

A Process for Competence Specification in Computing Education

Edeyson A. Gomes¹, Lais N. Salvador¹, Marcos V. B. Cerqueira¹

¹Instituto de Computação

PGCOMP - Programa de Pós-graduação em Ciência da Computação

Universidade Federal da Bahia (UFBA)

Salvador, BA – Brazil

{edeysongomes, laisns, cerqueira.marcos}@ufba.br

Abstract. *Competency-Based Education requires competence specifications capable of relating knowledge, skills, dispositions, and expected performance across diverse educational contexts. However, deriving coherent and reusable competence formulations from educational tasks remains a challenging activity in Computing Education. This paper presents the Competency Specification Process (CSP), an iterative approach for eliciting, reviewing, refining, and reusing competence specifications derived from educational tasks. The evaluation involved successive applications in Problem-Based Learning scenarios from Theory of Computation and complementary introductory Computing Education contexts, including integrated technical high-school education. The results revealed recurrent challenges involving knowledge granularity, competence overlap, action-verb selection, and Bloom-aligned cognitive expectations. Across successive cycles, CSP supported clearer competence boundaries, improved specification consistency, and increased reuse across related tasks. These findings suggest that competence specification benefits from iterative review and refinement activities and can be supported through a structured process that promotes stability, reuse, and curricular coherence across instructional contexts.*

1. Introduction

Competency-Based Education (CBE) has gained increasing relevance in Computing Education due to the growing demand for curricula capable of connecting conceptual understanding, problem solving, and professional practice in authentic learning contexts [Clear et al. 2020b, Force 2020]. Within this perspective, competences are expected to express not only what learners know, but also what they can perform in context-dependent educational situations [Paquette et al. 2021, Rezgui and Mhiri Sellami 2020].

Curricular initiatives such as Computing Curricula 2020 (CC2020) reinforce this movement by emphasizing the integration of knowledge, skills, and dispositions in Computing curricula [Force 2020]. The revised Bloom's Taxonomy proposed by Anderson and Krathwohl [Anderson et al. 2001] is also widely used to characterize expected cognitive performance and support instructional planning and assessment. Together, these perspectives increase the need for competence specifications capable of making expected learner performance explicit and interpretable across educational contexts.

Despite these advances, operationalizing competence-oriented curricula remains challenging in practice. Computing curricula have traditionally been organized around

bodies of knowledge and content coverage, whereas competence-oriented approaches require the integration of knowledge, skills, dispositions, contextualized tasks, and observable evidence into coherent instructional situations [Clear et al. 2020a, Frezza et al. 2018].

Competence development is often supported by specification practices that remain largely informal and dependent on natural-language descriptions, instructor judgement, and local curricular conventions. Consequently, ambiguities frequently emerge regarding terminology, knowledge granularity, cognitive expectations, and assessment interpretation [Calhau et al. 2021, Paquette et al. 2021, Clear et al. 2020a]. Although initiatives such as CC2020 and CoLeaF provide more structured conceptual foundations, translating educational tasks into consistent competence descriptions remains difficult, particularly in task-centered environments, including project-based and problem-based learning scenarios, where expected performance must remain aligned with learner actions, assessment criteria, and available evidence [Giaffredo et al. 2017]. Studies in Computing Education have reported recurring challenges involving overlapping descriptions, unstable knowledge granularity, inconsistent action-verb selection, and divergent interpretations of Bloom-aligned cognitive expectations. Such issues can hinder curriculum organization, competence reuse, instructional alignment, and the consistent interpretation of learner performance across educational contexts.

Previous research has proposed competency ontologies, metadata models, competence frameworks, and structured approaches for representing competence-related information [Miranda et al. 2016, Paquette et al. 2021, Rezgui and Mhiri Sellami 2020]. However, these efforts have focused primarily on representational concerns, offering comparatively limited guidance on how these activities can be conducted and consolidated across successive task-centered instructional cycles.

Because competences are inherently contextual and task-oriented, their specification requires more than selecting content elements or action verbs. It requires explicit procedures capable of relating learner actions, knowledge requirements, contextual constraints, and expected evidence into coherent and reusable specifications.

To address this need, this work proposes the Competency Specification Process (CSP), an iterative approach for deriving competence specifications from educational tasks and progressively improving them through review, refinement, and reuse activities. Grounded in competence-oriented curricular principles from CC2020 [Force 2020], CSP integrates task analysis, Bloom-aligned cognitive interpretation, collaborative review, controlled vocabularies, and cross-task refinement to support the systematic development, consolidation, and reuse of competence specifications across successive instructional cycles. Within the collaborative review process, controlled vocabularies were used to promote terminological consistency across competence specifications, reducing variations in competence descriptions and supporting coherent refinement and reuse across successive instructional cycles.

In contrast to representational approaches, CSP focuses on the activities through which competence specifications are derived, reviewed, and consolidated over time. Rather than addressing how competence-related information should be represented, classified, or organized, it emphasizes how competence specifications are constructed from authentic educational tasks and progressively refined through collaborative analysis and

review. Accordingly, competence specification is treated as an iterative process aimed at improving clarity, consistency, reuse potential, and curricular coherence across instructional contexts.

CSP was evaluated through successive applications in Problem-Based Learning (PBL) scenarios from Theory of Computation and complementary introductory Computing Education contexts, including cross-level applications in integrated technical high-school education. These successive applications allowed the examination of how competence specifications evolve through review, reuse, and consolidation across different instructional settings.

This study adopts a qualitative longitudinal research design based on successive applications of the CSP across multiple instructional contexts in Computing Education. Rather than comparing independent cases, the study investigates how competence specifications evolve through repeated applications of the same process over time. The analysis focuses on specification activities grounded in authentic educational tasks and collaborative review practices.

The research investigates the following questions:

- RQ1: How can competences be systematically elicited and structured from educational tasks in Computing Education?
- RQ2: How do iterative review and refinement activities contribute to the consistency, clarity, and reuse of competence specifications?

This paper contributes and evaluates an iterative process for competence elicitation, review, refinement, and reuse in task-centered Computing Education contexts. The following sections present the CSP, describe the evaluation design, report the results, discuss their implications, and outline limitations and future research directions.

2. Related Work

Competence-oriented approaches have become increasingly relevant in Computing Education as curricular discussions move beyond content coverage and isolated learning outcomes. Recent curricular initiatives, particularly CC2020 and the Competency Learning Framework (CoLeaF), conceptualize competence as the integration of knowledge, skills, dispositions, and contextual task demands, emphasizing performance in authentic educational and professional situations [Clear et al. 2020a, Force 2020, Clear et al. 2020b, Frezza et al. 2018]. These frameworks provide conceptual foundations for competence-based curriculum design by defining competence structures, constituent elements, and relationships between learning expectations and professional practice.

These frameworks distinguish natural-language competence statements from more structured formulations involving Knowledge–Skill relations, contextual descriptors, and Bloom-aligned cognitive expectations. Such structures contribute to greater consistency in competence specification by making expected performance and constituent elements more explicit. Nevertheless, studies associated with CC2020 and CoLeaF report persistent challenges related to knowledge granularity, contextual variation, disciplinary interpretation, and the need for collaborative refinement during competence formulation activities [Frezza et al. 2018, Clear et al. 2020a]. These observations suggest that competence specification involves not only defining competence structures, but also managing interpretation and consistency across different educational contexts.

Despite advances in competence-oriented Computing Education, establishing consistent competence specifications remains challenging because competence itself is interpreted differently across educational, professional, and disciplinary contexts [Westera 2001, Pikkarainen 2014, Mulder 2012, Stoof et al. 2002]. These variations affect how competence boundaries, constituent elements, and expected performance are defined and communicated, particularly when specifications must remain meaningful to multiple stakeholders [Frezza et al. 2018]. To reduce such ambiguities, several studies have proposed competency models, ontologies, metadata structures, controlled vocabularies, and other mechanisms for organizing and formalizing competence-related information [Miranda et al. 2016, Doderio et al. 2007, Rezgui and Mhiri Sellami 2020]. Together, these approaches contribute to greater consistency, interoperability, and curricular alignment by providing shared representations of competence concepts and their relationships.

This challenge becomes particularly evident in task-centered instructional approaches. Project-Based Learning and Problem-Based Learning can support competence-oriented education because they engage learners in authentic tasks, artifact production, situated performance, and increasing levels of autonomy [Giaffredo et al. 2017]. However, the relationship between tasks and competence development is not automatic. Target competences must be explicitly identified, assessment criteria must remain aligned with expected performance, and evidence of learning must be interpreted consistently across instructional contexts. Without such alignment, task-centered activities may emphasize task completion rather than competence development.

Taken together, existing research in Computing Education provides substantial support for competence modelling, curricular alignment, representational structures, and task-centered instructional design. However, comparatively less attention has been dedicated to the processes through which competence specifications are systematically derived, consolidated, and reused across successive educational tasks and instructional contexts. This gap motivates the present study, which investigates competence specification as an iterative activity aimed at promoting consistency, reuse, and curricular coherence through task analysis, collaborative review, and progressive refinement.

3. Method

This study adopts a qualitative longitudinal research design based on successive applications of the CSP across multiple instructional contexts in Computing Education. Rather than comparing independent cases, it investigates how competence specifications evolve through repeated applications of the same process over time. The longitudinal design supports the observation of recurring specification patterns across different educational tasks and educational levels, enabling the analysis of competence elicitation, refinement, consolidation, and reuse from authentic instructional activities. The study does not evaluate instructional effectiveness or student learning outcomes; instead, it focuses on the process through which competence specifications are constructed, collaboratively reviewed, and progressively refined.

The empirical investigation comprised three successive CSP application cycles representing progressive stages in the maturation of the process. These cycles involved Theory of Computation and introductory Computing Education scenarios, including five

initial PBL tasks (Task01–Task05), four validation scenarios (Task201–Task204), six advanced PBL tasks (Task25.00–Task25.05), and ten integrated technical high-school Computing Education tasks aligned with the Brazilian National Common Curricular Base (BNCC) Computing Complement. Across these successive application cycles, competence specifications, review records, refinement decisions, and longitudinal observations supported the identification of recurring specification challenges, refinement patterns, and review outcomes.

The CSP applications involved collaborators with expertise in Computing Education, competency-based education, PBL, Theory of Computation, Educational Informatics, and integrated technical high-school education. The university-level applications involved seven collaborators acting as competence modelers, reviewers, and domain specialists, whereas the high-school applications involved five reviewers, with additional instructors contributing authentic educational tasks. Review activities followed a structured expert-review procedure based on predefined analytical criteria, including competence granularity, conceptual overlap, instructional coherence, Bloom-aligned cognitive expectations, terminology consistency, contextual dependence, and reuse potential. Bloom alignment followed the revised taxonomy of Anderson and Krathwohl [Anderson et al. 2001], considering the intended cognitive process within the instructional context rather than relying exclusively on action verbs. Divergent interpretations were resolved through collaborative discussion leading to documented consensus rather than voting or independent agreement measures.

The resulting analyses informed decisions concerning reformulation, decomposition, specialization, consolidation, reuse, and exclusion of competence specifications. All review outcomes were documented in CSP reports and revisited across the application cycles, providing traceability for the refinement process. Collaborators participated exclusively in the construction and review of competence specifications; they were not investigated as research participants, their performance was not evaluated, and no personal data were collected.

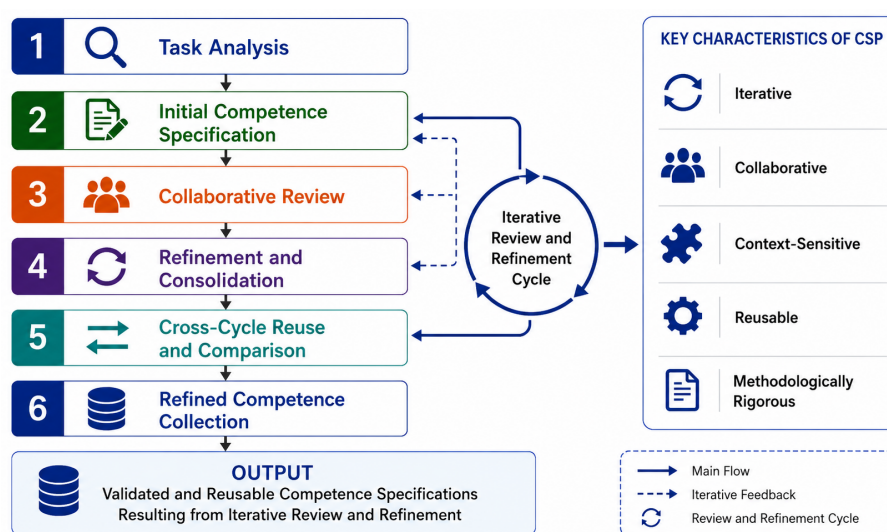


Figure 1. Iterative structure of the Competency Specification Process (CSP), illustrating task analysis, collaborative review, refinement, and cross-cycle consolidation across successive instructional cycles.

Figure 1 illustrates the iterative structure of the CSP, while Table 1 summarizes the specification activities applied across the investigated cycles.

Table 1. Operational procedure used to apply the CSP across the study

CSP Stage	Input	Main Action	Decision Criteria	Output
Task Analysis	Educational task	Identify knowledge requirements, learner actions, contextual constraints, and expected evidence	Sufficient information to infer expected learner capabilities	Structured task analysis
Initial Specification	Task analysis	Formulate preliminary competence specifications linking knowledge, skills, dispositions, context, and evidence	Represents a coherent learner capability	Initial specification
Collaborative Review	Initial specification	Assess clarity, granularity, Bloom alignment, overlap, and contextual dependence	Coherent, non-redundant, and aligned with task demands	Review recommendations
Refinement and Consolidation	Review recommendations	Revise, decompose, specialize, consolidate, reuse, or exclude specifications	Clearer boundaries, reduced ambiguity, and improved reuse potential	Refined specification
Cross-Cycle Comparison	Refined specifications	Compare specifications across tasks and cycles to identify overlap, specialization, and reuse opportunities	Equivalent specifications consolidated; contextual variations specialized when needed	Consolidated competence collection

The data analyzed comprised instructional tasks, competence specifications, review records, refinement decisions, cross-cycle comparison tables, and alignment artifacts produced throughout the CSP application cycles. These materials were interpreted to identify recurring specification challenges, refinement patterns, and review outcomes.

To support methodological transparency, traceability, and replicability, the supplementary materials have been made publicly available in a GitHub repository containing representative educational tasks, CSP reports, review reports, adjustment records, and supporting artifacts documenting the successive refinement of competence specifications throughout the study. The CSP repository is available at: <https://github.com/eduknowgpt/OCSP>.

Methodological trustworthiness was supported through data-source triangulation involving instructional tasks, competence specifications, review records, refinement decisions, and cross-cycle comparison artifacts. The longitudinal application of CSP enabled the identification of recurring refinement patterns, while documented review decisions provided traceability for the resulting outcomes.

The study is limited to competence specification and refinement activities in Computing Education and should not be interpreted as evidence of learning effectiveness or pedagogical superiority. Instead, it evaluates CSP as a structured process for competence elicitation, refinement, and reuse across diverse instructional scenarios.

4. Evaluation and Results

The evaluation investigated how CSP supported the systematic derivation, refinement, and reuse of competence specifications across successive Computing Education scenarios. Rather than assessing instructional effectiveness or learner performance, the study focused on the specification process itself, examining how iterative applications contributed to the stabilization, consolidation, and reuse of competence structures over time.

To investigate this process, CSP was applied across multiple PBL scenarios in

Theory of Computation and complementary introductory Computing Education tasks. Figure 2 summarizes these longitudinal applications and illustrates the progression from initial competence elicitation toward increasing levels of reuse, consolidation, and selective introduction of new competence specifications.

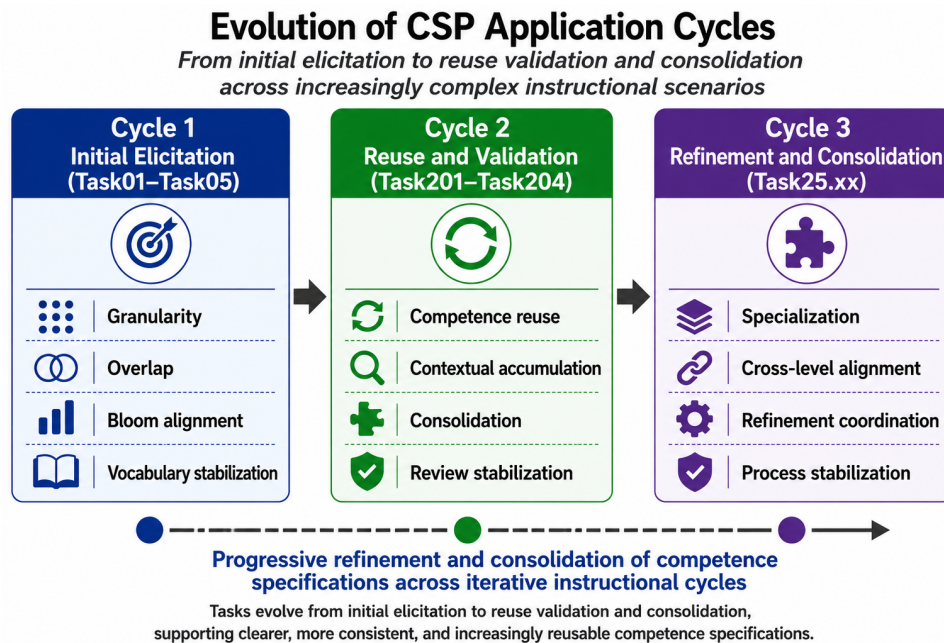


Figure 2. Longitudinal evolution of CSP application cycles.

Tables 2 and 3 summarize the evolution of the CSP across the investigated cycles. The former presents the primary objectives, representative activities, and observed outcomes of each cycle, whereas the latter provides complementary quantitative evidence derived from the resulting competence specification artifacts.

Table 2. Evolution of CSP objectives and outcomes across application cycles

Cycle	Primary Goal	Representative Activities	Main Outcome
Task01–05	Initial competence elicitation and structuring	Competence elicitation, boundary definition, merging, removal, initial reuse, and specialization	Establishment of stable and reusable competence specifications
Task201–204	Competence consolidation and reuse	Competence reuse, activation refinement, controlled specialization, and role clarification	Consolidation of reuse practices and activation structures
Task25.xx	Selective competence extension	Cross-level validation, semantic gap analysis, task reengineering, and selective competence creation	Introduction of new specifications driven by explicit instructional and representational needs while preserving reuse

The reported indicators distinguish between competences that participated in task specifications and those first introduced during a given cycle. This distinction highlights the increasing role of reuse across successive applications, as later cycles relied more heavily on previously consolidated competence specifications than on the creation of new

ones. Reuse events represent explicit instances in which specifications established in earlier tasks were subsequently adopted in new instructional contexts.

Table 3. Quantitative summary of CSP application cycles

Cycle	Tasks	Competences Used	Introduced	Reuse Events
Task01–05	5	16	16	5
Task201–204	4	15	2	5
Task25.xx (Theory of Computation)	5	18	6	10

Note. **Competences Used** denotes the number of distinct competences that participated in the task specifications of a given cycle. **Introduced** denotes competence definitions first created during that cycle. **Reuse Events** denote explicit instances in which competences defined or consolidated in previous tasks were reused in subsequent specifications. These categories are not additive because a competence may participate in multiple tasks and may be reused multiple times. The High School Computing cycle is analyzed separately because cross-task consolidation and reuse analysis are still in progress.

The quantitative results reinforce the longitudinal patterns observed across the CSP applications. The first cycle produced 16 competence definitions and established an initial set of reusable specifications. The second cycle introduced only two additional competences while maintaining five reuse events, indicating increased reliance on previously consolidated specifications. The third cycle exhibited the highest level of reuse (10 events) while introducing six new competences, suggesting that new specifications were created selectively when existing ones did not adequately represent emerging instructional requirements. Across the university-level cycles, the number of competences participating in task specifications remained relatively stable (16, 15, and 18), whereas reuse increased substantially, indicating progressive consolidation of the competence base over time.

Together, these patterns provide initial evidence that competences can be systematically derived from educational tasks and progressively improved through review and reuse activities. The following subsections examine the three university-level cycles in greater detail, illustrating how competence specifications were elicited, refined, reused, and consolidated through successive CSP applications.

4.1. Cycle 1: Initial Competence Elicitation and Structuring

The first CSP cycle (Task01–Task05) focused on deriving an initial set of competence specifications from PBL scenarios in Theory of Computation. At this stage, the primary challenge was to identify reusable learner capabilities while distinguishing them from task-specific requirements and contextual details.

The specification and review reports revealed recurrent issues related to overlap, knowledge granularity, terminological inconsistency, and divergent interpretations of expected cognitive performance. To address these issues, review activities focused on clarifying specification boundaries, refining knowledge requirements, reducing contextual dependencies, and improving the consistency of Bloom-aligned cognitive expectations. These refinements supported the identification of reusable specifications across tasks and led to mergers, removals, reuse relationships, and specialization decisions, including the consolidation of overlapping regular-expression competences in Task01 and the specialization of Turing Machine-related competences in Task05.

By the end of the cycle, 16 competence definitions, five reuse events, and four specialization decisions had been established (Table 3). These findings indicate that the principal contribution of the first cycle was the establishment of stable specification boundaries and reusable competence formulations, providing the foundation for subsequent CSP applications and directly addressing RQ1 regarding the systematic derivation and structuring of competences from educational tasks.

A representative refinement occurred in Task01, where the original competence *Develop Problem Solutions Using Automata* was replaced by *Develop Problem-Solving Solutions Using Finite State Machines*. The change improved semantic precision and instructional alignment, illustrating how CSP promotes refinement through competence reuse rather than through new competence creation.

4.2. Cycle 2: Competence Consolidation and Reuse

The second cycle (Task201–Task204) shifted the emphasis from initial competence creation toward the consolidation and reuse of previously established specifications. Most task specifications reused competences derived during the first cycle, while review activities focused on refining activation conditions, clarifying competence roles, and introducing controlled specializations when existing formulations became too broad for new instructional contexts. Representative outcomes included the specialization of C13, refinement of activation boundaries for Turing Machine-related competences, and the introduction of activation roles such as core, supporting, extension, and transversal competences.

The cycle introduced only two new competence definitions while recording five reuse events and one specialization decision (Table 3). These results indicate that iterative review activities increasingly supported the adaptation and reuse of existing specifications rather than the creation of new ones. The principal contribution of the cycle was therefore the consolidation of reuse practices and activation structures across tasks, providing evidence for RQ2 by illustrating how review and refinement activities can improve the consistency and reuse potential of competence specifications.

A representative example involved the specialization of C13, whose scope was refined to distinguish general grammar interpretation from task-specific applications. The refinement preserved the existing specification while clarifying its role and relationships to other competences, illustrating how CSP supports incremental refinement and reuse without unnecessary competence proliferation.

4.3. Cycle 3: Selective Competence Extension

The third cycle (Task25.00–05) evaluated CSP under more complex instructional conditions involving task reengineering, cross-level review, semantic coverage analysis, and increasing conceptual dependencies across Theory of Computation topics. Unlike the previous cycle, the objective was not only to reuse existing specifications but also to assess whether they provided adequate coverage for emerging instructional demands. While several tasks were specified entirely through competence reuse, others revealed semantic gaps that motivated the introduction of additional competences when existing specifications could not adequately represent the required learner capabilities.

The cycle involved 18 competences and generated 10 reuse events, representing the highest level of reuse observed among the university-level applications. At the same

time, six new competences were introduced to address instructional requirements that could not be satisfactorily represented by previously established specifications. These findings indicate that CSP supported not only the consolidation and reuse of existing competences, but also the selective extension of the competence base when explicit representational needs emerged. Together, these results provide additional evidence for RQ2 by demonstrating how iterative review and refinement activities can support both specification reuse and the controlled introduction of new competences.

A representative example emerged in the Musical Collections task, where existing specifications captured the required capabilities individually but not their combined application. This gap motivated the introduction of a new competence specification that integrated previously separate capabilities while preserving reuse of existing elements, illustrating how CSP supports selective competence extension when existing specifications no longer provide sufficient semantic coverage.

4.4. Cross-Level Application and Curricular Alignment

A complementary evaluation applied CSP to introductory Computing Education tasks situated in integrated technical high-school contexts and aligned with the BNCC Computing Complement. Unlike the university-level scenarios, which focused primarily on Theory of Computation, these tasks covered a broader range of topics, including programming, symbolic reasoning, data processing, computer architecture, and computational decision-making. This cycle examined whether CSP could be applied consistently across different educational levels and curricular domains.

The 10 analyzed tasks produced 50 competence specifications and established alignments with 30 distinct BNCC Computing references. Despite the diversity of topics and instructional goals, the resulting specifications exhibited a consistent structure combining core and supporting competences, explicit knowledge requirements, dispositions, Bloom-aligned cognitive expectations, and curricular alignments. The application required no substantive modifications to the core CSP activities or decision criteria, suggesting methodological consistency across educational levels and curricular domains.

5. Discussion and Threats to Validity

The evaluation suggests that competence specification in Computing Education is not merely a curriculum documentation activity, but a process of progressive interpretation and refinement. Across the analyzed scenarios, competence formulation required relating instructional goals, learner actions, contextual constraints, and anticipated evidence of performance into coherent specifications. These observations are consistent with prior discussions in the competence literature, which emphasize the contextual and interpretive nature of competence and argue that competence descriptions cannot be reduced to isolated knowledge elements, action verbs, or learning outcomes alone. Instead, their meaning depends on how expected performance is situated within specific instructional contexts.

A central observation concerns the role of refinement during competence specification. Initial formulations frequently combined reusable learner capabilities with task-specific characteristics, making specification boundaries difficult to identify and limiting reuse across instructional scenarios. Iterative review activities therefore played an important role in distinguishing enduring capabilities from contextual requirements and in

clarifying the scope of individual specifications. The observed refinements suggest that systematic review procedures contribute to greater boundary stability, reduced representational ambiguity, and increased reuse potential across related instructional contexts.

The results also reinforce the importance of contextual interpretation during Bloom-aligned competence formulation. Several CSP applications indicated that action verbs alone were insufficient to determine the expected cognitive level of performance. Tasks that initially appeared to involve procedural execution often required comparison, justification, abstraction, or evaluation when considered in relation to their instructional goals and expected evidence. Consequently, Bloom alignment emerged not as a simple verb-classification exercise, but as an interpretive activity grounded in task demands, contextual constraints, and anticipated performance. This observation is particularly relevant for competence-oriented curriculum design, where cognitive expectations must remain consistent across learning, instruction, and assessment activities.

Another implication concerns competence reuse. The analyzed application cycles suggest that reuse rarely occurs through direct replication of competence formulations. Instead, it becomes feasible when specification boundaries are sufficiently stable to allow the same underlying capability to be activated across different instructional contexts. From this perspective, reuse appears to depend less on the number of available specifications than on their maturity and conceptual stability. The observed transition from competence creation to competence reuse across the CSP cycles reinforces this interpretation. As specification boundaries became progressively clearer through review, specialization, and consolidation activities, new tasks could increasingly be represented through adaptation and reuse rather than through the creation of additional specifications.

The cross-level applications provide complementary evidence for this interpretation. Similar specification challenges emerged across university and high-school contexts despite substantial differences in curricular scope and domain coverage. At the same time, the core CSP activities remained applicable without major structural modifications, suggesting that the underlying difficulties of competence specification are associated less with particular Computing topics than with the interpretive and context-dependent nature of competence-oriented educational design.

This study presents several limitations. The evaluation focused on competence specification and refinement activities rather than on student learning outcomes, instructional effectiveness, or assessment performance. Consequently, the findings should not be interpreted as evidence that CSP improves learning outcomes or pedagogical effectiveness.

The investigated scenarios were primarily derived from Theory of Computation tasks, which provided opportunities to observe competence refinement and reuse across successive cycles. Although the complementary high-school applications introduced variation in educational level and curricular scope, additional studies involving other Computing sub-disciplines, instructional approaches, and institutional contexts are necessary before broader generalizations can be established.

Another limitation concerns the interpretive nature of competence specification. Despite the use of iterative review activities and collaborative refinement, alternative reviewers might reasonably arrive at different specification or refinement decisions when

analyzing the same instructional task. While the review process supported convergence and reduced inconsistencies, some degree of interpretive subjectivity remains inherent to competence-oriented educational modelling.

The study also emphasized methodological refinement and instructional coherence rather than automation, scalability, or large-scale curricular deployment. Likewise, the objective was not to establish correctness through independent ratings, but to examine how competence specifications evolved through successive cycles of analysis, review, and refinement.

Consistent with the qualitative longitudinal research design adopted in this study, the goal was to identify recurring specification patterns and refinement mechanisms across successive CSP application cycles through analytical replication rather than to achieve statistical generalization to a broader population of educational contexts. Future work should therefore investigate the application of CSP in additional Computing Education domains, larger curricular ecosystems, longitudinal educational settings, and integration scenarios involving competence management, curriculum planning, and learning analytics environments.

Taken together, the findings suggest that competences can be systematically derived from educational tasks through iterative analysis and review (RQ1), while successive refinement activities contribute to clearer competence boundaries and increased reuse across instructional contexts (RQ2). The recurrence of similar refinement patterns across multiple cycles and educational settings strengthens the plausibility of these observations.

Despite these limitations, the study suggests that competence specification benefits from being treated as an iterative activity of analysis, review, refinement, and consolidation rather than as a one-time curriculum description task. CSP contributes a structured process that supports competence stabilization, specification reuse, and curricular coherence while accommodating the evolving demands of different instructional contexts.

6. Conclusion

This work presented the Competency Specification Process (CSP), an iterative process for eliciting, reviewing, refining, and reusing competence specifications derived from educational tasks in Computing Education. Across successive application cycles, CSP showed that competences can be systematically derived from educational tasks while supporting clearer specification boundaries, progressive consolidation, and increased reuse. These findings suggest that competence specification benefits from being treated as a continuous process of analysis, review, refinement, and consolidation rather than as a one-time curriculum description activity.

By integrating task analysis, Bloom-aligned cognitive interpretation, collaborative review, controlled vocabularies, and cross-task refinement, CSP provides methodological support for competence specification in task-centered Computing Education. Although evaluated in Theory of Computation and introductory Computing Education contexts, the findings provide initial evidence of applicability across different educational levels. Future work will investigate additional Computing Education domains, larger curricular ecosystems, and integration with competence management, curriculum planning, learning analytics, and Learning Management Systems (LMS).

Declaration on the Use of Artificial Intelligence

Grammarly and ChatGPT were used solely as writing-support tools to assist with English language revision and improve textual clarity. These tools were not used to generate research questions, collect or analyze data, develop the proposed process, interpret findings, or formulate conclusions. All scientific contributions, methodological decisions, results, and interpretations presented in this paper are the sole responsibility of the authors.

References

- Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., Raths, J., and Wittrock, M. C. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Pearson, New York, NY, complete edition. Revised edition of: *Taxonomy of educational objectives* / Benjamin S. Bloom, 1956.
- Calhau, R. F., Azevedo, C. L. B., and Almeida, J. P. A. (2021). Towards ontology-based competence modeling in enterprise architecture. In *2021 IEEE 25th International Enterprise Distributed Object Computing Conference (EDOC)*, pages 71–81.
- Clear, A., Clear, T., Impagliazzo, J., and Wang, P. (2020a). From knowledge-based to competency-based computing education: Future directions. In *2020 IEEE Frontiers in Education Conference (FIE)*, pages 1–7.
- Clear, A., Clear, T., Vichare, A., Charles, T., Frezza, S., Gutica, M., Lunt, B., Maiorana, F., Pears, A., Pitt, F., Riedesel, C., and Szykiewicz, J. (2020b). Designing computer science competency statements: A process and curriculum model for the 21st century. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education, ITiCSE-WGR '20*, page 211–246, New York, NY, USA. Association for Computing Machinery.
- Dodero, J. M., Sánchez-Alonso, S., and Frosch-Wilke, D. (2007). Generative instructional engineering of competence development programmes. *Journal of Universal Computer Science*, 13:1213–1233.
- Force, C. T. (2020). *Computing Curricula 2020: Paradigms for Global Computing Education*. Association for Computing Machinery, New York, NY, USA.
- Frezza, S., Daniels, M., Pears, A., Cajander, A., Kann, V., Kapoor, A., McDermott, R., Peters, A.-K., Sabin, M., and Wallace, C. (2018). Modelling competencies for computing education beyond 2020: a research based approach to defining competencies in the computing disciplines. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE 2018 Companion*, page 148–174, New York, NY, USA. Association for Computing Machinery.
- Giaffredo, S., Mich, L., and Ronchetti, M. (2017). From the project-based learning method towards the competence-based approach to education. In *Proceedings of the 6th Computer Science Education Research Conference, CSERC '17*, page 56–65, New York, NY, USA. Association for Computing Machinery.
- Miranda, S., Orciuoli, F., Loia, V., and Sampson, D. (2016). An ontology-based model for competence management. *Data & Knowledge Engineering*, 107.

- Mulder, M. (2012). Competence-based education and training. *The Journal of Agricultural Education and Extension*, 18.
- Paquette, G., Marino, O., and Bejaoui, R. (2021). A new competency ontology for learning environments personalization.
- Pikkarainen, E. (2014). Competence as a key concept of educational theory: A semiotic point of view. *Journal of Philosophy of Education*, 48.
- Rezgui, K. and Mhiri Sellami, H. (2020). Towards a semantic framework for lifelong integrated competency management and development. *The Computer Journal*, 63:1004–1016.
- Stoof, A., Martens, R., Van Merriënboer, J. J. G., and Bastiaens, T. (2002). The boundary approach of competence: A constructivist aid for understanding and using the concept of competence. *Human Resource Development Review*, 1:345–365.
- Westera, W. (2001). Competences in education: A confusion of tongues. *Journal of Curriculum Studies*, 33(1):75–88.