講演番号	363

Interactive Extraction of 1D-CAE Modeling Requirements and Determination of Model Configuration using Generative AI

Kyohei Naito [1)] Junichi Ichimura [1) 2)] Yuki Wakimoto [1)]

1) NewtonWorks Co.
VPO Kyobashi Bld. 1-16-10 Kyobashi, Chuo, Tokyo, 104-0031, Japan
2) The Open University of Japan
2-11 Wakaba, Mihama, Chiba, Chiba, 261-8586, Japan

KEY WORDS: vehicle development, generative AI, 1D-CAE, Model Specification, LLM, Human-in-the-Loop [B2]

The quality of 1D-CAE models depends on their upstream specification, yet specification work is often skipped due to time constraints, leading to over- or under-detailed models and late-stage rework. In software engineering, the shift from ad-hoc AI code generation to specification-driven development (SDD) is well recognized. The same structural change is emerging in 1D-CAE, where AI-based Modelica code generation is becoming feasible, making clear specifications the essential starting point.

This paper proposes a six-step process for 1D-CAE model specification through AI dialogue (Table 1), referencing ISO/IEC/IEEE 15288 and implemented on Claude Code (Anthropic, Claude Opus 4.6). Each step proceeds through a four-element cycle: AI proposal, user correction, consistency checking via sub-agents, and critical review for structural gaps. This cycle is applied at every step.

Table 1 Six Steps of the Specification Process

Step	Name	Key Question
1	Problem & Requirements Definition	What should the model clarify?
2	Use Case Definition	Who inputs what, what is output, and how are decisions made?
3	I/O Definition	Where are the model boundaries and external interfaces?
4	Functional Decomposition	What physical transformations connect inputs to outputs?
5	Architecture Definition	Which components handle which energy domains?
6	Implementation Spec & Granularity	At what granularity should each component be modeled?

The process was applied to two domains. In Case 1 (battery cell thermal model), the initial input "I want to simulate battery temperature during charge/discharge" was developed through 12 dialogue versions into a specification of 11 components plus 3 parameter maps. Fig. 1 shows the Step 4 functional block diagram produced through this process. The diagram captures five functional blocks: charge control, electrical model, heat generation (Joule + reaction heat), thermal energy balance, and parameter lookup, along with their feedback loops linking the electrical, thermal, and control domains.

A key design decision visible in Fig. 1 is the bidirectional coupling between charge control and the electrical model. At Step 2 (use case definition), the AI initially treated CV-phase current as an externally prescribed input, like the CC-phase current. The user verified this: during CV charging, the upper voltage limit [V] is held constant while the current must be dynamically computed from the cell's internal state (OCV, internal resistance, SOC, and temperature). This required the cell model to expose an acausal electrical port (voltage–current pair) and solve simultaneously with the charge controller (represented by the bidirectional arrow in Fig. 1). This correction, made at Step 2, propagated through Steps 3–6, reshaping the I/O definition, architecture, and port specifications. Other user-driven refinements included redirecting the scope toward rapid-charging thermal management and BEV integration, while critical review detected the absence of discharge thermal behavior and prompted the addition of an emergency-stop state.

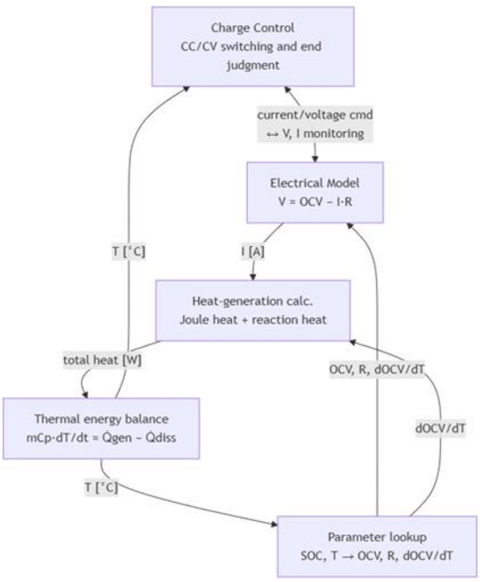

Fig.1: Functional block diagram of the battery cell thermal model (Step 4 output). Labels translated into English by the author from the original Japanese output.

In Case 2 (gasoline vehicle fuel economy), the same process was applied to verify domain generality. The initial input "I want to evaluate WLTC mode fuel economy" was specified into 12 components with 8 maps through 9 versions, confirming that the six-step framework is domain-independent. Across both cases, a natural role division emerged: users verified domain-specific physical validity, while AI targeted structural completeness. Both were completed within a few hours, significantly reducing the time for systematic specification.