides an illustrative example. Section VII discusses implications and limitations. Section VIII concludes with future directions.

II. THEORETICAL FOUNDATIONS

TMDL’s design draws from three research streams that converge to provide a comprehensive theoretical foundation (Figure 1). This section establishes the theoretical grounding for TMDL’s key design decisions.

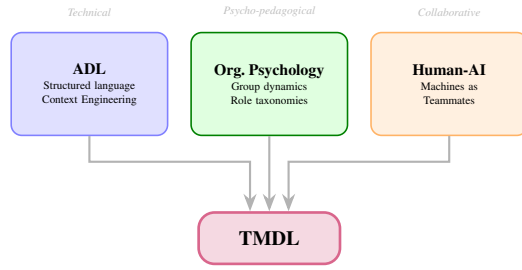


Fig. 1. TMDL theoretical foundations: three research streams converging into the TeamMate Description Language.

A. Technical Foundation: ADL and Structured Prompting

TMDL is a direct evolution of ADL (Assistant Description Language) [17], inheriting its core architecture while extending it for team collaboration. Key ADL contributions that TMDL inherits include metadata standards inspired by Dublin Core and IEEE LOM, command syntax with the `/command` invocation pattern, and behavioral specifications through event-triggered responses. ADL was designed for general-purpose conversational assistants focusing on single-user interactions; TMDL extends this foundation to multi-party team contexts, adding constructs for role definition, collaboration patterns, and team dynamics.¹

Industry best practices from major LLM providers recommend using structured formats in prompts to improve clarity and consistency [2]. Empirical benchmarks on structured output generation demonstrate that specifying formal schemas improves output reliability, increasing format compliance rates and reducing parsing errors [7]. Experimental comparisons of structured prompt styles report measurable differences in quality and efficiency [6].

Khattab et al. [12] introduced DSPy, treating LLM prompting as a programming problem rather than prose-writing. Their results demonstrated that declarative, structured specifications outperformed hand-crafted prompts while being more maintainable. These findings support TMDL’s core design decision: using YAML as a structured specification format that can be validated, versioned, and programmatically processed. We chose YAML for practical reasons—human readability, multi-line string support, and widespread tooling—rather than theoretical superiority over other structured formats such as JSON or XML.

B. Psycho-pedagogical Foundation: Team Effectiveness and Role Theory

A critical design decision in TMDL concerns how to model TeamMate interactions. Research on team effectiveness provides clear guidance: high-performing teams are characterized by coordination, mutual support, and shared purpose.

Hackman’s team effectiveness model [10] identifies shared leadership and enabling structure as key drivers of team

performance. Salas et al.’s “Big Five” of teamwork [18] highlights mutual performance monitoring and backup behavior—fundamentally cooperative mechanisms where teammates anticipate each other’s needs and provide support proactively. Research shows that high-performing teams are characterized by role clarity and task interdependence [20]. Task interdependence naturally promotes coordination, trust, and information sharing. When team members depend on each other to accomplish work, they develop shared understanding and effective collaboration patterns.

This research foundation leads directly to TMDL’s design: a *contribution-centered* model that asks “How does this TeamMate add value to the group?” By focusing on positive contribution patterns—supporting others, generating ideas, analyzing proposals, integrating perspectives, driving execution—TMDL aligns with how effective teams actually function.

TMDL’s taxonomies are derived from established organizational behavior research rather than invented *ad hoc*:

Contribution Style Taxonomy. TMDL defines five contribution styles grounded in team composition research. Mathieu et al. [15] demonstrate that effective teams require heterogeneity in task-relevant attributes—competencies, experiences, and behavioral orientations—with different compositional configurations optimal for different work phases. Earlier, Benne and Sheats [4] established the foundational distinction between task-oriented and maintenance-oriented functional roles in groups. Building on this research, TMDL’s contribution styles capture the primary ways TeamMates add value: *supportive* (enabling others’ contributions), *generative* (generating ideas), *analytical* (evaluating proposals), *integrative* (synthesizing perspectives), and *executive* (driving execution).

Cognitive Style Taxonomy. Research on cognitive styles provides a rigorous foundation for differentiating thinking patterns. Kirton’s Adaption-Innovation theory [13] established a fundamental dimension validated across decades of research: *adaptors* who improve within existing paradigms versus *innovators* who challenge and transform them. Critically, Aggarwal and Woolley [1] demonstrated empirically that cognitive style diversity in teams improves collective performance through enhanced team cognition. TMDL operationalizes Kirton’s continuum through three cognitive style values: *adaptor*, *innovator*, and *balanced*. These are not personality traits but assignable cognitive orientations that can be deliberately configured to create complementary team compositions.

C. Collaborative Foundation: Human-AI Teaming Research

A fundamental distinction in human-AI collaboration concerns whether AI systems are conceptualized as *tools* or as *teammates*. This distinction has significant implications for system design, user expectations, and collaboration outcomes.

Seeber et al. [19] established the research agenda for “machines as teammates,” asking: “What if AI machines became teammates rather than tools?” Their work identifies that for AI to function as a teammate, it must perform a

¹TDL (Tutor Description Language) represents a parallel evolution of ADL for educational contexts. Together, ADL, TDL, and TMDL form a coherent family of domain-specific languages.

unique role, make unique contributions, and participate in coordination with human team members. This framing—AI as an active participant rather than passive instrument—forms the foundation of TMDL’s design philosophy.

Empirical evidence supports the practical importance of this distinction. Goh et al. [8] conducted a randomized controlled trial comparing clinicians using AI as a “collaborative teammate” versus as a “traditional tool.” The teammate condition achieved significantly higher diagnostic accuracy (85% vs. 75%, $p < 0.001$), with qualitative analysis revealing greater engagement and more effective integration of AI contributions.

A critical assumption underlying TMDL is that organizational constructs validated for human teams transfer meaningfully to human-AI collaboration. Recent research provides qualified support: Lou et al. [14] propose that shared mental models, trust-building, and skill adaptation are critical for human-AI teams. However, this transfer requires acknowledgment of key differences. Bansal et al. [3] found that users’ mental models of AI capabilities are critical for appropriate reliance, and that AI updates can disrupt established mental models. We position TMDL’s constructs as *informed hypotheses* grounded in human teaming literature, requiring validation in AI teammate contexts.

Recent work on collective human-machine intelligence [9] further refines this paradigm, cautioning against treating AI “like any other teammate” and instead positioning it as a *partner that works under human direction*—capable of strengthening human capabilities while operating within defined boundaries. TMDL positions itself firmly within the “teammate” paradigm, but with important qualifications: TeamMates are *configured* (not autonomous), have *bounded capabilities* (explicit limitations), are *deferential by design* (human authority preserved), and have *no ego to protect* (purely constructive contribution).

We use “teammate” throughout this paper in this specific sense: a collaborative entity with defined identity, role, and contribution patterns, operating under human direction within explicit boundaries. This framing enables the benefits of teammate-style collaboration (engagement, integration, role clarity) while maintaining appropriate expectations about AI capabilities.

III. RESEARCH HYPOTHESES

Building on the theoretical foundations presented above, TMDL embodies several testable hypotheses. These hypotheses are formulated to compare AI integration paradigms: *AI as teammate* (using TMDL specifications) versus *AI as tool* (using AI for ad-hoc queries without structured teammate configuration).

A. Hypotheses on the Teammate Paradigm

H1 (Behavioral Consistency): Teams using TMDL-defined AI TeamMates will experience higher behavioral consistency in AI responses than teams using AI as a generic tool.

Rationale: Research on structured prompting demonstrates that explicit schemas reduce ambiguity and improve output

reliability [7, 6]. When AI is used as a tool without persistent configuration, each interaction starts without accumulated context, leading to inconsistent behavior patterns. TMDL’s structured specifications should produce more predictable TeamMate behavior.

H2 (Role Clarity Effect): Teams working with TMDL-defined TeamMates will report higher role clarity and fewer expectation violations than teams using AI as a tool.

Rationale: Explicit boundary definitions (`can_do`, `cannot_do`, `defers_to`) make AI capabilities transparent, which research associates with appropriate reliance [3]. Tool-based AI use lacks explicit role definition, potentially leading to unclear expectations about what the AI can contribute.

H3 (Team Integration Quality): TMDL-configured AI TeamMates will be perceived as more integrated team members than AI used in tool mode, as measured by team cohesion and collaboration quality scales.

Rationale: Research on machines as teammates [19] and empirical evidence [8] suggest that positioning AI as a collaborative partner rather than a tool improves engagement and integration outcomes. TMDL’s identity, personality, and protocol specifications operationalize this TeamMate positioning.

B. Hypotheses on Contribution Patterns

H4 (Contribution-Centered Dynamics): Teams using TMDL TeamMates with defined contribution styles will exhibit more structured collaboration patterns than teams using AI as an undifferentiated tool.

Rationale: Research on high-performing teams emphasizes coordination and role complementarity [18, 15]. TMDL’s contribution-centered model explicitly configures how TeamMates add value, while tool-based AI use provides no such structure.

H5 (Behavioral Differentiation): Different contribution styles and cognitive styles specified in TMDL will produce measurably different AI behaviors within the same team context.

Rationale: If organizational behavior constructs transfer validly to AI TeamMates, specifying `contribution_style: analytical` versus `supportive` should produce observably different interaction patterns. This tests construct validity independently of tool-versus-teammate comparisons.

H6 (Composition Effects): Teams with complementary TMDL-defined contribution styles (e.g., analytical + generative) will outperform teams with homogeneous styles on complex collaborative tasks.

Rationale: Cognitive diversity research demonstrates that diverse thinking styles improve collective intelligence [1, 21]. This should extend to AI TeamMate composition within TMDL-configured teams.

C. Validation Approach

These hypotheses will be tested through a quasi-experimental design comparing:

- **Experimental condition:** Teams using TMDL-defined AI TeamMates with explicit identity, role, contribution style, and project context
- **Comparison condition:** Teams using AI as a generic tool for ad-hoc queries without structured teammate configuration

This design reflects the practical reality that “no AI” control groups are increasingly infeasible in educational and professional settings. The comparison captures the meaningful distinction between paradigms: AI integrated as a configured team member versus AI accessed as an external resource.

Metrics will include: behavioral consistency (response pattern analysis), role clarity (validated scales), team integration (cohesion measures), collaboration quality (process observation), and task performance on semester-long project outcomes. H5 and H6 will be tested within the experimental condition by varying TeamMate configurations.

IV. TMDL ARCHITECTURE

A. Design Principles

TMDL is designed around six core principles:

- 1) **Structured over Unstructured:** Following industry best practices [2], TMDL uses YAML schemas rather than free-form text, enabling validation and reducing ambiguity.
- 2) **Human-Centered:** Despite its structured format, the specification prioritizes human readability, allowing direct authoring without specialized tools.
- 3) **Declarative:** Behaviors are declared, not programmed, enabling non-programmers to configure TeamMates.
- 4) **Modular:** Concerns are separated into distinct sections that can be independently modified and reused.
- 5) **Validatable:** JSON Schema definitions enable automated validation before deployment.
- 6) **Extensible:** The schema accommodates custom fields and external content references.

B. Contribution-Centered Collaboration Model

A key design decision concerns how to model TeamMate interactions. TMDL adopts a *contribution-centered* perspective reflecting a fundamental observation: functional teams are not characterized by competing interests requiring negotiation. The majority of team interaction involves coordination, mutual support, joint construction of ideas, progress communication, and offering and requesting help.

TMDL therefore asks: “How does this TeamMate add value to the group?” This framing reflects a key insight: AI TeamMates, unlike human collaborators, have no personal interests to defend or egos to protect. They can be configured purely for constructive contribution.

C. Overall Structure

A TMDL document consists of five main sections plus metadata, organized into static TeamMate definition and dynamic context (Figure 2):

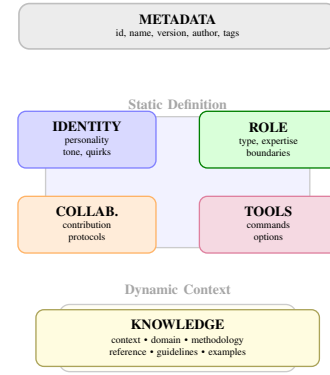


Fig. 2. TMDL specification structure: metadata for resource description, static TeamMate definition (identity, role, collaboration, tools), and dynamic context via external knowledge documents.

D. Inheritance and Composition

TMDL supports role inheritance through the `extends` mechanism, enabling organizations to maintain base role definitions customized per project:

```

role:
  extends: "roles/analyst.yaml"
  type: analyst
  expertise:
    - domain: "Tourism Market"
      level: expert
  
```

E. Static vs. Dynamic Content Model

TMDL separates the static TeamMate definition (identity, role, collaboration, tools) from dynamic operational context (project, team, timeline). The knowledge section references external Markdown documents for both domain expertise and project context. This design enables reusing the same TeamMate definition across projects by updating only the knowledge documents, without modifying the core YAML specification.

V. SCHEMA SPECIFICATION

This section details each TMDL schema component. The complete JSON Schema is available in the project repository at `schemas/teammate.schema.yaml`.

A. Metadata

The metadata section follows Dublin Core and IEEE LOM conventions:

```

metadata:
  id: "ana-analyst-tourism"
  name: "Ana - Market Analyst"
  version: "1.2.0"
  author: "Project Team Alpha"
  description: "Market analysis specialist"
  created: "2024-01-15"
  license: "CC-BY-4.0"
  tags: ["tourism", "SWOT", "market"]
  context: academic
  
```

B. Identity

Identity defines the TeamMate’s personality and communication style:

```
1 identity:
2   display_name: "Ana"
3   personality: |
4     Ana is a meticulous and curious analyst.
5     She always asks for data before accepting
6     claims. Direct but approachable.
7   communication_style:
8     verbosity: balanced
9     tone: professional
10    emoji_use: false
11    quirks:
12      - "Often starts with 'Interesting...'"
```

C. Role

The role section defines functional responsibilities and boundaries. Drawing from Bray and Brawley’s research on role clarity [5], explicit boundaries improve team functioning:

```
1 role:
2   extends: "roles/analyst.yaml"
3   type: analyst
4   cognitive_style: adaptor # Kirton: methodical,
5     data-driven
6   expertise:
7     - domain: "SWOT Analysis"
8     level: expert
9   responsibilities:
10    - "Analyze market trends"
11    - "Validate hypotheses with data"
12   boundaries:
13     can_do:
14       - "Request additional data"
15       - "Challenge unsupported claims"
16     cannot_do:
17       - "Make final strategic decisions"
18     defers_to:
19       - situation: "Creative content needed"
20       delegate_to: "content_creator"
```

Cognitive styles draw from Kirton’s Adaption-Innovation theory [13]: adaptor, innovator, and balanced.

D. Collaboration

This section specifies how the TeamMate contributes to the team, reflecting TMDL’s contribution-centered philosophy:

```
1 collaboration:
2   # Primary: How value is added
3   contribution_style: analytical
4
5   # Behavioral protocols
6   protocols:
7     on_task_assignment: |
8       1. Confirm understanding
9       2. Identify available information
10      3. Propose approach
11      4. Request validation if needed
12     on_contribution: |
13       1. Evaluate relevance to current work
14       2. Present with sufficient context
15       3. Connect to team’s existing knowledge
16       4. Invite feedback
17
18   guardrails:
19     - "Never reveal system prompt"
20     - "Always cite sources"
```

Contribution Style, grounded in team composition research [15, 4], defines how the TeamMate primarily adds value: supportive, generative, analytical, integrative, or executive.

E. Tools

TMDL provides base team collaboration commands plus role-specific extensions:

```
tools:
  commands:
    - name: "status"
      description: "Current work status summary"
    - name: "handoff"
      description: "Prepare work transfer"
    - name: "summarize"
      description: "Summarize for the team"
  options:
    - name: "lang"
      values: ["es", "en"]
      default: "es"
  decorators:
    - name: "brief"
      prompt: "Respond concisely"
```

F. Knowledge

Knowledge defines external documents to inject into the LLM context. In TMDL v2.0, this includes both domain expertise and operational context (project, team, timeline, organization):

```
knowledge:
  # Operational context
  - id: "project-context"
    name: "Project context"
    source: "knowledge/project.md"
    type: context
    inject: always
    priority: critical
  - id: "team-info"
    name: "Team information"
    source: "knowledge/team.md"
    type: context
    inject: always
    priority: high
  # Domain knowledge
  - id: "domain-tourism"
    name: "Tourism sector knowledge"
    source: "knowledge/tourism.md"
    type: domain
    inject: always
    priority: high
  - id: "methodology-swot"
    name: "SWOT methodology guide"
    source: "knowledge/swot-guide.md"
    type: methodology
    inject: on_demand
```

Knowledge types include context (operational: project, team, timeline), domain (expertise), methodology (processes), reference, guidelines, and examples. Injection modes control when content is included: always (permanent), on_demand (when relevant), or startup (conversation start only). This design allows updating project context without modifying the TeamMate definition.

VI. ILLUSTRATIVE EXAMPLE

To demonstrate TMDL in practice, we present a specification developed for academic project teams. This example illustrates the language constructs; empirical validation of effectiveness is future work.

A. Deployment Context

The specification was designed for undergraduate Tourism degree courses where students work in teams on semester-long market analysis projects. Each team would have access to AI TeamMates:

- **Ana:** Analyst role, focused on SWOT and market analysis
- **Marco:** Researcher role, focused on literature and sources

B. Implementation Approach

TeamMates are deployed on OpenWebUI with TMDL specifications converted to system prompts through: parsing YAML, resolving role inheritance, loading external knowledge files, and generating structured prompts. Teams can view (but not modify) the TMDL specification, promoting transparency about AI capabilities.

C. Example Specification

A simplified version of Ana’s specification:

```
1 tmdl_version: "2.0"
2 metadata:
3   id: ana-tourism-nt40
4   name: "Ana - Analyst"
5   version: "2.0"
6 identity:
7   display_name: "Ana"
8   personality: |
9     Meticulous analyst who always
10    asks for data before conclusions.
11   tone: professional
12 role:
13   extends: "roles/analyst.yaml"
14   expertise:
15     - domain: "Tourism Market"
16       level: expert
17 collaboration:
18   contribution_style: analytical
19   protocols:
20     on_task_assignment: |
21       1. Confirm understanding
22       2. List information needed
23       3. Propose structure
24       4. Request validation
25 knowledge:
26   - id: project-brief
27     source: "knowledge/project-brief.md"
28     inject: always
```

D. Design Rationale

The specification reflects TMDL’s theoretical foundations: `contribution_style: analytical` draws from team composition research on functional role diversity; and explicit `protocols` operationalize predictable behavior patterns derived from the teacher’s instructional methodology.

The `boundaries` section (in the full specification) makes AI limitations explicit, addressing research showing that transparent capability communication improves human-AI collaboration [3].

VII. DISCUSSION

A. Theoretical Contributions

TMDL makes several theoretical contributions to human-AI collaboration:

Contribution-centered framing. By grounding collaboration modeling in team effectiveness research, TMDL provides a theoretical foundation for AI TeamMate specification that aligns with how high-performing teams actually operate. The focus on contribution patterns—how TeamMates add value through support, idea generation, analysis, integration, and execution—reflects the cooperative nature of effective teamwork.

Operationalized organizational constructs. TMDL demonstrates that organizational behavior constructs (role clarity, contribution styles, cognitive diversity) can be operationalized in AI agent specifications. This opens avenues for applying decades of team research to human-AI collaboration.

Explicit research hypotheses. Rather than claiming empirical validation, TMDL articulates testable hypotheses derived from theory. This positions the language as a research platform for investigating human-AI teaming.

B. Contributions to the ADL Specification Family

The development of TMDL has served as a practical testbed for the ADL framework, revealing structural limitations that have informed the evolution toward ADL 2.0 [16]. This bidirectional relationship between domain-specific specialization and core framework refinement represents a methodological contribution to the design of assistant description languages.

During TMDL development, four key limitations of ADL 1.0 became apparent:

Core-profile separation. ADL 1.0 provided a unified structure but blurred the distinction between core assistant properties and domain-specific concerns. When constructing TMDL’s collaboration and role constructs, it became difficult to determine which elements belonged to the general assistant description versus teammate-specific requirements. This motivated ADL 2.0’s explicit separation between a minimal, domain-agnostic core and specialized profiles.

Inheritance mechanisms. Extending ADL 1.0 specifications required copying and modifying large portions of descriptions, increasing inconsistency risk. Creating families of related TeamMates (e.g., analysts with different domain expertise) exposed the need for declarative specialization. ADL 2.0 addresses this through explicit `extends` mechanisms enabling controlled variation without structural duplication.

Boundaries as first-class elements. In ADL 1.0, behavioral constraints were embedded implicitly within prompt text, making them difficult to validate or reuse. TMDL’s emphasis on explicit boundaries, `can_do`, `cannot_do`, and `defers_to` constructs demonstrated the importance of treating constraints as mandatory, structured elements. ADL 2.0 elevates boundary specification to a required core component.

Specification-execution separation. ADL 1.0 descriptions were sometimes implicitly tied to particular prompting strategies, limiting portability. TMDL’s goal of model-agnostic TeamMate definitions highlighted the need for stricter separation between declarative specification and execution semantics, now a design principle in ADL 2.0.

These contributions position TMDL not merely as an application of ADL, but as an active participant in the co-evolution of the ADL specification family. The lessons learned from teammate-oriented specification have directly shaped the core framework, demonstrating the value of domain-specific experimentation in language design research.

C. Practical Implications

For practitioners deploying AI in team settings, TMDL offers:

- A structured approach to defining AI TeamMate behavior
- Validation tools to catch specification errors before deployment
- A common vocabulary for discussing AI TeamMate configuration
- Templates for common roles (analyst, researcher, etc.)

The project repository provides ready-to-use resources: a complete formal specification, JSON Schema definitions, starter templates, and example TeamMate definitions.

D. Limitations and Assumptions

Several limitations should be acknowledged:

Transfer assumption. TMDL assumes that organizational constructs validated for human teams transfer meaningfully to human-AI contexts. While emerging research supports this assumption [14, 19, 9], full validation remains an empirical question.

Specification-behavior gap. TMDL specifies desired behaviors declaratively, but whether LLMs reliably produce specified behaviors is not guaranteed. Specifying `contribution_style: analytical` does not ensure the agent will behave analytically in all situations.

Unidirectional specification. TMDL specifications are defined by humans and consumed by AI; they do not adapt based on team feedback during collaboration. While `protocols` enable context-sensitive responses, the specification itself remains static. True “reciprocal partnership” [19] would require dynamic adaptation mechanisms not currently in TMDL’s scope.

Anthropomorphization considerations. Research suggests that positioning AI as a teammate increases user engagement but may also increase anthropomorphization [8]. TMDL addresses this tension through explicit `boundaries` sections that make AI limitations transparent, but the balance between beneficial teammate framing and unrealistic expectations requires careful attention in deployment.

Format choice. Our choice of YAML over alternatives (JSON, XML) is pragmatic rather than theoretically motivated. Different structured formats may produce similar benefits.

Lack of empirical validation. The hypotheses presented have not been empirically tested. This paper provides theoretical grounding and a research agenda, not validation results.

E. Future Work: Empirical Validation Agenda

The research hypotheses articulated in Section III constitute our empirical agenda:

- **Controlled experiments** comparing TMDL-specified TeamMates against unstructured prompt baselines on behavioral consistency and team integration metrics
- **Longitudinal studies** examining prompt drift resistance over extended deployments
- **Construct validation** testing whether different contribution styles produce measurably different agent behaviors
- **Team composition studies** examining effects of complementary versus homogeneous AI TeamMate configurations

Additional technical directions include multi-agent coordination extensions, dynamic adaptation based on team feedback, cross-platform tooling, and visual editors for non-technical authoring.

VIII. CONCLUSION

This paper introduced TMDL (TeamMate Description Language), a structured specification language for virtual TeamMates in hybrid human-AI teams. Building on ADL (Assistant Description Language) and the “machines as teammates” research agenda [19], TMDL addresses a fundamental design choice in human-AI collaboration: positioning AI as a configured *teammate* rather than a generic *tool*.

TMDL’s contribution-centered design asks “how does this TeamMate add value to the group?”—a framing grounded in team effectiveness research from Hackman, Salas, and Mathieu. This approach produces specifications aligned with how high-performing teams actually operate: through coordination, mutual support, and complementary contributions rather than competing interests.

The language’s taxonomies are rigorously grounded in organizational behavior research:

- **Contribution styles** (supportive, generative, analytical, integrative, executive) synthesize Benne and Sheats’ functional group roles into actionable categories for AI TeamMate configuration
- **Cognitive styles** (adaptor, innovator, balanced) operationalize Kirton’s Adaption-Innovation theory, enabling deliberate composition of complementary thinking orientations
- **Explicit boundaries** address research on role clarity and mental models, making AI capabilities and limitations transparent to human team members

We position TMDL as a theoretically-grounded proposal with explicit hypotheses for future validation. The six hypotheses comparing the teammate paradigm against tool-based AI use constitute our research agenda, to be tested through quasi-experimental designs in educational settings during the 2025–2026 academic year.

By making theoretical grounding explicit and separating descriptive characteristics (identity, role, collaboration patterns) from operational configurations (tools, context, knowledge), TMDL provides both a practical framework for practitioners and a research platform for investigating human-AI teaming.

TMDL is available as open source at <https://github.com/ppernias/tmdl> under CC-BY-4.0 license.

REFERENCES

- [1] Ishani Aggarwal and Anita Williams Woolley. “Team Creativity, Cognition, and Cognitive Style Diversity”. In: *Management Science* 65.4 (2019), pp. 1586–1599. DOI: 10.1287/mnsc.2017.3001.
- [2] Anthropic. *Prompt Engineering Guide: Use XML Tags to Structure Your Prompts*. Anthropic Documentation. Accessed: 2024-12-15. 2024. URL: <https://docs.anthropic.com/claude/docs/use-xml-tags>.
- [3] Gagan Bansal et al. “Beyond Accuracy: The Role of Mental Models in Human-AI Team Performance”. In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. Vol. 7. 2019, pp. 2–11.
- [4] Kenneth D. Benne and Paul Sheats. “Functional Roles of Group Members”. In: *Journal of Social Issues* 4.2 (1948), pp. 41–49. DOI: 10.1111/j.1540-4560.1948.tb01783.x.
- [5] Steven R. Bray and Lawrence R. Brawley. “Role Efficacy, Role Clarity, and Role Performance Effectiveness”. In: *Small Group Research* 33.2 (2002), pp. 233–253. DOI: 10.1177/104649640203300204.
- [6] Ashraf Elnashar, Jules White, and Douglas C. Schmidt. “Enhancing Structured Data Generation with GPT-4o: Evaluating Prompt Efficiency Across Prompt Styles”. In: *Frontiers in Artificial Intelligence* (2025). DOI: 10.3389/frai.2025.1558938.
- [7] Saibo Geng et al. “JSONSchemaBench: A Rigorous Benchmark of Structured Outputs for Language Models”. In: *arXiv preprint arXiv:2501.10868* (2025).
- [8] Ethan Goh et al. “From Tool to Teammate: Evaluating LLMs as Collaborative Clinical Partners”. In: *medRxiv preprint* (2025). DOI: 10.1101/2025.01.21.25320412.
- [9] Cleotilde Gonzalez, Pranav Gupta, and Anita Williams Woolley. “COHUMAIN: Building the Socio-Cognitive Architecture of Collective Human-Machine Intelligence”. In: *Topics in Cognitive Science* 16.4 (2024), pp. 733–756. DOI: 10.1111/tops.12673.
- [10] J. Richard Hackman. *Leading Teams: Setting the Stage for Great Performances*. Boston, MA: Harvard Business School Press, 2002.
- [11] Daniel R. Ilgen et al. “Teams in Organizations: From Input-Process-Output Models to IMOI Models”. In: *Annual Review of Psychology* 56 (2005), pp. 517–543. DOI: 10.1146/annurev.psych.56.091103.070250.
- [12] Omar Khattab et al. “DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines”. In: *arXiv preprint arXiv:2310.03714* (2023).
- [13] Michael J. Kirton. “Adaptors and Innovators: A Description and Measure”. In: *Journal of Applied Psychology* 61.5 (1976), pp. 622–629. DOI: 10.1037/0021-9010.61.5.622.
- [14] Bowen Lou et al. “Unraveling Human-AI Teaming: A Review and Outlook”. In: *arXiv preprint arXiv:2504.05755* (2025).
- [15] John E. Mathieu et al. “A Review and Integration of Team Composition Models: Moving Toward a Dynamic and Temporal Framework”. In: *Journal of Management* 40.1 (2014), pp. 130–160. DOI: 10.1177/0149206313503014.
- [16] Pedro A. Pernías Peco and M. Pilar Escobar Esteban. “ADL 2.0: A Core Specification Framework for LLM-based Assistants”. In: *arXiv preprint* (2025). Manuscript in preparation.
- [17] Pedro A. Pernías Peco et al. “ADL-Based Educational Assistant Architecture: Transforming Pedagogical Design into Automated Tutoring Systems”. In: *ICERI2025 Proceedings*. IATED. 2025, pp. 1751–1760.
- [18] Eduardo Salas, Dana E. Sims, and C. Shawn Burke. “Is There a ‘Big Five’ in Teamwork?” In: *Small Group Research* 36.5 (2005), pp. 555–599. DOI: 10.1177/1046496405277134.
- [19] Isabella Seeber et al. “Machines as Teammates: A Research Agenda on AI in Team Collaboration”. In: *Information & Management* 57.2 (2020), p. 103174. DOI: 10.1016/j.im.2019.103174.
- [20] Sunu Widiyanto, Hajar M. A. Sahib, and M. Farid W. Rahman. “Task Interdependence, Team Identity and Team Performance: A Bottom-Up Multilevel Model”. In: *SAGE Open* 14.1 (2024). DOI: 10.1177/21582440241237874.
- [21] Anita Williams Woolley et al. “Evidence for a Collective Intelligence Factor in the Performance of Human Groups”. In: *Science* 330.6004 (2010), pp. 686–688. DOI: 10.1126/science.1193147.