Clarification Before Execution: Interaction Design for Trustworthy Conversational Business Intelligence

Literature review and conceptual analysis

Abstract

Conversational business intelligence promises to let people explore organizational data without mastering query languages, but its central design problem is not translation. It is the management of ambiguity before an irreversible interpretation becomes an authoritative result or a production data artifact. This conceptual analysis examines clarification as a control mechanism across two levels of analytics work. SiriusBI places semantic completion, knowledge-guided clarification, and proactive querying inside a deployed BI architecture; SiriusDeliver extends agentic assistance into warehouse delivery, where ambiguity can affect workflow configuration, transformation code, and platform submission. Their relationship suggests that clarification must be treated as a lifecycle capability rather than a chat feature. Drawing on conversational text-to-SQL, interactive correction, schema linking, constrained decoding, retrieval, human-AI interaction, and documentation research, the article develops a framework for deciding when to ask, what to ask, and how to carry the answer into executable artifacts. It argues for minimum-regret questions, visible commitments, progressive authorization, and persistent decision records. The goal is not maximal dialogue. It is the smallest amount of well-timed interaction that prevents costly semantic and operational errors while preserving user agency.

The Hidden Contract in a Simple Question

"Show weekly active customers by region" looks like a complete request. A database professional immediately sees several missing contracts. Which event makes a customer active? Is a customer a person, an account, or a paying organization? Which region wins when billing, residence, and sales territories disagree? Does a week begin on Monday, follow a fiscal calendar, or use a local timezone? Should late-arriving events revise previous weeks? A language model can conceal these gaps by choosing defaults that yield valid SQL. Fluency then becomes dangerous: a polished chart can make an arbitrary interpretation appear settled.

Text-to-SQL benchmarks made schema generalization measurable. Spider requires systems to interpret questions over databases not seen during training, revealing the difficulty of mapping language to new relational structures (Yu et al., 2018). Schema-aware models such as RAT-SQL explicitly encode database relations and align mentions with columns (Wang et al., 2020). Intermediate representations such as SemQL reduce the mismatch between natural-language intent and low-level SQL details (Guo et al., 2019). Yet none of these mechanisms can recover a business definition that is absent from both the utterance and the available metadata. The system must either ask, abstain, or invent.

The interaction literature gives a better starting point than the metaphor of perfect translation. CoSQL models database use as a multi-turn exchange in which an expert may clarify ambiguous questions, communicate that a request is unanswerable, or explain results while maintaining dialogue state (Yu et al., 2019). Its construction recognizes that real inquiry changes as people see intermediate answers. Clarification is therefore not a repair applied after generation; it is a normal operation in analytic sensemaking.

SiriusBI: Clarification as Part of the Query Architecture

SiriusBI directly targets limitations of single-round SQL generation. Jiang et al. (2024) present a multi-module production system whose dialogue-and-query mechanism combines semantic completion, knowledge-guided clarification, and proactive querying. Semantic completion attempts to reconstruct omitted context. Knowledge-guided clarification draws on domain information to ask a better question. Proactive querying lets the system advance the task rather than waiting passively for perfectly specified input. These elements are paired with a data-conditioned selection between one-step fine-tuning and a two-step semantic-intermediate-representation approach.

This arrangement is significant because it locates clarification inside the generation pipeline. A question about "orders" might be completed from conversation history, checked against an approved metric catalog, and linked to relevant schema elements before the system decides whether a question is necessary. The question can then present real alternatives grounded in available data: "Do you mean submitted orders or fulfilled orders?" That is much more useful than asking the user to restate the request.

Interactive text-to-SQL research supports this strategy. Li et al. (2020) show that a parser-independent method can improve performance by asking users multiple-choice questions about uncertain interpretations. PHOTON similarly combines a parser, an executor, response generation, and a human-in-the-loop corrector that flags or repairs inputs the system cannot reliably map (Zeng et al., 2020). These studies suggest that a clarification interface should expose a compact decision, not transfer the whole burden of formal specification to the user.

SiriusBI also offers production evidence: its report describes use across finance, advertising, and cloud contexts, more than 93 percent SQL-generation accuracy, and reductions in analyst query time from minutes to seconds (Jiang et al., 2024). The finding is encouraging, but interaction quality cannot be inferred from aggregate SQL accuracy alone. A useful evaluation would distinguish correct silent completion, successful clarification, unnecessary interruption, misunderstood answers, and cases in which the system confidently avoided asking. Clarification has costs, and an agent that asks about every field is safe only in a narrow sense; it may be abandoned.

Choosing the Right Moment to Ask

A clarification policy should compare the expected regret of acting with the burden of asking. Regret grows with semantic uncertainty, action impact, and the difficulty of reversal. A low-cost exploratory query over a sampled dataset may justify a reasonable default that is clearly displayed. Publishing a dashboard metric to executives, modifying a shared table, or scheduling a recurring transformation demands a higher threshold.

Four uncertainty types deserve separate treatment. Lexical uncertainty concerns phrases such as "customer" or "conversion." Structural uncertainty concerns joins, filters, aggregation grain, and the intended output shape. Contextual uncertainty concerns time, organizational scope, permissions, and prior turns. Operational uncertainty concerns what should be executed, where, at what cost, and with which downstream effects. Models are often calibrated differently across these types, so a single confidence score is not an adequate policy.

The content of the question matters as much as its timing. A good question should be discriminative: each answer should materially change the proposed artifact. It should be grounded: options should come from cataloged definitions or detectable schema choices. It should be answerable by the current user: a marketing analyst may choose a campaign definition but not a warehouse partition policy. It should reveal consequence: the interface should say when a choice changes historical results, access, cost, or downstream tables.

General human-AI interaction guidance reinforces these requirements. Systems should make clear what they can do, show contextually relevant information, support efficient correction, and preserve user control (Amershi et al., 2019). In conversational BI, these principles imply a visible interpretation panel. Before

execution, the panel can summarize the selected metric definition, filters, grain, sources, and estimated scope. The user need not read SQL to confirm the system's commitments.

From Answered Question to Durable Commitment

Clarification fails if its answer remains only in chat history. When a user selects "fulfilled orders," that choice should bind the generated query and, if the work becomes reusable, become structured metadata. Retrieval-augmented generation offers a way to fetch current metric definitions and examples from an external memory rather than relying on model parameters (Lewis et al., 2020). But retrieval must be paired with persistence: the system records which definition version it used, and future runs can determine whether that version is still valid.

Documentation research offers useful analogies. Datasheets for datasets ask creators to state motivation, composition, collection processes, recommended uses, and limitations so that downstream consumers do not treat a dataset as context-free material (Gebru et al., 2021). Model cards similarly structure the reporting of intended use, evaluation conditions, and limitations (Mitchell et al., 2019). An analytics artifact needs a compact "decision card" containing the originating question, resolved terms, data sources, temporal assumptions, owner, validations, and known exclusions. This card is not a substitute for lineage or tests; it connects human intent to them.

Constrained decoding completes the picture. PICARD prevents inadmissible tokens from entering a SQL sequence by checking partial outputs during generation (Scholak et al., 2021). In enterprise BI, constraints can be semantic and organizational as well as grammatical. If the decision card specifies fulfilled orders, a policy checker can reject a query that uses an order-created timestamp without the required fulfillment status. The clarification answer thus becomes an executable assertion.

SiriusDeliver: When Clarification Crosses into Operations

SiriusDeliver expands the cost of ambiguity. Warehouse delivery includes retrieving context, configuring a workflow, generating code, submitting to a platform, and diagnosing failures. Xie et al. (2026) organize this work around a hierarchical delivery agent, artifact lifecycle control before and after execution, and trace-driven skill evolution. The deployment report covers six business teams and four task types, with 3,600 monthly active users and 18,240 sessions over two months. It reports 87.2 percent end-to-end success, 73.5 percent autonomous submission, and large reductions in median delivery time and engineer effort during an A/B test.

The system makes clear why conversational decisions need operational scope. Suppose an analyst asks SiriusBI for daily net revenue and later requests that the result be materialized. At query time, the system may safely run a temporary calculation. At delivery time, the same definition requires ownership, dependency scheduling, late-data policy, backfill behavior, service level objectives, and resource constraints. The transition should not silently promote exploratory defaults into production commitments.

A hierarchical agent can route questions to the right role. Business semantic questions return to the requester or metric owner. Source-quality questions go to a data steward. Scheduling and resource questions go to a platform owner or are resolved through policy. This is better than treating "human in the loop" as one undifferentiated approval button. The loop should involve the person with authority and knowledge for the specific decision.

Artifact lifecycle control also changes question timing. Before submission, the system can ask about an unresolved destructive overwrite or a missing service level. After execution, it may ask whether an unexpected row-count change represents valid business growth or a defect. The second question is grounded in observation rather than speculation. Progressive interaction therefore follows the artifact: specify, preview, validate, execute, observe, and revise.

A Practical Interaction Protocol

A trustworthy conversational analytics system can implement six stages without forcing users to experience six separate screens. First, it constructs an intent frame containing requested measure, dimensions, population, period, and output purpose. Second, it retrieves candidate definitions and schema evidence. Third, it enumerates material ambiguities and ranks them by expected regret. Fourth, it asks the minimum set of discriminative questions or declares explicit defaults for low-impact choices. Fifth, it generates an artifact plus a plain-language commitment summary. Sixth, it validates the artifact against the accepted decisions and requests authorization proportional to impact.

The protocol should support reversal. Users need to see which answer caused a filter or join, edit earlier decisions, and regenerate downstream artifacts deterministically. A conversational correction must not leave hidden fragments from the abandoned interpretation. Versioned decision records and artifact diffs can make the change legible.

It should also resist false precision. Schema linking can rank a column highly without establishing that its business meaning is correct. RAT-SQL demonstrates how relation-aware representations improve alignment (Wang et al., 2020), but an interface should distinguish model evidence from organizational approval. A label such as "matched to orders.region with high model confidence" is different from "approved regional reporting field." Both signals are useful; conflating them is not.

Measuring Clarification Quality

Evaluation must capture more than final execution. At the turn level, useful measures include ambiguity detection recall, question relevance, option coverage, answer incorporation, and the number of turns to a stable interpretation. At the task level, measures include semantic correctness, user effort, time to insight, downstream revision, and harmful action avoidance. At the organizational level, measures include definition reuse, metric consistency, incident frequency, and whether decision records improve audits.

Counterfactual evaluation is especially informative. Reviewers can compare the artifact produced with clarification against the artifact produced under the system's default. The difference estimates the value of the question. Logs can also reveal negative value: questions whose answers never change an artifact are candidates for removal. Care is required because logging analytic conversations may expose sensitive business questions and schema details.

User studies should sample expertise and authority. A novice may need explanations of terms; an expert may prefer direct editing. An owner may authorize publication; an explorer may only run a personal query. The same prompt and uncertainty therefore require different interaction policies. Human-AI guidance emphasizes supporting efficient correction and matching assistance to context (Amershi et al., 2019); role-aware evaluation tests whether that ideal survives real workflows.

Limits and Research Agenda

Clarification is not a cure for missing governance. If an organization has multiple undocumented revenue definitions, the agent cannot ask its way to a canonical answer. It can reveal the disagreement, route ownership, and preserve a local choice. Similarly, retrieval can amplify stale or politically convenient documentation. Decision records need owners, expiry conditions, and links to verifiable data contracts.

The connection between SiriusBI and SiriusDeliver raises several research questions. How should uncertainty propagate from a conversational query into a production workflow? Which decisions can be inherited, and which must be reauthorized at deployment? How should agents explain conflicts between a user's requested definition and a governed metric? Can trace-driven skill learning distinguish a useful local convention from a policy violation repeated in historical practice? These questions involve organizational legitimacy as well as predictive accuracy.

Conclusion

Trustworthy conversational BI is not achieved when a model produces SQL that runs. It is achieved when the system helps people form an explicit, defensible interpretation and carries that interpretation into every executable artifact. SiriusBI makes multi-round clarification a first-class element of industrial BI; SiriusDeliver shows that similar discipline is necessary as agents move from queries to production delivery. Together with research on interactive text-to-SQL, constraints, retrieval, documentation, and human-AI interaction, they support a simple principle: ask before the cost of ambiguity exceeds the cost of interaction.

The best system will not maximize either autonomy or conversation. It will know when a reversible default is enough, when a targeted question prevents semantic harm, and when authority must remain with a person. That calibrated behavior is the difference between an impressive interface and a dependable analytics institution.

References

Jiang, J., Xie, H., Yang, J., Shen, S., Wang, Z., Zheng, Y., ... & Jiang, J. (2024). Siriusbi: A comprehensive llm-powered solution for data analytics in business intelligence. *arXiv preprint arXiv:2411.06102*.

Xie, H., Zhou, X., Yang, J., Shen, S., Wang, Z., Zheng, Y., ... & Jiang, J. (2026). SiriusDeliver: Automating Data Warehouse Delivery at Tencent. *arXiv preprint arXiv:2608.09185*.

Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., et al. (2019). Guidelines for human-AI interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1-13). https://doi.org/10.1145/3290605.3300233

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daume III, H., & Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM, 64*(12), 86-92. https://doi.org/10.1145/3458723

Guo, J., Zhan, Z., Gao, Y., Xiao, Y., Lou, J. G., Liu, T., & Zhang, D. (2019). Towards complex text-to-SQL in cross-domain database with intermediate representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4524-4535). https://doi.org/10.18653/v1/P19-1444

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*.

Li, Y., Chen, B., Liu, Q., Gao, Y., Lou, J. G., Zhang, Y., & Zhang, D. (2020). What do you mean by that? A parser-independent interactive approach for enhancing text-to-SQL. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (pp. 8351-8361). https://doi.org/10.18653/v1/2020.emnlp-main.561

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., et al. (2019). Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (pp. 220-229). https://doi.org/10.1145/3287560.3287596

Scholak, T., Schucher, N., & Bahdanau, D. (2021). PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 9895-9901). https://doi.org/10.18653/v1/2021.emnlp-main.779

Wang, B., Shin, R., Liu, X., Polozov, O., & Richardson, M. (2020). RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7567-7578). https://doi.org/10.18653/v1/2020.acl-main.677

Yu, T., Zhang, R., Er, H., Li, S., Xue, E., Pang, B., et al. (2019). CoSQL: A conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 1962-1979). https://doi.org/10.18653/v1/D19-1204

Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., et al. (2018). Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 3911-3921). https://doi.org/10.18653/v1/D18-1425

Zeng, J., Lin, X. V., Hoi, S. C. H., Socher, R., Xiong, C., Lyu, M., & King, I. (2020). Photon: A robust cross-domain text-to-SQL system. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 204-214). https://aclanthology.org/2020.acl-demos.24/