

Gates Instead of Prompts: Development and Methods of a Large-Language-Model Expert System for Root- Cause Analysis in a Steel Foundry

Miroslav Holčák¹

miroslav.holcak@gmail.com

Independent Researcher

Research Article

Keywords: large language models, root-cause analysis, steel castings, solidification simulation, green sand, Industry 5.0

Posted Date: October 14th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11125630/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: Competing interest reported. The author is an independent consultant and offers services related to the methodology described in this paper.

Abstract

Large language models (LLMs) can read foundry documentation, simulation output and machine records, but their answers sound equally confident whether right or wrong. This paper describes the development and methods of an offline expert system for root-cause analysis of casting defects and equipment failures in a steel foundry with a green-sand molding line. Between March and September 2026 it went from a single-model prototype, through a digital twin that was deliberately torn down, to a tool-using agent. At its core are deterministic decision gates: checks in code that stop the run or flag output without the required evidence, such as a material anchor, verbatim citations and an image-reference lock. A separate judge, kept apart from the expert roles, must try to refute the expert opinions before writing its report. A further rule reads simulation fields only at the place, moment and threshold of defect formation, and every claim about equipment carries one of three trust tiers. Analysis depth is measured by a deterministic scorecard against a reference analysis: on the reference casting, five task mandates, with no new tools, raised mentions of the actual defect location from zero to 16, 11 and 3 in three runs, while quantification stayed below the reference. Plant results are reported as classes of findings. The system works offline on historical data and is not in trial or routine production use. The recommended ceiling of autonomy is that the system proposes and a human signs.

Introduction

Root-cause analysis (RCA) of a casting defect or an equipment failure draws on many sources at once: drawings and process sheets, solidification and filling simulations, laboratory results, the manuals of the molding line and the pouring unit, and months of machine records. In many foundries this material exists, but nobody reads all of it for a single defect. Large language models (LLMs) can read it quickly. The difficulty is that their answers sound just as confident when they are wrong as when they are right.

The casting literature confirms both the promise and the risk. An empirical study in this journal gave questions from a foundry-engineer qualification examination to eleven general-purpose models; the best exceeded 85% accuracy, yet the authors judged their competence closer to a novice than to an experienced practitioner and called for careful expert oversight¹. A 2026 review of LLMs for foundry intelligence names heterogeneous data, weak traceability across process stages and shop-floor constraints as the main barriers, and recommends validation with a human in the loop². The Editor of this journal has asked authors to document any use of LLMs in their work³.

The plant described here provided an example. In May 2026, a general-purpose chatbot was asked why the pouring unit kept interrupting its cycle, with values from the operator screen attached. The model assumed that aluminum was being poured, declared the actual temperatures of more than 1,300°C physically impossible, and recommended replacing the temperature sensor and downgrading the temperature alarm to a warning that no longer stops the cycle. The same answer also contained sensible observations. That was the dangerous part: the error and the sound advice were written with equal

confidence. A system with access to the equipment manual corrected the answer the same day, citing a specific page.

The European Commission's concept of Industry 5.0 places the human at the center of industrial systems, with technology extending rather than replacing expert judgment^{4,5}. In a foundry this has a concrete meaning. The analysis of a defect is a judgment for which a process engineer, a metallurgist or a maintenance manager is accountable to the customer. A model may collect evidence, read hundreds of pages and point out connections, but the signature stays with a person. The project summarized this as a rule: the system proposes, a human signs.

This paper describes the development and methods of Digital Foundry PRO, an expert system for RCA of casting defects and equipment failures. The author conceived the system and developed it on his own initiative between March and September 2026, outside any assigned task or company project, and tested it on historical data of one steel foundry. The system is not a production control system. It works offline on exported historical data, has no connection to the manufacturing execution system (MES) or to the line's control systems, runs on one workstation and has one user. Time savings were not measured and are not reported. Plant results are given as classes of findings, without the company's operating figures.

The contribution is methodological. It consists of (i) a set of deterministic decision gates, checks in code, that stop or flag unsupported LLM output; (ii) a rule for reading solidification-simulation fields only at the place, moment and threshold at which a defect forms; (iii) three trust tiers for claims about equipment, with only the tier verified on the plant's own machine allowed into operator procedures; (iv) a deterministic scorecard which indicated that changing the task, rather than adding tools, deepened the analysis; and (v) two documented negative results, a digital twin that was deliberately torn down and a sensor-mapping error that looked like a plant finding. None of the individual techniques is new. The claim is limited to their combination for foundry RCA and to the fact that the key checks are enforced by code rather than by instructions to a model.

The Czechoslovak foundry society, founded in 1923, set out to unite all foundry professionals, regardless of education, in joint work for the advancement of the industry⁶. In that spirit, the paper describes methods that another foundry could rebuild. The following sections review related work, describe the setting and the system, present the methods, report results, and discuss limitations and the recommended ceiling of autonomy.

Related Work

LLMs and Machine Learning in Casting

Published LLM applications in casting are mostly reported as pilots and feasibility studies rather than as systems in routine production use. Besides the benchmark and the review cited above^{1,2}, a multimodal assistant tuned with foundry knowledge combined an open model with retrieval over books, articles, user

manuals and metallographic images, including sand-related problems; the extended abstract gives no evaluation details⁷. Machine-learning studies in this journal include defect prediction from process data: combining domain knowledge with production data⁸, supervised prediction of porosity in low-pressure die-cast wheels⁹, and prediction of defects in magnesium high-pressure die casting from die sensor data¹⁰. A broader review of intelligent casting covers these data-driven directions¹¹. These predictive approaches need many labeled cycles. The RCA problem addressed here is different: explaining individual defects and failures from records, simulations and documents in a plant where scrap is not recorded per mold and each defect family has few examples.

Knowledge Graphs and LLM Cause Analysis

Knowledge graphs are a common way to ground cause analysis. A two-branch reasoning network over a quality knowledge graph has been used for root causes of continuous-casting defects¹². CausalKGPT combines an LLM with an industrial causal knowledge graph for quality problems in aerospace manufacturing¹³, and a knowledge-graph-enhanced LLM pipeline supports fault diagnosis and maintenance decisions in an Industry 5.0 setting¹⁴. Root causes of product defects have also been learned from historical production data with ensemble Bayesian networks¹⁵. The system described here grounds the model differently: in page-level documentation, simulation fields read under explicit rules, and deterministic tools. Its causal graph is only a prototype.

LLM Agents and Root-Cause Analysis

Tool-using agents that interleave reasoning and actions were introduced with ReAct¹⁶. In IT operations, an LLM-based RCA system with incident handlers has run in production¹⁷. Debate between several model instances improves factuality and reasoning¹⁸, and a multi-agent framework for collaborative RCA in quality assurance uses specialized agents with an orchestrator¹⁹. Hybrid agentic systems for smart manufacturing have been validated on historical industrial datasets rather than live data²⁰, which is also the situation here. The closest methodological analog is an agentic RCA framework for process industry that keeps a table of competing hypotheses with supporting and refuting evidence, accepts normal operation as a valid diagnosis and restricts what the agent may read according to the operating regime; without that restriction its accuracy dropped markedly²¹. Commercial tools have appeared as well, including an agent for defect detection and RCA over computed tomography, machine vision and service records that escalates findings to a quality engineer with evidence²², and agents for industrial automation²³.

Retrieval, Citations and Evaluation

Retrieval-augmented generation grounds model answers in retrieved documents²⁴. Lexical ranking uses the Okapi BM25 function from the probabilistic relevance framework²⁵. Models with long contexts overlook information placed in the middle²⁶, and generated citations often fail to support the statements they accompany^{27,28}. Even commercial retrieval-based legal research tools were found to hallucinate in 17% to 33% of cases²⁹. For evaluation, LLM judges show position, verbosity and self-enhancement biases^{30,31}, single runs of a stochastic model need error bars³², and a metric that becomes a target loses its value³³. Preregistration of hypotheses and analysis plans protects against adjusting hypotheses to results³⁴.

Position of This Work

Deterministic control layers, citation checks, the right of a model to return no finding, separation of assessors from a judge and evaluation against a pre-written reference each have known precedents. What is described here is their combination in one tool for RCA of castings, with the key checks enforced by code. A search of the literature available up to September 2026 did not find two further elements in this form: restricting the reading of simulation fields to the place, moment and threshold of defect formation, and trust tiers based on whether a claim has been verified on the specific machine. The search covered publicly available literature and press releases in English and Czech, not patents or internal company systems, and the sentence describes what that search found, not a claim of priority.

Setting, Data and System Overview

Plant and Scope

The system was developed and tested on data from a foundry producing steel castings on an automatic green-sand molding line with a sand plant (cooler, mixer and automatic sand tester) and an automatic pouring unit. Castings are simulated with commercial casting-simulation software before production. The company, its suppliers, the equipment types and the individual castings are not named, and no operating figures of the plant are given. Numbers in this paper describe the system itself or counts in the text of its outputs, and were recomputed from the stored source files.

Data

Process data reach the system as weekly exports from the plant's information systems: sand batch records, analog records of the sand plant and molding line, mold records and the pouring-unit log. Laboratory data, manual temperature measurements, simulation output (protocol, setup sheet, images and videos) and the documentation of the equipment are added by hand. An inventory of the molding line's control-system signal table, taken in late May 2026, showed that the overwhelming majority of signals were status bits; measured analog quantities made up about 2% of the signals, and the moisture

and temperature of the molding sand were not in the table at all. The analysis therefore rests on historical exports. After consolidation, the system holds tens of millions of rows of machine records. The model never reads them directly (see “Task Triage”).

Application and Architecture

The application has five layers (Fig. 1): a user interface; fourteen domain modules, thirteen of which correspond to tabs and one of which is disabled and kept as a fallback; a layer of language models and deterministic checks; a documentation library with the plant data; and data intake. The tabs cover a technological passport that reconciles the drawing, acceptance conditions and process sheet; an audit of the passport against standards; RCA of defects; interpretation of simulation videos; a causal graph; service diagnostics; validation of simulation against radiography and metallography; zoning (experimental); a management summary; a shift inspector for incidents; knowledge management; and an overview. The thirteenth tab, for the digital twin, is intentionally empty after its interface was removed (see “Negative Results”).

A principle links most of the methods: the model proposes and the program decides. The model may propose the direction of a correction, but an envelope of limits decides whether it is admissible. It may formulate reasoning, but a script computes the numbers. It may cite, but the reference list is built by code only from quotations found verbatim in the source. The date and version of every report are written by the program; the model once repeated an outdated date in nine consecutive reports.

Operating Status

Today the system is a forensic tool for historical data. It has no MES connection, does not use live data, runs on one workstation for one user without role management, and receives data days to weeks after events. Writing to the control system exists only as a simulated dispatcher (a half-second wait and a log line); its interface was removed with the digital-twin interface, and the project contains no industrial communication library. The LLMs are called through the vendors’ developer interfaces. At analysis time only the extract needed for a given query is sent to the model; documentation pages were sent once to the cloud vision model to create the library cards. The library is a searchable index, not a model trained on company data.

Methods

Development Process and Use of LLMs

LLMs are both the object of this study and tools of its development, so their use is documented here. Commercial models of two vendors, referred to here as vendor A and vendor B, were used through the vendors’ cloud interfaces between March and September 2026; trade names are not given. The application calls a large reasoning model of vendor A for reasoning, and a smaller, faster model of the same family for structured extraction and for converting documentation pages into text cards. An

experimental branch allowed the reasoning stage of the RCA tab to be switched to a model of vendor B, text only; the program records in each result which model and version produced it. The agent prototype described below uses the reasoning model of vendor A. The agents of the digital twin, removed in May, called models of vendor B, and the rejected vector-search trial used an embedding model of vendor A. The versions of the models used in the architect and executor roles during development are not recorded in the project documentation. Model names and versions are available from the author on request. The gates described below are enforced by code outside the model; the depth measurements in the Results were obtained with the specific models described here and may not carry over to other models.

The concept and design of the system are the author's. The software was written by two LLMs in separate roles under the author's direction (Fig. 2). An architect model (vendor A) proposed technical solutions, raised objections and issued written directives with acceptance criteria. An executor model (vendor B) implemented them, wrote tests and kept decision records. The models never communicated directly; the author relayed directives and objections as files, and the author's decision overrode both. The executor had a formal right to object and changed several proposals by doing so.

Every major decision has a written record; 87 were written during development. Since late May 2026 each record has had to contain a failure prediction written before implementation, in the manner of a project premortem³⁵, and a field for a later check against reality; many earlier records lack the prediction, and the reality check has been filled in for about a quarter of the records. Instructions for the executor grew after incidents rather than in advance: six of the current ten rules cite the date and incident that produced them. The arrangement has known weaknesses. The architect did not read the repository or run tests and worked from the executor's summaries, and agreement between the two models was not confirmation: once, an identifier introduced by one model was repeated by the other as a fact, although it appeared in no source.

Task Triage

Every task was first sorted into one of three groups, using the Cynefin framework loosely³⁶. Routine tasks were programmed directly. Raw data were never given to a model: machine records were first reduced deterministically to a summary of at most 50 rows (counts, means and variances before and after an event, anomalies), and simulation results reached the model only as selected images and frames. Complex tasks started as isolated prototypes with a written directive. For each complex design the architect had to answer three questions: can the problem be solved by physics, mathematics or an algorithm without a model; where is the model likely to cut corners, invent content or lose context; and, imagining the system has failed in the foundry a year from now, why did it fail? Each answer had to contain a falsifiable claim. Following the first question, the casting modulus $M = V/A$ and the adequacy of a feeder are computed by code from Chvorinov's rule, $t_f = B \cdot M^n$ with $n \approx 2^{37}$, not estimated by the model.

Two-Stage Reasoning and Extraction

A single model call proved unable to reason deeply about defect physics and fill in a strict output structure at the same time. The work is therefore split. The reasoning model writes free text; the extraction model, at zero temperature and with its internal reasoning switched off, converts the text into a structure defined by a data schema, and the program validates it. If validation fails, the extraction gets one correction attempt with the list of errors; if it fails again, the program accepts the unvalidated data and records a warning. Truncated structured output was a recurring failure: an empty result caused by a truncated response could be mistaken for a result reporting no defect. In the experimental branch the program reads the stop reason of each response and raises a visible error on truncation; in the default branch, a rescue parser keeps the complete part of the data and logs the error.

Expert Panel and Refutation-First Judge

The RCA of a defect starts with four narrowly defined expert roles: solidification, metal flow and filling, mold and sand, and stresses and cracking (Fig. 3). Each role has its own instructions, its own retrieval from the library and its own model call. All roles work under the same rules. Each may find no defect; its instructions state that inventing marginal problems in a sound casting is a fatal failure that harms production more than admitting uncertainty. Each must name at least one alternative hypothesis it considered and rejected on the data. Each physical claim must cite a specific page of a source or be marked as an expert estimate, and quotations from literature must be verbatim, in the original language, so that the program can find them in the source text. The solidification role must compute modulus ratios. Conditional prohibitions are inserted into the instructions: for a casting without cores, the mold-and-sand role may not reason about core permeability, and the material from the specification may not be replaced.

A separate model call acts as a judge in the sense used in the LLM literature, with one difference: before it writes anything, it must try to refute the expert opinions. That order is what the term refutation-first judge denotes here. The judge, with its own model call at zero temperature, receives the four opinions. It first acts as an inquisitor: it names contradictions between the opinions and places where they ignore the geometry of the casting, and writes rejected hypotheses, with reasons, into a mandatory section. Only then does it write the report, from what survived. It may not average opinions. Its first version did exactly that and gave a secondary factor, such as turbulent filling, the same weight as an isolated hot spot. It may not pass claims that a gate flagged as unverified, may not estimate dimensions from images, and orders corrective actions by causal distance from the root cause. The report has five parts: morphology of the defect, the point in filling or solidification where it formed, the root cause including whether the defect reaches into the machining allowance or the functional body, an action plan, and a reference list compiled by the program.

Merging the experts and the judge into one call would make each analysis faster and cheaper, and it is the first change anyone would propose. The application therefore refuses to start if the expert instructions contain verdict instructions or if the separate calls disappear, and the same tests run on every commit.

Deterministic Decision Gates

The central lesson of the project is that a rule whose observance depends on the model remembering it is not a rule. Whatever must always hold was rewritten into checks in code (Fig. 4). The checks have three levels of enforcement. A STOP ends the run. A flag lets the run continue but attaches a mark that the judge must consider and the user sees. Brakes in tests and at start-up protect the architecture itself from being weakened “temporarily”.

The STOP points are few. Without a material anchored verbatim in the input documents, an RCA in the application does not start. If data requested from the line cannot be retrieved, the analysis stops rather than continuing without them. The application does not start if the expert and judge roles are merged. An agent that calls the same tool with the same arguments three times in a row is stopped; this guard was added after an agent tried twenty times in a row to call a tool that was advertised to it but not registered. Everything else is flagged: a quotation not found verbatim in the card the model actually received; a reference to a document that does not exist in the project; a request for a simulation frame at a time outside the simulated process; visual evidence whose type does not match the claim, such as a filling frame for a solidification claim; and a reference to an image that fails the image-reference lock described below. Every report ends with a deterministic list of inputs it did not use. For one casting, more than seven hundred images were available, mostly frames from simulation videos, and the conclusion cited two; the system counted this itself and listed the rest as a risk.

Gates are reactive. They catch fabricated facts, not fabricated solutions, and they do not create depth of reasoning. Flags are read by the judge, which is again a language model and can argue them away; this happened when the judge explained a “fabricated evidence” flag as a typing error. A rule followed from these limits: a rule that cannot be enforced is not introduced.

Page-Level Documentation Library and Verbatim Citations

The plant documentation runs to thousands of pages, often in foreign languages. Each page of a PDF document is rendered once as an image, and a vision model writes a structured card from it: topics and machines (foreign terms also given in the original language), technical data, the physical process, proposed solutions and maintenance, and a description of the figures on the page (Fig. 5). Each card carries the source name and page number, and the page scan is stored beside it. Documents from word processors are split deterministically by headings, tables are converted row by row and never split between cards, and for a few selected books the page cards are reassembled into chapters according to the table of contents. Electrical schematics are read in two passes, first building a map of cross-references between terminals and control-program addresses, because a signal whose address lay on another sheet had been recorded as a terminal. Drawings are sent whole, only downscaled, since cutting a large drawing into tiles once severed the connections between its elements. The library holds more than 1,000 documents in about 14,000 cards, mostly equipment manuals.

Retrieval uses BM25²⁵, computed separately for each part of the library (manuals, procedures, standards, literature), interleaved by priority, with at most two cards from one book. Every hit is returned with the page before and after it, because tables and procedures run across page breaks. The query is cleaned of drawing numbers and codes, and a Czech query is extended with technical terms in the language of the manual.

The model receives only the retrieved cards, never whole documents, partly because long contexts are poorly used²⁶. Each quotation in a report is then searched for in the text of the cards the model actually received. The check compares text only. An unmatched quotation is flagged and the judge replaces the claim built on it with an expert estimate. A book without a single verified quotation disappears from the reference list, which the program compiles only from verified quotations, with the source identified from the card where the quotation was actually found. In the audit tab, documents uploaded for the case (for example, a deviation approved by the customer) take precedence over library cards, following an order of precedence among the documents themselves (customer drawing, acceptance conditions, internal process sheet), and the judge that settles disputes between the normative and the physical assessor sees only the titles and headers of cards, so that it can check that a cited card exists but cannot introduce new arguments. A test plants a unique marker in the card body and checks that it never reaches the judge. The precedence of case documents, by contrast, is only an instruction; the decision record itself estimated that the model would follow it in 70% to 90% of cases.

Reading Simulation Fields at Place, Moment and Threshold

A solidification and filling simulation produces dozens of fields at every moment. A model allowed to search for anomalies anywhere always finds some and attaches the defect to one of them. The reading was therefore reversed (Fig. 6). First, the defects that the casting actually has are listed from inspection, using the plant's defect classification; defect names in this paper follow the International Atlas of Casting Defects³⁸. For each defect, the condition of formation is defined as the intersection of the place where it forms, the moment when the mechanism acts, and the threshold above which a field becomes significant. Fields are read only inside this intersection, every claim is tied to a specific frame or stated to be without direct evidence, anomalies outside the intersection are treated as noise, and defects that the simulation does not model, such as fins, cracked cores or sand inclusions, are given no "evidence" from the fields.

Metal velocity is an example. The model searched for high velocity everywhere and found the highest values in the pouring basin, the sprue and after the filter. There, however, the running system is closed, full and pressurized, and high velocity is intended. Velocity starts to cause harm at the ingates and at the front of the stream entering the mold cavity, where oxide films can be entrained; for steel, a critical velocity of the order of 0.5 m/s is reported³⁹. Claims about velocity are therefore anchored at the ingates and the cavity entrance.

The standard porosity export of the simulation shows only regions above a threshold of a few percent. Microshrinkage below this threshold, even in functional zones, is invisible, and a final image of the body can look clean while the casting has microporosity; the system once judged such an image to be a sound casting. Porosity is therefore never judged from a single image. It is compared with the liquid fraction during solidification (isolated pools of last liquid show the limits of feeding), with the Niyama criterion $Ny = G/\sqrt{T^{40}}$, whose thresholds depend on alloy and conditions⁴¹, and with the map of hot spots and moduli. Thresholds cut both ways: a low threshold drowns a defect in the body in the coarse shrinkage of the sacrificial feeder, and a high threshold hides sub-threshold microporosity.

Simulation videos carry no machine-readable time, but every frame carries a burned-in stamp with the phase, process time and phase percentage. The program reads the stamp in all frames of a video (about 130) by optical character recognition, rejects outliers and builds a time map; if too many frames are unreadable, no map is written. The model then requests a frame at a process time, not “the middle of the video”, which matters because video time is not linear in process time. The time maps are used by the agent prototype; their use in the production tab is not yet complete. Videos that animate the position of a section plane instead of time were correctly refused.

When a model refers to a simulation image, the program resolves the reference only if every number in the request also appears in the name of the file found. A typing error passes; an invented solidification percentage or revision does not. This lock was introduced after the export module had silently substituted the most similar existing image for a non-existent one. Instead of a substitution, a conspicuous warning now appears in the interface and in the printed report.

Plant Data: Contracts, Profiling and Pitfalls

Data suppliers do not keep to a fixed format. The data contract for the line begins with the sentence “The data will arrive, but the format will not be kept.” It has two parts: a reference schema of fields (names, types and units) and a processing contract that turns each delivery into a canonical form (header, time, identifier continuity, removal of duplicates) and marks missing parts instead of inventing them. A separate contract covers deliveries from simulation, and each report records the quality of its input data. A read-only profiling tool produces a “data passport” of every export, with a file hash taken before and after the run.

The exports contained traps that the model did not recognize. A value with a decimal point instead of a comma was silently truncated to an integer, and the model then reasoned correctly about a wrong number. A zero often meant “sensor off” rather than a physical zero, so zeros are treated column by column. A counter of metal age in the control system did not reset between ladles and overflowed; an apparent finding that castings with inclusions had been poured with much older metal disappeared once impossible values were removed. Seven time formats are unified by a deterministic converter. One database of manual temperature measurements was shifted by one hour by daylight saving time, which an audit of the data revealed from the missing night hours. A first analysis of an interruption of pouring

placed it in the period when the molds were made, whereas the pouring log showed it about an hour later: the time a mold is made is not the time it is poured. A rising current of the sand-cooler motor, taken as evidence of a clogged exhaust, turned out to be an average that included standstills; the current while running had been constant for two years. Aggregation must follow the machine state, not the calendar.

Trust Tiers for Equipment Claims

The most expensive lesson was physical. A procedure for a pattern change on the pouring unit cited the manual page by page, but one step was the system's own recommendation: to align an axis offset in manual mode. That task belongs to a setter, and position protection is limited in manual mode. During a trial the axis traveled beyond its working range, and the procedure was withdrawn and prohibited. A comparison of the manual with years of machine records then showed several functions described in the manual that had never once been triggered on this machine. The manual describes the whole product range of the production line, with all the options the line manufacturer offers; it does not describe the specific equipment installed on the shop floor. A car owner's manual also covers all versions and modifications, but every function that depends on the version carries a note that it applies only if the vehicle has it. Manuals from manufacturers of production equipment often lack such notes. Every claim about equipment therefore carries a tier. T1: in the documentation and confirmed on this machine by operating data, a test or the supplier. T2: in the documentation, but dependent on the variant or not verified. T3: a synthesis or recommendation of the model, always with a way to verify it. Only T1 may enter an operator procedure; a model recommendation never becomes an instruction without a test or the signature of a setter or the supplier.

Tool-Using Agent and Task Mandates

In June, a month of corrections to the one-shot RCA improved citations but not its core, and a retrospective found a common root: an expert role received all inputs at once and had to write an essay in a single call, without being able to fetch a fact, read it and build on it. In all examined cases, the decisive evidence had been delivered and the role had passed over it. The expert role was therefore rebuilt as an agent that must obtain facts through tools (Fig. 7), following the ReAct pattern¹⁶. It exists as an isolated prototype of one role, without a user interface.

The agent has nine tools. They read the hard facts of the part from the simulation setup sheet and protocol (material, pouring temperature, mass, yield, number of castings and cores), images of the part geometry, individual simulation fields with an optional threshold, and a frame at a given time of filling or solidification; compute the modulus by Chvorinov's rule, feeder adequacy and the ratio of solidification times of two locations; search the literature; and attempt to read cooling curves, which the simulation deliveries do not yet contain, so that tool always returns "not found". One tool is registered but still missing from the catalog given to the model. Every tool returns either a result or an explicit "not found" with a reason, never silent emptiness. The number of cores has three states, present, absent or not recognized, because "unknown" is not zero.

After the run, the program compares the report with the trace of tool calls: at least four calls are required; a report that names the defect location must be preceded by reading the geometry, and a report that mentions the modulus by reading a field or computing it. A run halted by a guard is a violation, as is a report that ignores the cores of a part that has them. The check works with keywords, not individual claims, and violations are currently only printed.

After a fix to the parser of its responses, the first runs with the production model called tools correctly and passed all checks, yet their conclusions were shallow: three runs produced three different textbook conclusions, one confirming a hypothesis the reference analysis had refuted. Five mandatory instructions were then added to the task, with no new tool:

1. Examine the surroundings of the cores: modulus, Niyama criterion and porosity around the core cavities, not only the central hot spot under the feeder.
2. Separate the central hot spot from the ring zones around the cores and decide which zone is the real defect.
3. Read porosity above a stated threshold.
4. Use literature only as verbatim quotation from a retrieved card.
5. Make a serious attempt to refute the naive hypothesis, such as an undersized feeder neck, against the geometry of this casting.

Measuring Analysis Depth

The reference is an analysis of one steel casting with cores. It was produced by the author working with an LLM of vendor B in an interactive session with tools; the model investigated the defect and obtained its own evidence. It is not the report of an independent human expert.

A scorecard, a short program without a language model, counts five things in the text of a report: whether the argument uses the geometry of the part; the number of values read with a physical unit and of image references; the number of verbatim quotations; the presence of sections that admit uncertainty (verdict, confidence, what would refute the conclusion, timeline); and the number of mentions of the actual defect location. Rerunning it on the stored reports gives the same numbers. Mentions of the defect location are counted as occurrences of the Czech word stem for “ring” in the report; values are numbers followed by one of the units mm, cm, K, °C, s, %, kN, N·m or MJ; verbatim quotations are counted by their citation markers; and the uncertainty axis checks for six section headings. The scorecard measures a floor, not a ceiling: counts say nothing about the correctness of reasoning, which a person judges. It counts negated mentions too, and its uncertainty axis in practice measures the presence of headings, which every agent report has because the task requires them. A deterministic counter was preferred to an LLM judge, whose biases are well documented^{30,31}; the project had also seen an evaluating agent award full marks to a text whose gaps it had itself listed. The risk that a metric turned into a target stops measuring³³ remains.

For analyses that must withstand review by an expert opponent, a protocol of ten checkpoints in fixed order was written: an inventory of all sources, preregistration of defects, fields, hypotheses, thresholds and statistical plan³⁴, numbers only from scripts, a causal model with confounders, credibility of each simulation field, a physical arbiter (metallography, fractography or radiography), weighting of evidence with a discount for mutual dependence, an independent adversarial audit, a reproducible package with at least two runs, and a definition of done. The protocol is enforced by the discipline of the development model and the author, with helper scripts; no program enforces it yet.

Results

The most valuable plant findings did not come from clicking through the tabs of the application. Most were prepared by the development model working directly with the author on the data, under the same rules as the application: gates, trust tiers, a mandatory inventory of sources and reading at place, moment and threshold. The archive of outputs of the application itself covers one reference casting and ten days of tuning. The results therefore show what the method produced on real plant data; they do not show that the application runs in routine production, and that is not claimed.

Analysis Depth

Table 1 summarizes the scorecard counts on the reference casting. All numbers are counts in the text of stored reports, not plant data, and were recomputed from the source files.

Table 1

Scorecard counts on the reference casting. “Mentions of the defect location” counts mentions of the ring zone around the cores, where the reference analysis located the defect; “values with a unit” counts numbers read with a physical unit.⌘

Reports	<i>n</i>	Mentions of the defect location	Values with a unit
Reference analysis	1	34	84
One-shot analyses, first half of June	16	0 in every report	4 to 21 (median 9)
Agent without mandates	1	0	15
Agent with five mandates	3	16, 11 and 3	6, 12 and 20

⌘ Only one stored report without mandates is counted; the report of an earlier run was overwritten, and its documented counts were 0 mentions and 4 values.

None of the sixteen archived one-shot analyses mentioned the actual defect location. None contained a section admitting uncertainty, and none used the geometry of the part in its argument; the month of corrections had moved only the citations. The agent without mandates also never mentioned the defect location and wrote twelve times about the textbook feeder neck. With the five mandates, on the same casting, with the same model and without any new tool, mentions of the defect location rose from zero to 16, 11 and 3 in the three runs; two tool behaviors did change between the runs, as described under

“Limitations”. The mechanism of the reference analysis appeared in all three runs with mandates; in two the naive hypothesis was refuted, and in the third it was accepted for the central hot spot and supplemented by the ring zone. The density of numbers stayed far below the reference, 6 to 20 values against 84. The remaining gap lies in the depth of quantification, not in where the model looks. The three runs also show how much a single run can mislead: the same casting, the same model and the same mandates gave those three counts, although the input files differed between runs. The comparison base is the single stored run without mandates, and with three runs on one casting the counts describe a direction, not an effect size; they carry no confidence interval and are not a statistical result.

Retrieval

In a single common index, the long cards of standards, procedures and book chapters were systematically outranked by thousands of short manual pages, because BM25 normalizes scores by document length. After the index was stratified by library part, the number of company procedures and standards among the five best results rose about fivefold on a test set of 35 queries. Vector (“semantic”) search was tried twice. The second attempt, a hybrid of lexical ranking and vector similarity over all of the roughly 14,000 cards, was measured on real queries: the share of long chapters among the best results fell by up to 87% and that of procedures and standards to about half. A variant fusing the two rankings by reciprocal rank fusion⁴² did not help, and vector search was rejected and switched off within two days. The failure prediction written in the decision record beforehand came true. The benefit of returning neighboring pages was immediate during tuning but was not quantified.

Negative Results

The digital twin. In May 2026, the project’s largest build, a digital twin of the operation, grew within about three weeks to more than twenty components: a model of the line’s machines built from the signal table, a scanner of limit exceedances, a diagnostic agent, a maintenance planner that created work orders with deterministic identifiers, and set-point proposals checked by a safety envelope with an “approve and write” button. All 517 automated tests passed, and the architect model declared four control loops closed. The envelope rejected a proposal if the parameter did not exist at the node, if the value lay outside absolute limits, or if the step from the current value was too large. But the limits had been entered by the developers and could not be traced in the manufacturers’ documentation, the “current value” was supplied by the model rather than by live data, and the “write” waited half a second and added a line to a log; no industrial communication library was ever part of the project. The interface was not used in the plant. On 24 May 2026 the author removed the interface of the twin and the whole tab for inspection support, and the machine model with its limits was deleted. The code of the agents and of the envelope remains, without limits.

The sand moisture mapping error. The batch export contains two moisture measurements: an electrode sensor in the return-sand hopper before the mixer, which the control uses to compute the water addition, and an automatic sand tester that measures moisture, compactability and strength of the prepared sand after the mixer. Water is added between the two points, so their levels differ naturally; how much of the

difference is due to mixing and how much to instrument error remains open, and the tester was itself unstable in June. An early decision record placed the first sensor on the cooler and the data dictionary described it as prepared sand; both errors came to light when the sensor mapping was checked, after which the system reviewed the whole dictionary in three passes and put twelve further doubtful entries up for review. When a new node was added to the machine model, the system applied the limits of the prepared sand to the return-sand sensor, and almost every measurement was flagged as an anomaly. The error was found the same day. Counts of simultaneous exceedances computed by the twin are not published even as findings: the moisture limit had been derived from data averages, and the temperature limit belonged to a different measuring point. A system that flags almost every measurement has not found a fault; it has found an error in its own picture of the process. A related lesson from the same period: the compensation algorithms for sand moisture and temperature had been built on a machine whose dust extraction had not been cleaned for a long time and whose water spray was uneven. The decision record summarized it as “we are optimizing noise”: software optimization may only build on a mechanically sound machine.

Retracted findings. A statistically convincing result, higher pouring temperature associated with more gas defects, was withdrawn when a check of the source data showed that the “temperature” came from the line’s optical pyrometer, not from immersion measurements, and that reliable immersion data existed for only a few months. The temperature conclusions were withdrawn as resting on an unreliable channel and, for the earlier period, closed as unprovable rather than “probably valid”.

Classes of Plant Findings

Plant results are reported as classes. A chronic fault of a positioning axis of the pouring unit and its position sensor turned out to be the common root of outages that had been recorded as unrelated, including the longest pouring outage in the analyzed period; three independent records confirmed the mechanism (the pouring-unit event log, the molding-line machine records and the manual downtime log), and the system stated what the data do not prove. Forgotten and provisional settings were found, such as a cooler with practically no air flow while running, a control element parked at a fixed value, and a purchased control function that had never been switched on. For a steel casting with cores, where the simulation offered shrinkage in the feeder neck as the textbook answer, reading the fields at the place and moment of defect formation, the metallography of a defective casting and the literature pointed to microporosity in rings around the cores, where a core stops extracting heat and starts to insulate; the standard porosity export did not show it at all. Statistics on production data refuted several long-held beliefs and confirmed factors that nobody had tested: scrap came in waves spanning several days rather than individual heats, the chemical composition of the heat did not discriminate scrap, the pouring regime was the strongest measurable signal, and slag appeared to be its consequence rather than its cause. In zones of castings that radiography could not reliably examine, few defects were recorded; the system treated this as data censored by the detection limit rather than as sound metal, and metallography of samples from such a zone later confirmed an increased occurrence of internal defects. The inability to detect a defect belongs in the detection rating of the failure mode and effects analysis

(FMEA) and raises its action priority⁴³. The system also read compiled archives of control programs without modifying them, extracted signal interfaces and alarm texts, and counted downtime records that had lain unused in the data; the largest downtime item was core setting, concentrated in one shift, although attention had focused on pouring. Finally, the system withdrew several of its own findings. How many findings a system withdraws says more about its reliability than how many it produces.

Discussion and Limitations

What Carried the Result

Three observations recur throughout the development. First, instructions to a model were not enough for anything that had to hold every time. The material rule was in the instructions when the model, reading a bare material-standard number, took a cast steel for a ductile iron and recommended a pouring temperature below the liquidus of steel, a setting at which the metal would freeze in the ladle; the rule applied to the experts but not to the judge, and an audit a week earlier had shown that the model caught a wrong material in only two of three runs. Once the material was anchored by code, the confusion did not recur in the agent runs. Second, gates guard only the floor. The June reports passed every check and still missed the defect. Depth came from the task: five sentences in the mandate moved the analysis to the right mechanism, while the first reaction, adding tools, had not. Third, measurement was indispensable. Without the scorecard, neither the month of ineffective corrections nor the effect of the mandates would have been visible; judgment by impression (“it is shallow”) gives no direction.

The development process itself was part of the method. Separating an architect from an executor, with a human relaying every directive, produced written premises that could be checked, and the executor’s objections stopped at least one directive based on a misreading of a document title. It also had costs: the architect worked from summaries, and agreement between two models could be an echo rather than confirmation. Neither model noticed a month of circling around a flawed core; the author, who had followed all the runs, did.

Limitations

The measurement has clear limits. The reference analysis was written by an LLM under the author’s guidance; it is not an independent expert report, and a blind comparison with independent experts on historical cases is still missing. The mandates were derived from the same casting on which they were measured, one mandate contains a term that the scorecard counts, the runs were few and their inputs varied, and the improvement was measured in the isolated agent package, not in the production tab. The scorecard counts words and headings and cannot judge the correctness of reasoning. The trace check of the agent currently only prints violations and does not store the trace with the report. Between the runs without and with mandates, the fact tool also began reminding the agent to examine the cores when their number was unknown, and a loop guard and handling of missing files were added; no new tool was introduced.

The gates protect against errors of the model, not against instructions deliberately hidden in supplier documents. Retrieval quality was measured only in one-off audits, without a continuously maintained labeled set, and the citation check verifies that a quotation is verbatim, not separately that it comes from the right page; the author considers this the largest real gap. There is no regression set run on every change of model or instructions, and a change of a model at the provider can shift behavior without any change in code.

Operationally, the system is offline, receives weekly exports and calls cloud models; the decision about which data may leave the company belongs at the start of any deployment. Cooling curves and nodal data are not delivered from the simulation, the dual geometry of casting and machined part is missing for some parts, the tapping time exists only in the melt-shop system, and scrap is not recorded per mold. No formal governance of the AI system has been set up with respect to the EU Artificial Intelligence Act⁴⁴ or an AI management system standard⁴⁵, and no conformity assessment has been made. The system depends on the knowledge of one person; its documentation was written so that this need not be so, but a handover has not been tested. Time savings were not measured, so none are claimed.

In summary, the evidence base is narrow. The depth measurement rests on one reference casting, three runs with mandates and a single stored run without them. The plant findings come from one plant and are reported as classes of findings, not as validated performance. Missing are a blind comparison with independent human experts, a regression set across model versions, and any trial or routine production use.

Transferability

What transfers is the ability to build such a system again: the gates, the methods and the measurement. What does not transfer is the content: the library of a plant's documentation, the knowledge cards of its equipment, the data contracts for its exports, the dictionary of its signals and alarms, reference analyses of its castings and the concrete content of the gates. T1 status must be verified on each machine again. Like most published LLM applications in casting, this system has not gone beyond development and offline work on historical plant data; it is in neither trial nor routine production use. What distinguishes it is not the degree of deployment but the traceability that code enforces.

Roadmap and Autonomy Ceiling

The three cheapest improvements require neither development nor an MES connection: using the plant's existing data systematically; adding six fields per mold that the pouring unit's control system already knows but does not export (ladle arrival time, treatment time, flags for new metal, ladle change and pattern change, and ladle identity), without which the true metal age cannot be computed for some heats; and finding out where the line data are stored and whether read access is possible. The development plan has six stages: completing the analysis core (quantification density, cooling curves, dual geometry, further roles as agents), automatic data intake, a read-only MES connection, live data with

a continuous deterministic layer and the model called only on anomalies, deployment with roles and authentication, and a deliberate boundary of the loop. Only the third stage depends on the MES.

Table 2 lists the levels of autonomy, adapted from the classification of automation levels by Parasuraman et al.⁴⁶.

Table 2
Levels of autonomy and the recommended ceiling

Level	What the system does	Who decides	Status
0 On request	builds an analysis from historical data when a person asks	human	current operation
1 Informative	reports an anomaly with explanation and evidence on its own	human	requires automatic data intake and live data
2 Advisory	proposes a specific intervention with its mechanism	human	requires automatic data intake and live data
3 Preparatory (ceiling)	prepares a value, checks it against the safety envelope and waits for confirmation	human confirms	envelope code exists; writing only simulated
4 Autonomous	writes without confirmation	nobody	not recommended even after full validation

Level 3 is the realistic target. On the evidence of this project the author does not recommend Level 4 even after full validation, because an envelope protects against gross errors, not against a value that is physically admissible and still wrong for the casting at hand; the digital twin also showed how easily an envelope can be built on limits without a source and on a current value supplied by the model. Automatic writing to the control system is technically the smallest part and organizationally the largest risk. That responsibility lies with whoever releases the casting to the customer, and that should never be software. For a pilot, four indicators are proposed instead of impressions: time from detection of a defect to a supported conclusion, the share of conclusions accepted by the process engineer without substantial correction, the share of outputs carrying a warning flag, and the share of claims with a verified citation.

Conclusions

These conclusions come from one system in one plant and from the measurements reported above. They are lessons from that work, not general results.

1. A rule that relies on the model to follow it is not a rule. Whatever must always hold has to be enforced by code; the material anchor, verbatim citation checks and the image-reference lock are examples.
2. Decision gates guard only the floor. Depth came from the task given to the model: on the reference casting, five mandates, without any new tool, raised mentions of the actual defect location from zero to

16, 11 and 3 in three runs, while quantification stayed below the reference.

3. Simulation fields should be read where the defect forms: at the place, the moment and above the threshold of its mechanism. An empty porosity image is not a sound casting.

4. A manual describes a product, not the machine on the shop floor. Only claims verified on the specific machine belong in operator procedures.

5. Understanding the process precedes computation. Two moisture measurements at two points of the sand plant, a counter that never resets and a mold-making time that is not the pouring time were traps in the data.

6. Tearing down finished and tested work that nobody needs is a legitimate result, and measuring is necessary even when the result is unwelcome. The system proposes; a human signs.

Nomenclature and Abbreviations

Symbol	Meaning	Unit
A	surface area of a casting section	mm ²
B	mold constant in Chvorinov's rule (for n = 2)	s·mm ⁻²
G	temperature gradient	K·mm ⁻¹
M	casting modulus, $M = V/A$	mm
n	exponent in Chvorinov's rule (≈ 2)	–
Ny	Niyama criterion, $Ny = G/\sqrt{T}$	(K·s) ^{0.5} ·mm ⁻¹
t _f	local solidification time	s
T	cooling rate	K·s ⁻¹
V	volume of a casting section	mm ³
T1, T2, T3	trust tiers of claims about equipment (Sect. "Trust Tiers for Equipment Claims")	–

BM25, Okapi best-matching ranking function; FMEA, failure mode and effects analysis; LLM, large language model; MES, manufacturing execution system; RCA, root-cause analysis.

Declarations

Acknowledgements

The author thanks all his former co-workers from the companies he has worked for during his career, the specialists, technicians and colleagues in foundries, production and maintenance, for the inspiration that came from solving real situations in his practice.

In generating manuscript content, the author used the following AI tool: a general-purpose LLM assistant of vendor B (name and version available to the editor on request). The tool reviewed the project documentation, drafted the text of all sections from the author's material and from a verified Czech manuscript, and translated it into English; the author directed the work, reviewed and edited the manuscript content and takes full responsibility for it. The LLMs that are the subject of the study, and their roles in developing the software, are described in the Methods. The figures are schematic diagrams and charts drawn programmatically by scripts written with the assistance of the same tool, from the text and from recomputed counts; they are not images from generative image models.

Funding. No funding was received for conducting this study.

Competing interests. The author holds exclusive rights to the system described in this article (see "Origin of the system and rights"). The author has no other relevant financial or non-financial interests to disclose.

Origin of the system and rights. The system described in this article, in its concept and design as presented here, was conceived by the author personally, without the assistance of other persons, and the author holds exclusive rights to it. No employer assigned or commissioned the development of the system; it was undertaken on the author's own initiative.

Ethics approval and consent to participate. Not applicable.

Consent for publication. Not applicable.

Data availability. The operational data, documentation and simulation results used in this study are proprietary to the foundry that provided them and cannot be shared. The scorecard counts in Table 1 were recomputed from stored reports using the definitions given in the Methods.

Materials availability. Not applicable.

Code availability. The source code of the system is not publicly available.

Author contribution. The author conceived and designed the system, directed its implementation with the LLM tools described in the Methods, performed the analyses and prepared the manuscript with the AI assistance disclosed in the Acknowledgements.

Use of AI. LLMs are the subject of this study (see Methods, "Development Process and Use of LLMs"). AI assistance in preparing the manuscript and the figures is disclosed in the Acknowledgements.

References

1. H. Yu, Z. Yin, Z. Jin et al., Assessing the proficiency of large language models in the metal casting: an empirical study. *Int. J. Metalcast.* (2026). <https://doi.org/10.1007/s40962-026-01888-5>
2. Y. Guo, J. Yang, C. Yang, A review of applications and challenges of large language models for foundry intelligence in the casting industry. *Comput. Mater. Contin.* 88(1) (2026). <https://doi.org/10.32604/cmc.2026.077820>
3. T. Prucha, From the editor. *Int. J. Metalcast.* 17(2), 589–591 (2023). <https://doi.org/10.1007/s40962-023-01025-6>
4. M. Breque, L. De Nul, A. Petridis, *Industry 5.0: Towards a Sustainable, Human-Centric and Resilient European Industry* (European Commission, Directorate-General for Research and Innovation; Publications Office of the European Union, Luxembourg, 2021). <https://doi.org/10.2777/308407>
5. J. Leng, W. Sha, B. Wang et al., Industry 5.0: prospect and retrospect. *J. Manuf. Syst.* 65, 279–295 (2022). <https://doi.org/10.1016/j.jmsy.2022.09.017>
6. Česká slévárenská společnost, Historie [History; in Czech]. <https://ceskaslevarenska.cz/historie/>. Accessed 21 September 2026
7. J. Nieves, A. Perez, Supporting foundry workers with an AI-based multimodal system tuned using foundry knowledge, in 75th World Foundry Congress (Deyang, China, 2024), pp. 768–769
8. N. Sun, A. Kopper, R. Karkare, R.C. Paffenroth, D. Apelian, Machine learning pathway for harnessing knowledge and data in material processing. *Int. J. Metalcast.* 15(2), 398–410 (2021). <https://doi.org/10.1007/s40962-020-00506-2>
9. T.Ç. Uyan, K. Otto, M.S. Silva, P. Vilaça, E. Armakan, Industry 4.0 foundry data management and supervised machine learning in low-pressure die casting quality improvement. *Int. J. Metalcast.* 17(1), 414–429 (2023). <https://doi.org/10.1007/s40962-022-00783-z>
10. S. Pachandrin, N. Hoffmann, K. Dilger et al., Data-driven prediction of casting defects in magnesium high-pressure die casting using machine learning. *Int. J. Metalcast.* 20(1), 112–131 (2026). <https://doi.org/10.1007/s40962-025-01592-w>
11. J. Kang, B. Liu, T. Jing, H. Shen, Intelligent casting: empowering the future foundry industry. *China Foundry* 21(5), 409–426 (2024). <https://doi.org/10.1007/s41230-024-4056-z>
12. X. Wu, X. Wang, Y. She, M. Sun, Q. Gao, Root cause analysis of cast product defects with two-branch reasoning network based on continuous casting quality knowledge graph. *Appl. Sci.* 15(13), 6996 (2025). <https://doi.org/10.3390/app15136996>
13. B. Zhou, X. Li, T. Liu et al., CausalKGPT: industrial structure causal knowledge-enhanced large language model for cause analysis of quality problems in aerospace product manufacturing. *Adv. Eng. Inform.* 59, 102333 (2024). <https://doi.org/10.1016/j.aei.2023.102333>
14. Y. Ma, S. Zheng, Z. Yang, H. Pan, J. Hong, A knowledge-graph enhanced large language model-based fault diagnostic reasoning and maintenance decision support pipeline towards industry 5.0. *Int. J. Prod. Res.* 64(12), 5239–5260 (2026). <https://doi.org/10.1080/00207543.2025.2472298>
15. K. Wang, C. Liu, Y. Lu, Ensemble Bayesian network for root cause analysis of product defects via learning from historical production data. *J. Manuf. Syst.* 75, 102–115 (2024).

<https://doi.org/10.1016/j.jmsy.2024.06.001>

16. S. Yao, J. Zhao, D. Yu et al., ReAct: synergizing reasoning and acting in language models, in International Conference on Learning Representations (ICLR 2023) (2023).
<https://doi.org/10.48550/arXiv.2210.03629>
17. Y. Chen, H. Xie, M. Ma et al., Automatic root cause analysis via large language models for cloud incidents, in Proceedings of the Nineteenth European Conference on Computer Systems (EuroSys '24) (ACM, 2024), pp. 674–688. <https://doi.org/10.1145/3627703.3629553>
18. Y. Du, S. Li, A. Torralba, J.B. Tenenbaum, I. Mordatch, Improving factuality and reasoning in language models through multiagent debate, in *Proceedings of the 41st International Conference on Machine Learning*, PMLR 235 (2024), pp. 11733–11763
19. V.I. Bocanet, M.H. Muntean, C. Fleşeriu, Multi-agent framework for AI-supported collaborative root cause analysis in quality assurance, in Advances in Production Management Systems (APMS 2025), IFIP AICT, vol. 766 (Springer, Cham, 2025), pp. 202–216. https://doi.org/10.1007/978-3-032-03538-7_15
20. M.A. Farahani, M.I. Khan, T. Wuest, Hybrid agentic AI and multi-agent systems in smart manufacturing. *J. Manuf. Syst.* 86, 612–623 (2026). <https://doi.org/10.1016/j.jmsy.2026.04.002>
21. A. Wei, O. Fink, Agentic root cause analysis through evidence-grounded reasoning. *arXiv:2607.22385* (2026). <https://doi.org/10.48550/arXiv.2607.22385>
22. Lumafield, Lumafield introduces Quality Agent, a physical AI technology that automates defect detection and root-cause analysis for manufacturers [press release]. *GlobeNewswire* (3 September 2026). <https://www.globenewswire.com/news-release/2026/09/03/3355956/0/en/>. Accessed 21 September 2026
23. Siemens AG, Siemens introduces AI agents for industrial automation [press release] (12 May 2025). <https://press.siemens.com/global/en/pressrelease/siemens-introduces-ai-agents-industrial-automation>. Accessed 21 September 2026
24. P. Lewis, E. Perez, A. Piktus et al., Retrieval-augmented generation for knowledge-intensive NLP tasks, in Advances in Neural Information Processing Systems 33 (NeurIPS 2020) (2020), pp. 9459–9474
25. S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.* 3(4), 333–389 (2009). <https://doi.org/10.1561/15000000019>
26. N.F. Liu, K. Lin, J. Hewitt et al., Lost in the middle: how language models use long contexts. *Trans. Assoc. Comput. Linguist.* 12, 157–173 (2024). https://doi.org/10.1162/tacl_a_00638
27. T. Gao, H. Yen, J. Yu, D. Chen, Enabling large language models to generate text with citations, in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (Association for Computational Linguistics, Singapore, 2023), pp. 6465–6488.
<https://doi.org/10.18653/v1/2023.emnlp-main.398>
28. N. Liu, T. Zhang, P. Liang, Evaluating verifiability in generative search engines, in Findings of the Association for Computational Linguistics: EMNLP 2023 (Association for Computational Linguistics,

- Singapore, 2023), pp. 7001–7025. <https://doi.org/10.18653/v1/2023.findings-emnlp.467>
29. V. Magesh, F. Surani, M. Dahl et al., Hallucination-free? Assessing the reliability of leading AI legal research tools. *J. Empir. Leg. Stud.* 22(2), 216–242 (2025). <https://doi.org/10.1111/jels.12413>
 30. L. Zheng, W.-L. Chiang, Y. Sheng et al., Judging LLM-as-a-judge with MT-Bench and Chatbot Arena, in *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track (2023)*, pp. 46595–46623
 31. P. Wang, L. Li, L. Chen et al., Large language models are not fair evaluators, in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Association for Computational Linguistics, Bangkok, 2024)*, pp. 9440–9450. <https://doi.org/10.18653/v1/2024.acl-long.511>
 32. E. Miller, Adding error bars to evals: a statistical approach to language model evaluations. *arXiv:2411.00640* (2024). <https://doi.org/10.48550/arXiv.2411.00640>
 33. D. Manheim, S. Garrabrant, Categorizing variants of Goodhart’s law. *arXiv:1803.04585* (2018). <https://doi.org/10.48550/arXiv.1803.04585>
 34. B.A. Nosek, C.R. Ebersole, A.C. DeHaven, D.T. Mellor, The preregistration revolution. *Proc. Natl. Acad. Sci. U.S.A.* 115(11), 2600–2606 (2018). <https://doi.org/10.1073/pnas.1708274114>
 35. G. Klein, Performing a project premortem. *Harv. Bus. Rev.* 85(9), 18–19 (2007)
 36. D.J. Snowden, M.E. Boone, A leader’s framework for decision making. *Harv. Bus. Rev.* 85(11), 68–76 (2007)
 37. N. Chvorinov, Theorie der Erstarrung von Gußstücken. *Giesserei* 27, 177 (1940)
 38. M.T. Rowley (ed.), *International Atlas of Casting Defects* (American Foundrymen’s Society, Des Plaines, IL, 1974)
 39. J. Campbell, *Complete Casting Handbook: Metal Casting Processes, Metallurgy, Techniques and Design*, 1st edn. (Butterworth-Heinemann, Oxford, 2011)
 40. E. Niyama, T. Uchida, M. Morikawa, S. Saito, A method of shrinkage prediction and its application to steel casting practice. *AFS Int. Cast Met. J.* 7(3), 52–63 (1982)
 41. K.D. Carlson, C. Beckermann, Prediction of shrinkage pore volume fraction using a dimensionless Niyama criterion. *Metall. Mater. Trans. A* 40(1), 163–175 (2009). <https://doi.org/10.1007/s11661-008-9715-y>
 42. G.V. Cormack, C.L.A. Clarke, S. Büttcher, Reciprocal rank fusion outperforms Condorcet and individual rank learning methods, in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM, 2009)*, pp. 758–759. <https://doi.org/10.1145/1571941.1572114>
 43. AIAG, VDA, AIAG & VDA FMEA Handbook, 1st edn. (Automotive Industry Action Group, Southfield, MI; VDA QMC, Berlin, 2019)
 44. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). *Off. J. Eur. Union L*, 12

45. ISO/IEC 42001:2023, Information Technology – Artificial Intelligence – Management System (International Organization for Standardization, Geneva, 2023)

46. R. Parasuraman, T.B. Sheridan, C.D. Wickens, A model for types and levels of human interaction with automation. IEEE Trans. Syst. Man Cybern. A Syst. Hum. 30(3), 286–297 (2000).
<https://doi.org/10.1109/3468.844354>

Figures

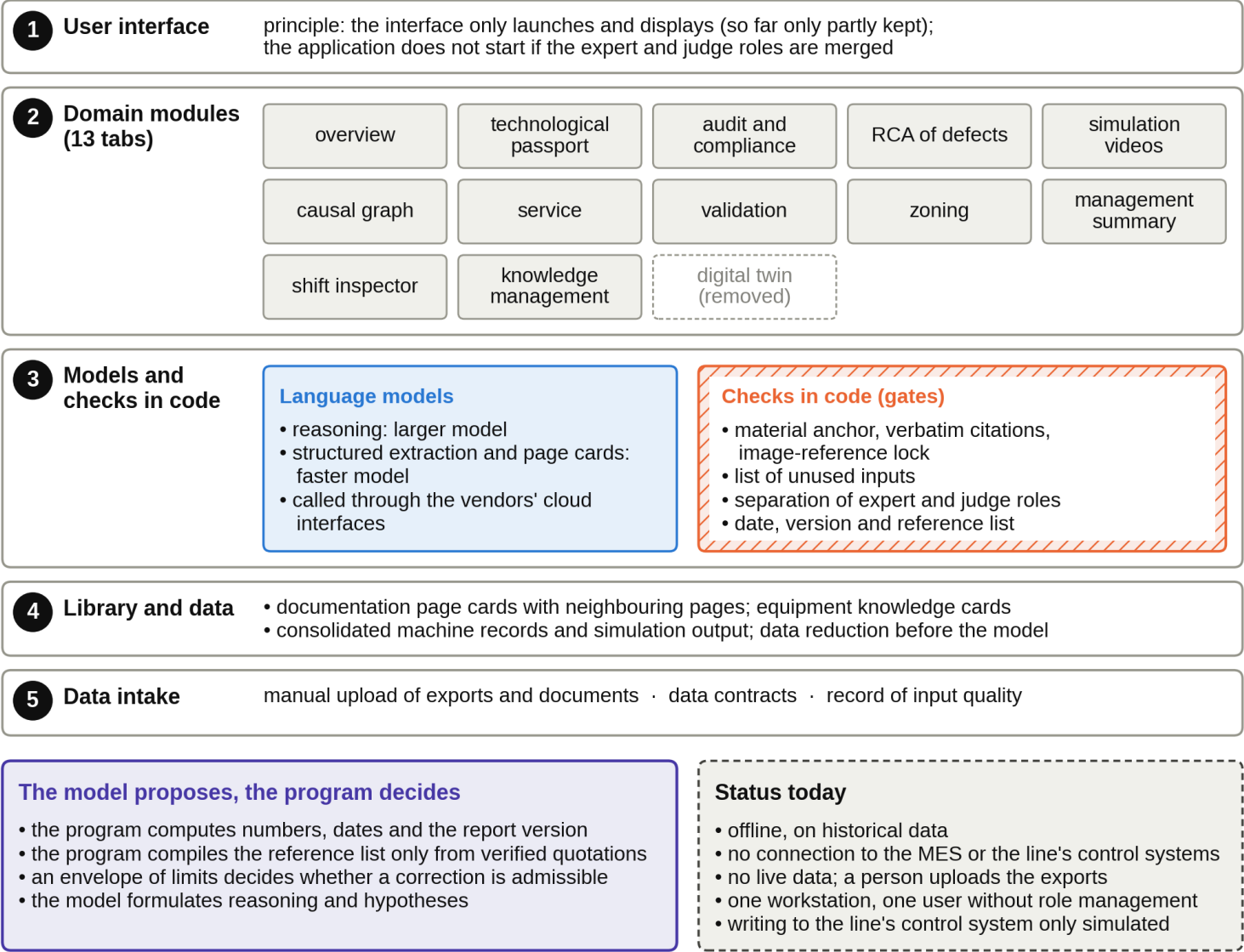


Figure 1

Five layers of the system. Numbers, dates, versions, the reference list and safety checks are handled by code, not by the language model.

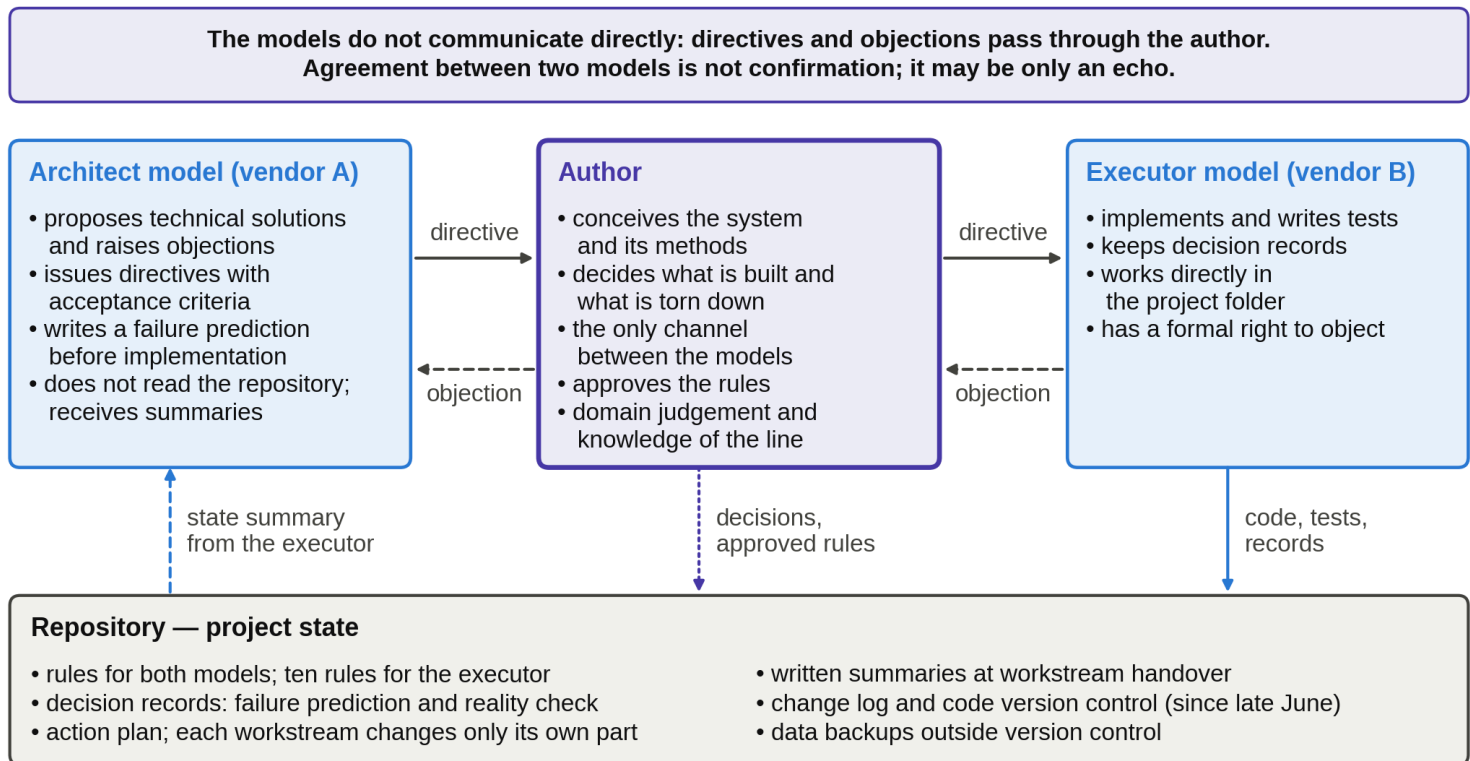


Figure 2

Division of labour during development. The models do not communicate directly; directives and objections pass through the author, and the state of the project lives in the repository.

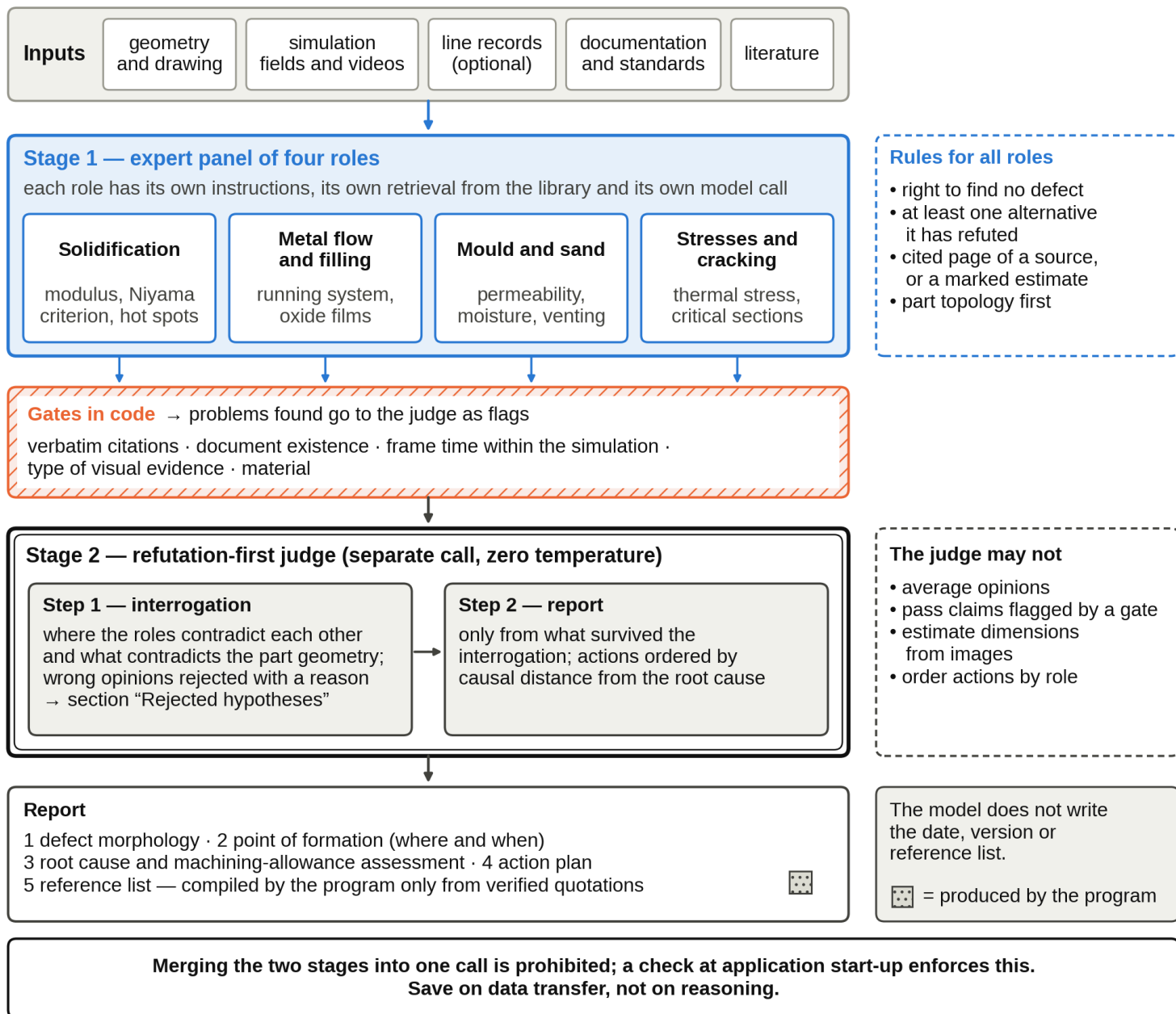


Figure 3

Two separate stages of reasoning. The judge starts by refuting the expert opinions.

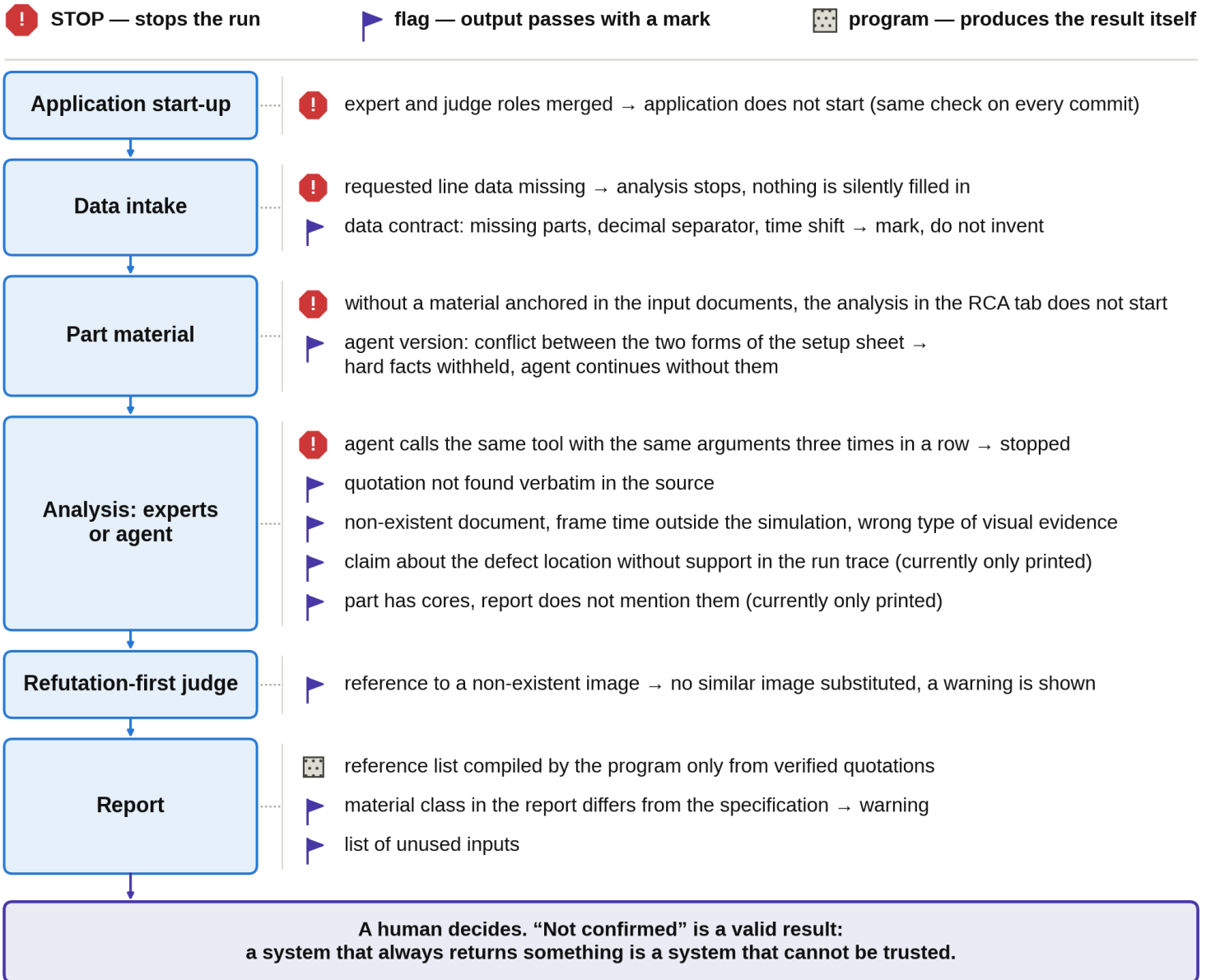


Figure 4

Points on the way from inputs to report where the program stops the work (STOP), flags the output, or produces the result itself.

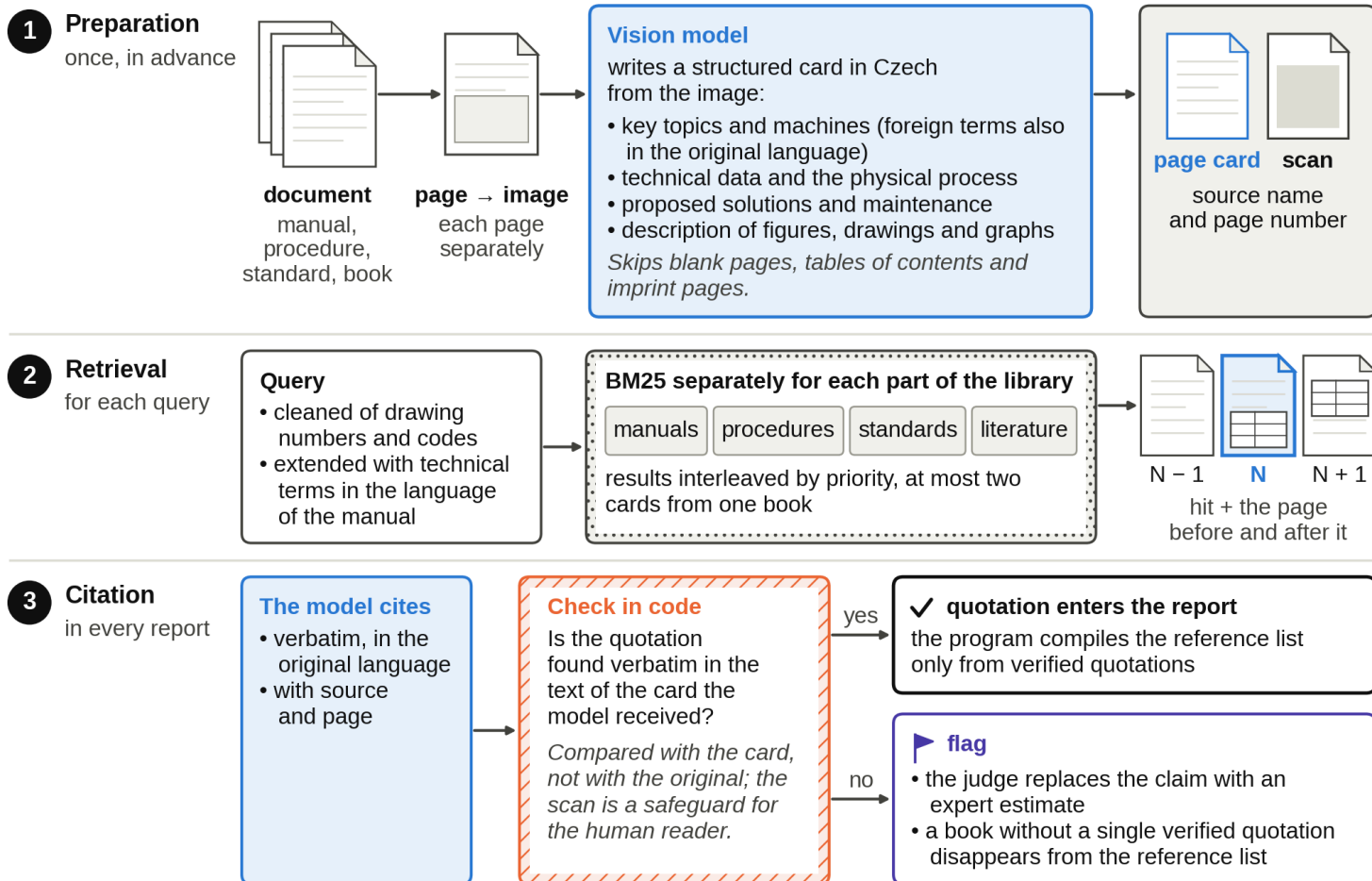


Figure 5

From document page to verified citation. Retrieval always returns the neighbouring pages; a quotation enters the report only if it matches the card verbatim.

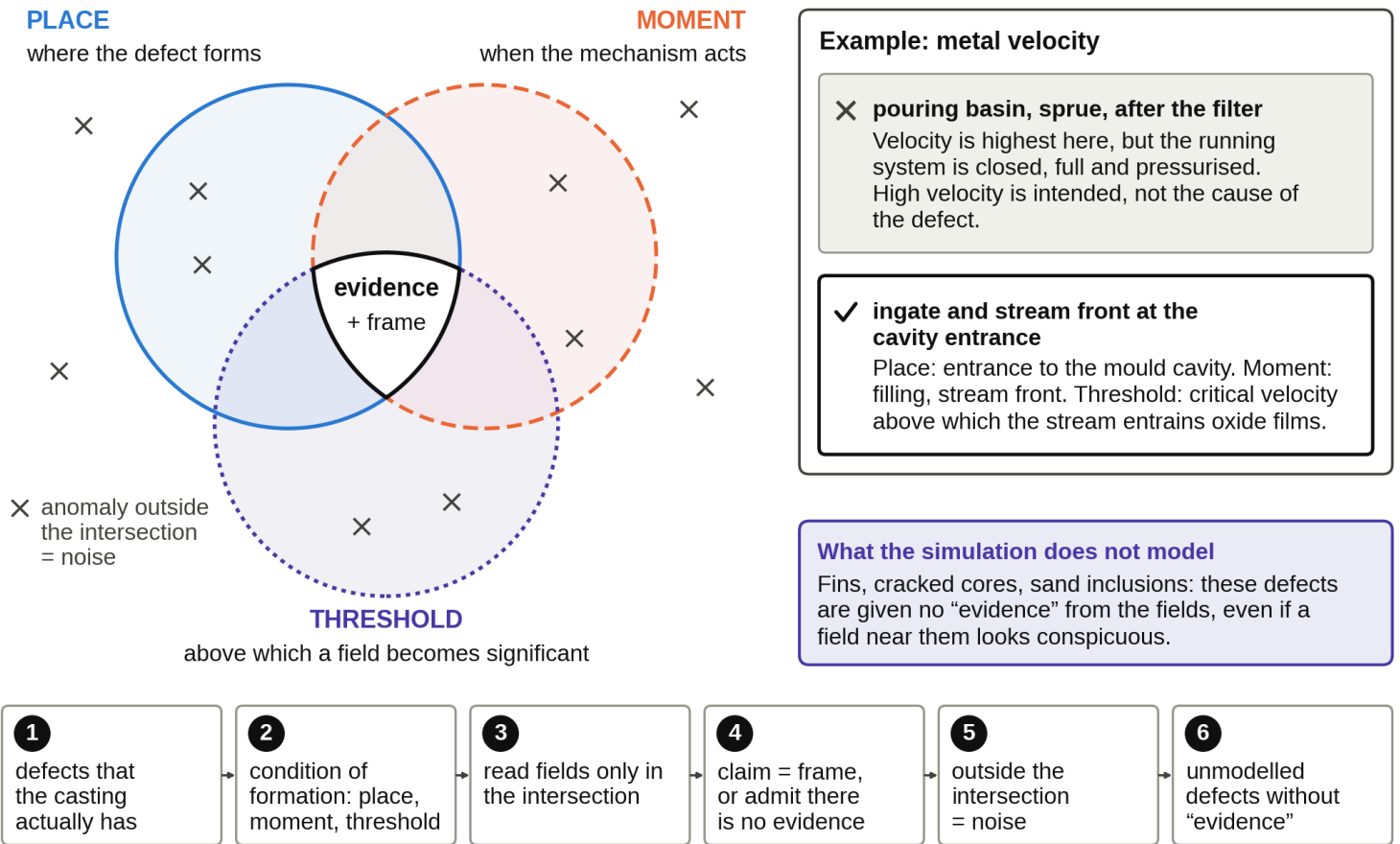


Figure 6

When a simulation field may serve as evidence of a defect: only in the intersection of place, moment and threshold, and with a specific frame.

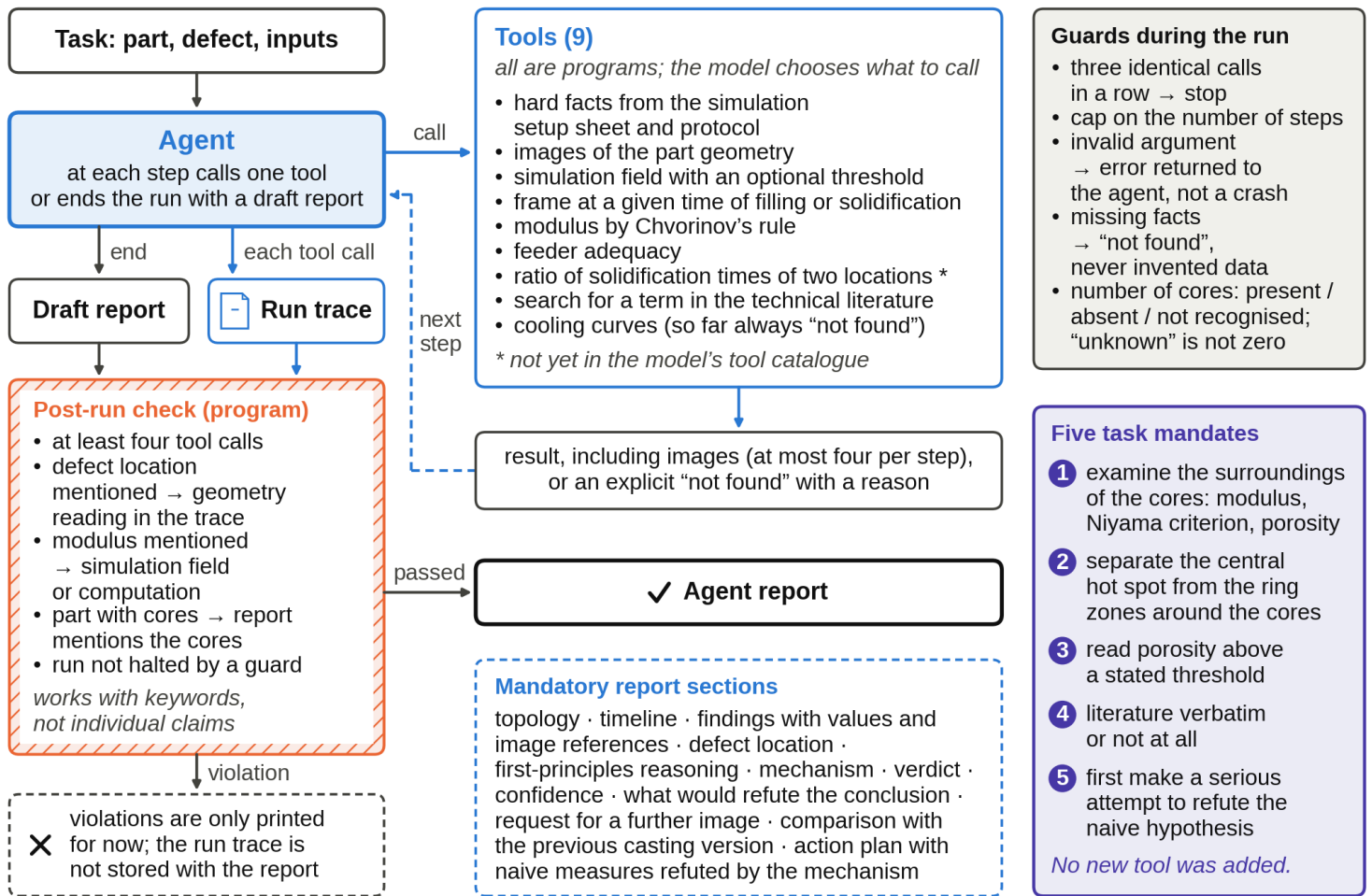


Figure 7

The tool-using agent. Each tool call is written to the run trace, and a post-run check compares the claims of the report with the trace. The five mandates changed only the task; no tool was added.