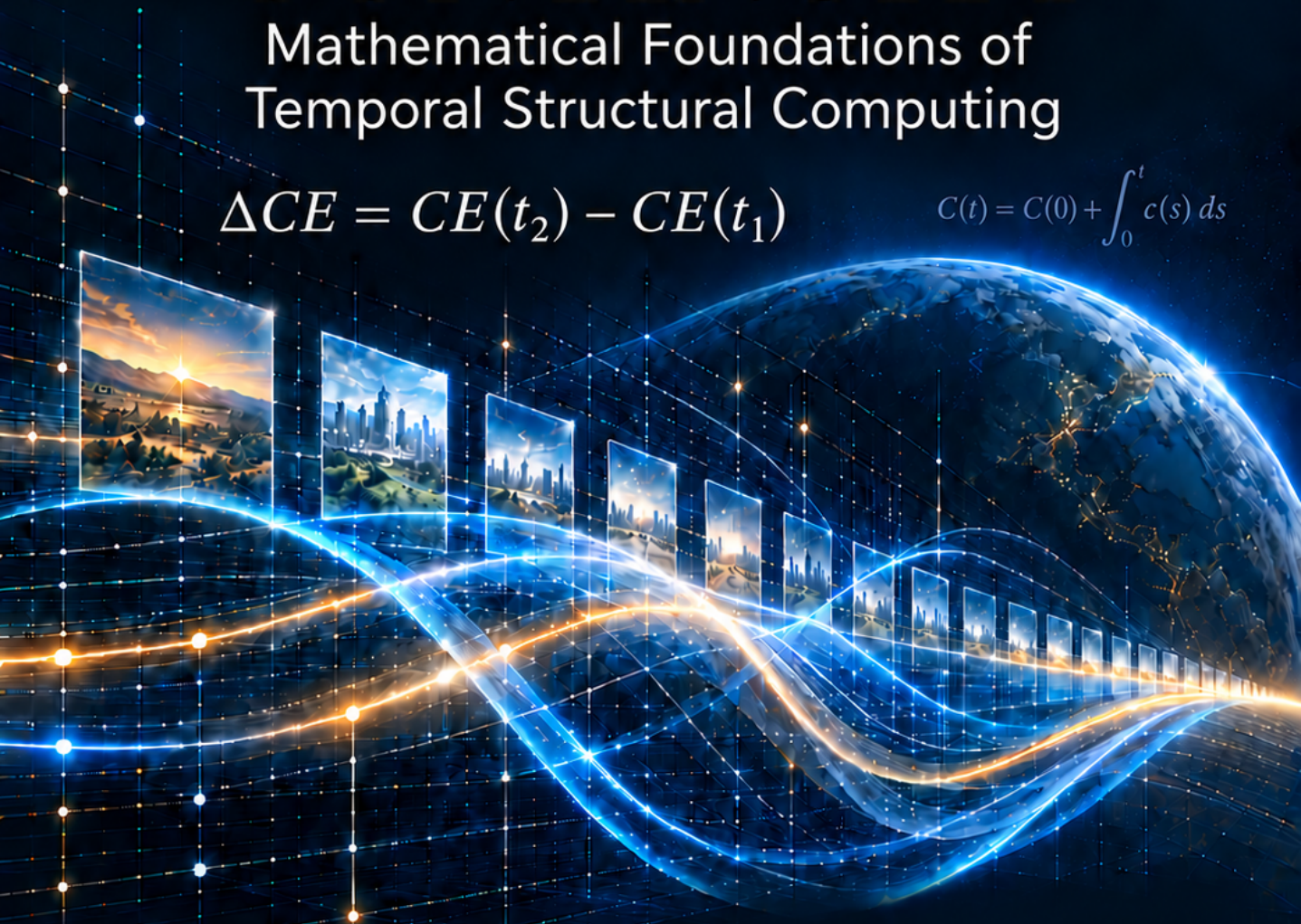
COMPUTING • SYSTEMS • TIME

ENGINEERING STRUCTURAL CONTINUITY

Mathematical Foundations of Temporal Structural Computing

$$\Delta CE = CE(t_2) - CE(t_1)$$

$$C(t) = C(0) + \int_0^t c(s)\, ds$$



$$\Delta G_T(t) = G(t+T) - G(t) = \int_t^{t+T} G'(s)\, ds$$

$$C_{\mathrm{H}}^{0}(h) \Rightarrow C_{\mathrm{H}}^{m}(h)$$

RAFAEL FELIPPE VOIGT

RAFAEL FELIPPE VOIGT

Engineering Structural Continuity

Mathematical Foundations of Temporal Structural Computing

Trajectory, memory, identity and transformation

A mathematical introduction to the continuity of systems

INDEPENDENT AUTHOR'S EDITION | 2026

Brazil, 2026

AI-assisted translation for linguistic and technical review. The Portuguese master governs interpretation.

The approved cover is localized in English. Titles and text are translated in this edition; mathematical symbols and original bibliographic titles are retained.

Authorship, edition and nature of the work

Rafael Felippe Voigt

Research, conception and scientific authorship

Engineering Structural Continuity
Mathematical Foundations of Temporal Structural Computing

Rafael Felippe Voigt is an independent researcher, founder of VoigtCore and creator of VCore technology. He is the author of the proposals of Temporal Structural Computing and Structural Continuity Engineering. His research investigates memory, identity, integrity and the evolution of computational systems. VoigtCore identifies the technological ecosystem associated with his research. Rafael Felippe Voigt is responsible for the authorship and independent publication of this work.

© 2026 Rafael Felippe Voigt. DEMONSTRATION EDITION — 2026. A selection prepared for presentation and editorial assessment; this is not the definitive commercial edition.

The work brings together established foundations of mathematics and computing, proposed formulations by the author, teaching models and open validation questions. Formulating a model, proving results within it and confirming it experimentally are distinct stages. An intellectual property registration or a DOI replaces none of them.

Artificial intelligence assisted the editorial preparation, supporting prose, layout, diagrams and consistency checks. AI image generation assisted the preparation of the cover. The numerical examples are synthetic; they do not describe measurements of VCore products. The work presents neither universal empirical validation of the framework nor independent assessment of the commercial products.

This work presents an engineering proposal and a formalization programme grounded in Temporal Structural Computing. Definitions, propositions and hypotheses have different levels of maturity, made explicit throughout the text. The empirical validity of applications requires specific experimental procedures. This sample does not include an executable supplement.

Scientific conventions

Definition: fixes meaning within the chosen domain. Hypothesis: states a condition to be examined. Proposition: is accompanied by a proof under explicit conditions. Example: develops a calculation or an interpretation. Teaching model: provides public relationships for teaching and does not identify a product estimator.

FREE DEMONSTRATION SAMPLE | 2026

ISBN of the complete Portuguese-language edition: 978-65-970107-0-7.

This ISBN identifies the complete Portuguese-language edition, not this translated sample.

Acknowledgments

First and foremost, I thank God. All honour, all glory and all gratitude belong to Him. In faith I found guidance and strength to continue through times of doubt and perseverance. This book also grows out of gratitude for the experiences and opportunities to learn throughout life.

In memory of my father, Sigold Voigt, who passed away at 49. Writing these words moves me deeply and brings back how much I miss him. Dad, you will always be in my heart.

To my mother, Ivete M. Schneider Voigt, I offer my thanks for her emotional, spiritual and financial support, and for her trust as these ideas took shape. Her presence accompanied the reflections, uncertainties and work needed to turn questions into research.

To my daughter, Isabella Voigt, one of the greatest inspirations in my life, I offer my love and the wish that knowledge, discipline and perseverance may contribute to her future.

To my sister, Fernanda Voigt, I give thanks for her support and encouragement. To my nephew Bernardo, I offer the wish that curiosity and the courage to investigate may accompany his journey.

To my friend Marcos Aurélio, I give thanks for his trust and encouragement, even when the results existed only as ideas.

To my aunt Rute Salete Schneider, my dear Tia Rute, I offer my heartfelt thanks for my first computer. That gift belongs to the beginning of my story with computing, and I am deeply grateful for her part in that journey.

To my aunt Janice, my aunt of light, I give thanks for always praying for me. Please receive my love and gratitude for your affection and spiritual presence in my life.

To the Schneider and Voigt families, I offer my affection and gratitude for being part of my story.

To everyone who contributed through conversations, questions, criticism and support: thank you. May this work invite careful study, investigation and the building of knowledge for the benefit of society.

Rafael Felippe Voigt

Author's preface

What remains true when a system changes? This question guides the book. A machine receives components, a program changes version, an organization replaces teams and an agent uses another model. The present state remains important, but investigating a prior obligation requires relationships that an isolated reading may not preserve.

The course of my research brought together questions of sustainment, coherence and temporal governance and connected them to computational systems. Physical observation of computing prompted questions about consumption, disturbance and maintenance effort. Identity or memory does not follow automatically from these signals; the relationships must be constructed and their conditions examined.

I call Structural Continuity Engineering the proposal to organize this problem as engineering work: declare the boundary, represent the trajectory, define what must persist, formulate hypotheses, calculate consequences and confront them with evidence. In this approach, MovieShot names a longitudinal representation in which frames preserve relationships and context. The observed history must remain interpretable for the questions we undertake to answer.

Logs, temporal systems, formal verification, control theory and recovery already provide important tools. This book engages with those traditions. The contribution sought is to bring questions of trajectory, memory, identity, integrity, assets and future capacity together in an examinable formulation of continuity.

Mathematics serves both teaching and critical examination here. A dependency relationship is not yet a calibrated law. A quantity needs a domain, a unit, an interpretation rule and conditions for comparison. A proof holds under its assumptions. When a formulation needs revision, scientific work becomes more precise by making that revision explicit.

I want the reader to be able to repeat a calculation, test a hypothesis, construct a counterexample and formulate a better question. The book is a foundation for study and an invitation to independent examination of the proposal. The continuity I seek to understand also begins with a commitment to preserving the conditions under which knowledge was produced.

How to study this engineering approach

A cumulative path

Chapters 1 to 4 introduce states, events, time, measurement and representation. Chapters 5 to 8 develop memory, identity, quantities and governance. Chapter 9 prepares the trial; the last three apply the foundation to a machine, an agent and physical infrastructure. The examples are synthetic. Section 10.8 presents a finite migration model and describes its optional supplement, independent of products.

Before calculating

Begin with the question, the boundary and the obligations. Declare the object, units, instants, assumptions and gaps. Repeat the example and construct a counterexample. An equality may be a definition, an identity, a model hypothesis or an estimated relationship: recognize which use has been declared. Proof and experimental confirmation answer different questions.

Notation and practice

t denotes physical time when its scale is declared; k and τ may simply index transformations. Symbols such as C, H and T have local scope, made explicit in each model. The exercises progress from calculation to interpretation, proof and investigation. Consult the selected solutions after attempting the problems. Definitions and proofs make explicit the domains and assumptions required for the results being taught.

For students and teachers

Explain each result in your own words. Check where each assumption enters the proof and change a condition to test its scope. In class, begin with the concrete problem, repeat a calculation and ask for a conclusion that distinguishes what was calculated from what requires evidence. Transferring an example to practice requires instrumentation, data and criteria from the domain; observing a difference does not establish its cause.

Guide to reading the symbols

This page serves as a reference. Each symbol reappears in the context in which it is needed. Reading it in words helps turn an expression into a statement that can be examined.

Expression	Reading and meaning
$x \in D$	“x belongs to D”: x is a case admitted in domain D.
$A \subseteq B$	“A is contained in B”: every element of A also belongs to B.
$A \times B$	The set of ordered pairs (a,b), with a in A and b in B; the order of the components matters.
$f: D \to Y$	The function f assigns an output in Y to each input in D.
$g \circ O$	Composition: first apply O, then g. Thus, $(g \circ O)(h)=g(O(h))$.
$\Rightarrow$	“Implies”: under the stated assumptions, the first statement leads to the second.
Σ	The sum of terms in a family. The indices and limits specify which terms are included.
$\int$	Accumulation on a declared scale. Integrating a rate in J/s over time in s yields J.
ΔG_T	In teaching model MD7.2 (section 7.5): the difference $G(t+T)-G(t)$, with numerical G and comparable evaluations. Difference and rate have distinct units; this convention does not redefine all the author’s uses of ΔGT.

A complete reading

If O(h) is the preserved report of a history h and g interprets that report, $q=g \circ O$ states that the answer q can be obtained by applying g to the information O preserved. If two histories produce the same report and require different answers, that recovery is impossible in the given domain. Chapters 4 to 6 prove the statement and explore its consequences.

Reading guide to this demonstration edition

This selection preserves the opening and initial sequence of V9. The acknowledgments are reproduced in full.

Opening of the work · source pages 1–8

Cover, title page, authorship and credits, acknowledgments, author's preface, study guidance, symbol guide and this reading guide.

Chapter 1 · source pages 9–20

Engineering continuity: from the instant to MovieShot. The problem, definitions, examples, three propositions and chapter exercises, in their original order.

Beginning of Chapter 2 · source pages 21–24

Sections 2.1 and 2.2: states, transitions, executions, prefixes and preservation by induction. The excerpt ends at the conclusion of section 2.2.

Selected solutions · source page 25

Original solutions to exercises 1.2 and 1.3.

Editorial bridge to Chapter 7 · source pages 26–27

Example 7.4 and Table 7.1, preserved and contextualized. The supplementary explanations are identified.

References for this selection · source pages 28–29

Only sources cited in the included content; this is not the complete catalogue of the author's work.

How to use the sample

Section, equation, figure and exercise numbers are those of the complete work; pagination has been adapted. Cross-references point to the development in the complete book. The ΔGT guide has been clarified in accordance with section 7.5. Proofs remain conditional on the assumptions in the text; numerical examples are synthetic.

CHAPTER 01

Engineering continuity: from the instant to MovieShot

Engineering continuity begins when preserving operation is no longer enough: the meaning of commitments and transformations must also be preserved.

Objectives. Distinguish present operation from continuity of a commitment; relate the physical execution of a computer to the history that can be observed; define the scientific scope of Structural Frame and MovieShot; examine the scope of an observation presented live.

Prerequisites. Reading simple graphs, arithmetic operations and a willingness to make assumptions explicit. Formal models will be developed progressively.

1.1 The engineering problem: a commitment survives change

A service receives a request, displays a confirmation message and restarts. The interface returns quickly. However, the request does not appear in storage. The equipment is on, the application responds and the obligation undertaken toward the user remains unresolved. This situation distinguishes two questions: does the system work now? Was the commitment presented before the interruption preserved?

Structural Continuity Engineering takes the second question as a central part of its subject. This case will accompany the book because it connects mathematical models, observation, memory, identity and decision-making without depending on a specific product.

In the perspective proposed by Rafael Felippe Voigt, continuity involves the examinable preservation of relevant relationships as computational systems evolve [MAN]. Relevance must be declared. In a request service, it may matter that every confirmation corresponds to a persistent record; in a migration, that each obligation remains associated with the same entity; in an investigation, that the meaning of an old event remains recoverable. The expression continuity gains technical substance when these requirements can be distinguished, checked and, when necessary, refuted.

Definition 1.1 — Continuity obligation

A continuity obligation is a declared property that must remain valid through a specified class of transformations. Its formulation identifies the system, the boundary, the events considered and the condition for fulfilment. In this work, the term organizes the engineering problem; it does not establish an automatic guarantee for any real system.

An organization may choose different obligations. Preserving all data, recovering a function within a specified time and retaining the association between a decision and its reasons are not equivalent requirements. Some can be fulfilled simultaneously; others may conflict with cost, capacity or privacy. The engineer's first task is to make these commitments explicit before condensing them into an indicator. A number does not decide which obligations should exist.

Proposition 1.1 — Present operation does not imply a preserved commitment

The statement that an application is currently available does not, by itself, imply that a prior confirmation has the required persistent record.

Proof by counterexample. Consider three binary variables: q indicates that the application is running, a indicates that confirmation has been issued and d indicates that a persistent record is available. State $(q, a, d)=(1,1,0)$ satisfies present operation and violates the requirement that confirmation imply a record. An application can reach this state by confirming before writing and losing volatile work during an interruption. Therefore, $q=1$ does not imply $a \leq d$. The proof refutes a general implication; it does not assert that every application confirms in this way.

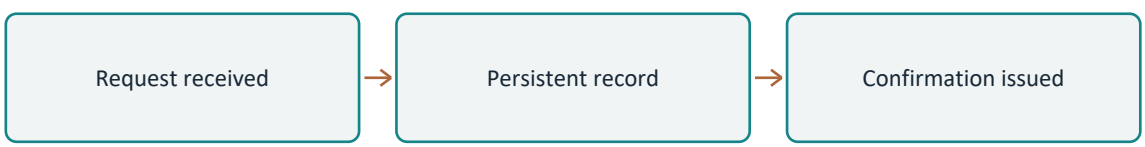


Figure 1.1 — An external commitment depends on internal events. The order and meaning of confirmation are engineering assumptions, not consequences of the appearance of the interface.

The requirement must also specify how long it is to hold and which transformations it covers. Preserving a request during a restart does not demonstrate preservation during a migration, a change in meaning or the loss of a participant. The boundary selects what is considered, and the horizon selects where the obligation is examined. We can start with a small scenario and broaden its assumptions. Rigour lies in tracking that expansion without silently transferring a local guarantee to conditions the proof did not consider.

1.2 The physical computer and our description of it

A computer performs work through physical transformations. Electronic components change states, exchange signals and dissipate energy. Processing, communication and storage depend on material conditions of power supply, temperature and operation. This physical basis matters for continuity: a power interruption can eliminate volatile information; heating can change operating conditions. At the same time, observing a physical effect does not automatically reveal the identity of a request, the meaning of a decision or the validity of an obligation.

Power and energy illustrate the difference between a quantity and its interpretation. Power is a rate of energy transfer; a watt corresponds to a joule per second. Under constant power P for duration T, the energy transferred is E=PT. If power varies, accumulation requires considering the segments or integrating an appropriately defined function. Units follow the operation. A reading in watts is not an amount in joules, just as a completion rate is not the total number of completed requests. In expression 1.1, square brackets indicate the units used to express the quantities; they do not denote their dimensional formulas.

$$E = PT,\ [P] = J/s,\ [E] = J \quad (1.1)$$

Example 1.1 — The same energy, different conditions

In the first teaching scenario, a device transfers 40 W for 20 s: E=40×20=800 J. In the second, it transfers 80 W for 10 s: E=800 J. Equality of energy does not make the powers or durations equal. It also does not determine which instructions were executed, how many requests completed or whether information was preserved. The calculations assume constant power in each scenario; an isolated reading does not demonstrate that constancy.

Choosing the level of description is essential. We can represent voltages in components, memory states, application events or organizational obligations. A model can relate these levels, but we must not silently substitute one for another. A fluctuation in consumption can prompt investigation without identifying its cause. A sequence of records can preserve a logical relationship without directly measuring its electrical manifestation. Each transition between levels requires an assumption and a compatible form of observation.

The book uses this connection between material reality and representation as a discipline of reasoning. It does not deduce a new physical law from utilization graphs or reduce a system's history to an energy curve. Its problem is to understand how observed transformations, records and relationships can support conclusions about continuity. Quantitative studies will explicitly identify which numbers are measurements, which are assumed parameters and which were calculated within a model.

The same process can change the physical condition without changing the external obligation. Replacing a storage device requires work and changes components but can preserve in full the records defined as essential. The reverse is also possible: the machine can retain its appearance and performance while the meaning of a field changes or a historical relationship is lost. Engineering continuity must therefore relate physical state, logical representation and commitment while preserving the differences between these descriptions.

The acquisition of a physical quantity also has a boundary. The power of an entire computer includes components that may not directly participate in the request being investigated. The energy of a time window includes simultaneous activities and background conditions. Dividing that total by the number of completions yields a possible ratio, but interpreting it as energy caused by each request would require controlling the other components. The precision of a dimensional calculation does not

eliminate the attribution problem. Measuring a quantity and explaining its origin are related but distinct tasks.

1.3 Snapshot, Structural Frame and MovieShot

A state snapshot answers local questions. How much is being used? Which version appears to be installed? Is a record present now? These observations are useful. However, a question about evolution requires relating them: did the version change before or after a failure? Does the current record correspond to the earlier confirmation? Does the same indicator name retain its definition? The author's MovieShot proposal organizes observation around these historical relationships [MAN].

History and time already form part of established computing fields. Logs, distributed tracing, temporal databases, state machines and formal specification study sequences, causality and transformations [L02, L78]. The contribution proposed in this book must be examined in that dialogue. Its focus is the continuity of memory, identity, obligations and capacity to evolve. Presenting a distinctive perspective does not require claiming that all earlier computing ignored histories.

Definition 1.2 — Structural Frame and MovieShot

A Structural Frame is a contextualized observation of the system within the declared scope. In addition to values, its meaning may depend on a temporal reference, observed identity, source, collection conditions and relevant relationships. A MovieShot is an organized representation of Frames and the declared relationships between them. The term identifies the author's observational proposal; the sufficiency of this representation depends on the questions and on the coverage actually obtained.

$$\mathcal{M} = (F_0, F_1, \ldots, F_n) \quad (1.2)$$

In expression 1.2, parentheses indicate a sequence: position and repetition matter. A set written only in braces would not necessarily preserve these characteristics. With a single observer, the sequence can follow acquisition order. Among distributed participants, there may be only a partial order. In that case, we must declare the known relationships and not

invent a complete physical ordering from the position of the records. Chapter 2 will formalize this distinction.

A long list does not become a useful MovieShot simply by accumulating items. If units change without notice, identifiers are reused or gaps disappear from the presentation, apparent continuity can combine incompatible observations. The quality of the representation will be examined through properties: which questions can it answer? Under what conditions? Which alternatives remain possible? This observational commitment is more precise than promising a complete reproduction of everything a machine did.

Structural Time in the proposal describes the organization of relevant transformations. A step count can index them, but it is not equivalent to physical duration and does not automatically measure the significance of change. Two systems can pass through the same clock interval with different numbers and types of transformations. To turn that difference into a comparable quantity, criteria for relevance, domain and scale must be chosen. The chapters devoted to quantities will treat this construction as part of scientific investigation.

The term observational unit must not be confused with a minimum physical unit of nature. A Frame can be sufficient for one question and insufficient for another. Its scientific composition must follow the scope: preserving an identity reference may matter in a migration; preserving the relationship between confirmation and writing may matter during an interruption. The idea of a minimal Frame only becomes meaningful after specifying which distinctions must remain. Record size or production frequency does not determine that sufficiency.

1.4 The same present can conceal incompatible histories

Let us return to the confirmed request. History A receives the request, writes the record, confirms and experiences an interruption. History B receives, confirms before writing, then writes and experiences the same interruption. After resumption, both may show confirmation issued and record present. Yet only B passed through an interval in which the confirmation lacked the required support. An identical present does not decide this historical property.

Proposition 1.2 — A final state does not determine every historical property

If two admitted histories end in the same observed state and a property has different answers in those histories, no function of that observed state alone correctly decides the property for both.

Proof. The function receives the same input in both cases and, by the definition of a function, produces the same answer. Since the correct answers differ, it cannot agree with both. For the request, the property is that every confirmation occurred after the required write. Histories A and B described above form the pair that supports the argument. The conclusion is relative to the chosen observation: adding relevant historical information may distinguish the pair.

Example 1.2 — A commitment the final state does not prove

Suppose the final report contains only a=1 and d=1. It is compatible with write→confirm and also with confirm→write. Under the obligation to write before confirming, the first sequence is valid and the second is not. To investigate this obligation, we must preserve some distinction about the relationship between the events. Within the model, the final state establishes the current presence of the record; it does not determine the earlier order.

This impossibility guides an observational design decision: retain the necessary distinctions, not simply more numbers. For some questions, a well-defined historical indicator is enough. Others require relationships between events, versions and entities. Chapter 4 will establish a general criterion for recovering properties. Here, we can already recognize the problem without anticipating a storage architecture.

The argument also rules out an exaggerated conclusion. It does not prove that a state is useless. A sufficiently enriched state can contain memory of the relevant differences. Nor does it require retaining every physical detail. The choice depends on the family of questions the system undertakes to answer. In engineering, a representation is assessed in relation to a purpose and declared conditions, not an abstract promise of completeness.

An obligation can remain true without the service making progress. If there is never a confirmation, the condition confirmation implies a record can hold indefinitely while no user obtains a result. Preservation and progress must therefore be formulated separately. The continuity examined in the

rest of the book will include both relationships that must not be violated and conditions under which the system must continue to provide a useful function.

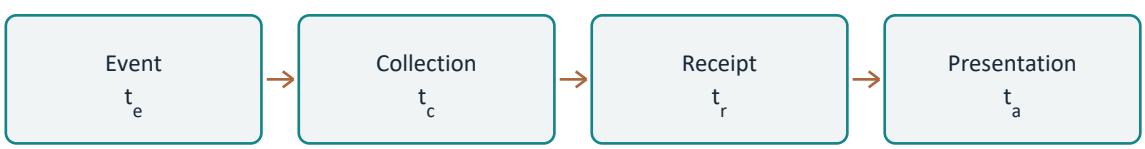
An unknown answer also needs a place in the description. If records are missing between two observations, we must not replace the gap with a convenient history. We can state what the endpoints show and leave compatible alternatives open. This stance does not prevent operational decisions; it allows those responsible for them to know which conclusions depend on an assumption. When new evidence arrives, the argument can be revised without rewriting the nature of the data available at the earlier moment.

1.5 Live observation includes delay, coverage and uncertainty about what happened

The expression live often suggests immediate access to what is happening. A real observation follows a path: the event occurs, an instrument detects it, the information is transported and a presentation makes it available. Each stage has a duration and operating conditions. Updating a dashboard frequently does not eliminate that path. To examine how current a piece of information is, we need to distinguish its instants.

Definition 1.3 — Instants of an observation

t_e is the instant of the event; t_c, of collection; t_r, of receipt; and t_a, of presentation. To subtract these values as durations, they must belong to comparable scales. The age of the information at presentation is $t_a - t_e$. This age does not demonstrate that the state remained unchanged throughout the interval.



The reading time does not replace the event time.

Figure 1.2 — Event, collection, receipt and presentation are distinct stages. Distances in the drawing are illustrative; they are not delay measurements.

Proposition 1.3 — Decomposition of the age at presentation

If the four instants are comparable, the age $t_a - t_e$ is the sum of the intervals event→collection, collection→receipt and receipt→presentation. To interpret each term as a nonnegative duration along the path, the instants must also occur in that order.

$$t_a - t_e = (t_c - t_e) + (t_r - t_c) + (t_a - t_r) \quad (1.3)$$

Proof. Expanding the right-hand side cancels t_c and t_r, leaving $t_a - t_e$. This is an algebraic identity. Its interpretation as three durations requires the stated ordering and comparability. If the values come from different clocks without a known relationship, the numerical identity does not, on its own, provide a valid physical age.

Example 1.3 — The displayed state has an age

An event is placed at 10.00 s, collected at 10.20 s, received at 10.65 s and presented at 10.80 s, on the same scale. Its age is 0.80 s: 0.20+0.45+0.15. A dashboard can be updating now while presenting information about an earlier event. The calculation does not determine what happened in the 0.80 s following the event; that would require further observations.

Coverage and currency are also different. One instrument may quickly present only some events. Another may present a more complete history with greater delay. For one obligation, low latency may be the priority; for another, preservation of relationships and disclosure of gaps. Engineering must make that choice explicit. The instrument's silence must remain distinguishable from the statement that nothing happened.

Times displayed to many decimal places can convey undue confidence. Clock uncertainty, collection delay and the convention for the instant represented continue to exist even when the screen displays microseconds. The next chapter will teach how to obtain duration intervals when errors have declared bounds. The initial question is to preserve the information that makes this calculation possible. A time without source, scale and meaning can be less useful than a simpler timestamp accompanied by its conditions.

1.6 From a research programme to engineering practice

The manifesto provides a direction: study systems through their trajectories and examine what remains useful when they change. Turning this direction into an engineering book requires distinguishing four levels. Definitions establish the objects. Hypotheses delimit models. Proofs deduce consequences from those hypotheses. Experiments examine whether the model and chosen conditions describe an observed domain. Validity at one level does not automatically replace validity at the others. The origins of this programme are presented in the foundational manifesto of Temporal Structural Computing, the monograph on principles and the foundations of the Voigt School [TSC; PRINC; ESC]. These sources document the proposal and its evolution; they do not replace the specific proofs and evaluations in this book.

A study procedure begins by choosing an obligation and a boundary. It then identifies entities, events, temporal references and observations. It formulates a conclusion and searches for an alternative history compatible with the same data that contradicts it. If such an alternative exists, information is missing or an additional assumption is required. If a property is proved within the model, the investigation must still examine whether its premises correspond to the real system.

This book's scientific commitment is to present the mathematics of the proposal in a form open to discussion and verification. Quantities proposed by the author will be accompanied by meaning, domain, units where defined and stage of development. A conceptual relationship will not be treated as an already calibrated instrument. A physical analogy will not be elevated to a universal law simply because it shares a word with another discipline. This distinction allows science to evolve without concealing its open questions.

The examples use synthetic data to make calculations reproducible on paper. When a conclusion requires measurement, the data source and protocol must be specified. The objective is not to prove a product's quality through a classroom example. It is to educate a reader able to recognize a commitment, construct a model, check a deduction and assess the evidence required to transfer it into operation.

In the next chapters, the language of states and events will allow invariants and trajectories to be formulated. Measurement will link changes to durations, accumulations and

uncertainties. Relationships between functions and observations will show which properties remain recoverable. This foundation will serve the central problems of the work: memory, identity, structural quantities, recovery, governance and integrative studies. Continuity will be examined through the relationships engineering can justify throughout change.

The reading path alternates between a concrete problem, a definition, deduction and examination of limits. On the first pass, the reader can follow the question and repeat the calculations. On the second, the reader should check where each assumption enters the proof. On the third, the reader can construct an alternative that violates a condition and observe which conclusion no longer holds. This method brings study closer to engineering practice: understanding a tool includes recognizing its domain, the data it requires and the decision it cannot justify on its own.

Structural continuity is assessed throughout transformations through fulfilment of the declared obligations of memory, identity and integrity, according to an explicitly stated domain, conditions and temporal resolution. This is the working definition examined in Chapter 7. Persistence means remaining according to a criterion; identity recognizes compatibility with a reference; integrity requires valid relationships; recovery returns to acceptable conditions. None of these statements, on its own, establishes the entire continuity commitment.

The proposal engages with established knowledge. Formal methods already address states, invariants and refinement [L02]; dynamic systems relate evolution, inputs and outputs [DYN11]. Temporal databases distinguish the validity and recording of facts [TDB00]. Provenance organizes origin and responsibility [PROV13], while telemetry brings together signals and context [OTEL]. Requirements verification and validation of use already form part of systems engineering [NASA16]. Incomplete data also receive their own treatment in reliability [NREL]. Anomaly detection depends on domain and context [ANOM09]; forecasting requires temporal assessment [HA21]. Agents with memory and reflection have already been studied [GA23]. Statistical information and measurement have their own definitions [IT16; VIM]. The contribution proposed here is to organize historical questions, references and continuity obligations into an auditable argument. This synthesis does not establish priority for each component or superiority over alternatives. Demonstrating additional benefit requires comparisons with approaches equally instrumented for the same questions.

1.E Selected exercises

1.1 | Interpretation. An application restarts, responds again and cannot find a request confirmed before the failure. Formulate two different obligations and state which information would allow each to be examined.

1.2 | Calculation. Compare 50 W for 12 s with 75 W for 8 s. Calculate the energies under constant power and explain two operational properties these totals do not determine.

1.3 | Proof. Construct two histories with the same final state of confirmation and writing, but different orders of those events. Show why the final state does not decide the property writing before confirmation.

1.4 | Investigation. Information is presented without a collection time or an indication of gaps. Propose a question that can still be answered and another whose answer requires additional context. Explain why frequent updates alone do not resolve the second question.

CHAPTER 02

States, events and time

An engineering history must state what can change, which relationships constrain the change and how time enters the conclusion.

Objectives. Construct a model of states, events and executions; prove an invariant and identify an assumption that supports it; relate dependencies, concurrency and duration bounds; separate duration, order and accounting for presence.

Prerequisites. The continuity problem and contextualized observation from Chapter 1; finite sets, functions, inequalities and induction as explained in the text.

2.1 A boundary turns a situation into a model

The confirmed-request problem involves too many components for us to start with the entire machine. We will choose a description with four variables: p indicates volatile work, d a persistent record, a an issued confirmation and q operation. The first three values are binary; q can be on or off. They are not the complete contents of memory. They are the distinctions the model preserves to investigate a specific obligation.

Read the symbols as abbreviations of simple ideas. A tuple is an ordered list; changing the positions of its elements can change its meaning. The symbol × forms combinations, and ⊆ reads “is contained in”. Thus, $R \subseteq X \times E \times X$ selects admitted triples: departure state, event and arrival state. X_0 collects the states accepted as initial. The following formula organizes these choices; the table will show how to use them.

Definition 2.1 — Transition system

A transition system is a tuple $M=(X, E, R, X_0)$: X is the set of states, E the set of events, $R \subseteq X \times E \times X$ a transition relation and $X_0 \subseteq X$ the set of initial states. A transition (x, e, y) permitted by R describes a possible change from x to y with event e. It does not, on its own, assign probability or duration.

$$X=\{0,1\}^3 \times \{\text{on, off}\},\ x=(p, d, a, q),\ X_0=\{(0,0,0,\text{on})\} \quad (2.1)$$

The space has $2\times2\times2\times2=16$ possible descriptions. Some may never be reached under the chosen rules. It is important to separate the value space from the admitted executions. A mathematically representable state can violate the contract; this does not automatically make it impossible in the real system. Assumptions about the initial state and transitions will allow us to prove which descriptions the model reaches.

Event	Condition	Change
receive	q=on	p becomes 1
write	q=on and p=1	d becomes 1
confirm	q=on and d=1	a becomes 1
fail	q=on	p becomes 0; q becomes off
resume	q=off	q becomes on

Table 2.1 — Teaching model of a single request. Components not mentioned remain unchanged; the persistent record is not lost between the admitted transitions.

The table specifies possibilities for study, not an implementation. The same event can occur again when enabled. Within this scope, repetitions refer to the same request; distinguishing multiple requests would require adding identities and a population. The assumption that failure preserves d is decisive and must be checked against the type of storage and failure being studied when the model is applied.

Granularity also matters. Treating an entire write as a transition excludes its intermediate states from the description. If a failure can occur within it and produce a partial result, we must expand the model. Calling a transition atomic means an explicit choice of representation or a commitment demonstrated at another layer; the adjective does not create a material guarantee.

The boundary also defines what we call persistent state. A record recognized by an application can depend on another component whose condition is not in X. If the question requires that dependency, it must be included or the assumed external contract declared. The precision of a small model comes from this

discipline of delimiting scope. Adding variables without a question does not ensure a more useful description.

Set E names events, but relation R determines when they are admitted. The existence of the word confirm does not allow it to be applied in any state. An execution must satisfy each condition at the point where it uses the event. Separating vocabulary from rules avoids a frequent confusion: a history of names can look plausible while omitting the conditions necessary for its validity. To check a history, the reader must follow the successive states rather than merely recognize a familiar sequence of operations.

2.2 Executions, prefixes and invariants

Definition 2.2 — Execution and reachability

A finite execution is a sequence $\rho=(x_0, e_1, x_1,\ldots, e_n, x_n)$, with x_0 initial and every transition belonging to R. A prefix ends at an earlier stage of the same execution. A state is reachable when it is the last state of an execution so defined. The zero-step execution already reaches the initial state.

An execution preserves order and repetition. The events receive, write and confirm do not merely form a set: exchanging their positions can change validity, and repeating one of them can matter to the question. Sequence order represents an execution admitted by the model. Its duration will appear only after temporal references are associated with the events.

The obligation confirmation requires a record can be written $I(x)$: $a \leq d$. Since the values are binary, $a=1$ requires $d=1$. An invariant is a property that holds in every reachable state. The variables need not remain constant; what remains is the truth of the relationship. This distinction allows reasoning about transformation without confusing preservation with immobility.

Proposition 2.1 — Preservation by induction

If I holds in the initial states and every transition from a reachable state in which I is true leads to a state in which I is true, then I is an invariant of M.

Proof. An execution without transitions ends in an initial state, where I holds. Assume I is true at the end of every n-step execution. An

execution of n+1 steps has an n-step prefix; by the induction hypothesis, its last state satisfies I. The assumption about transitions preserves I in the final step. Induction covers every finite number of steps and therefore all reachable states.

Example 2.1 — The role of the confirmation condition

In the initial state, a=d=0. Receive and resume change neither a nor d. Write increases d without increasing a. Confirm requires d=1; fail preserves d and a. Therefore, all transitions preserve a≤d, and Proposition 2.1 proves the invariant. If confirm instead required only p=1, the execution receive→confirm would reach a=1, d=0. This counterexample shows which condition supported the guarantee.

A proof of this kind has two obligations: the base case and preservation. A correct transition does not repair an incorrect initial condition. A correct initial condition does not protect against a rule that destroys the property. A stronger auxiliary relationship is often necessary to prove preservation. Encountering this difficulty can reveal information the model does not yet distinguish.

Safety and progress require different arguments. The condition a≤d prevents confirmation without a record; it does not force the request to complete. The sequence receive→fail→resume can repeat without a write. An obligation that every admitted request receive an answer requires assumptions about failures, scheduling and the environment. Preserving an invariant does not, by itself, prove that the service will continue to fulfil its function.

For an infinite execution, a state property preserved in every finite prefix remains true at each position of that execution. This adds no progress guarantee. The sentence there is always a record when there is a confirmation examines every position; the sentence every request will be answered requires reasoning about the future of the execution. A condition that failures cease or that enabled activities receive service may be essential to the second conclusion. This difference will matter when we call a recovery sufficient: we need to state which obligation it actually restores.

Selected solutions

Chapter 1

The texts below reproduce the solutions from the complete work for the exercises included in this sample.

Exercise 1.2

The products are 50×12=600 J and 75×8=600 J. The unit is the joule because W=J/s. The equality depends on the assumed constant powers. It determines neither individual event times nor the correctness of a confirmation; nor does it imply equal maximum power or duration.

Exercise 1.3

The histories receive→write→confirm and receive→confirm→write can end with confirmation issued and record present. The first respects the obligation of a prior write; the second violates it. A function of the final state alone receives the same input and therefore cannot correctly decide both. The conclusion requires that no other historical information be available to the function.

From the initial problem to formalization

EDITORIAL BRIDGE — EXCERPTS FROM CHAPTER 7

The opening followed a request whose commitment must survive a change. Chapter 7 returns to this question: what must remain recoverable and which relationships must continue to hold? This editorial bridge presents two excerpts from the complete work, with supplementary explanations identified. The example is synthetic; the table organizes conclusions within the model. Neither constitutes a product test.

The commitment during change

Editorial context. Adequate memory means preserving answers to the questions declared in the chosen domain. Identity and integrity require their own criteria. A milestone is an assessment point; checking two endpoints does not prove what happened between them.

Worked example 7.4 — Endpoints do not settle the assessment

A maintenance operation has three milestones. At the first, a request and its policy are related. During migration, the request remains accessible but its policy becomes unavailable. At the third, a copy allows the relationship to be recovered. If the commitment required identifying the policy at all three milestones, the final result does not establish fulfilment at the second. If the commitment allowed unavailability during that period, the conclusion will be different. The family of questions, the access obligation and the schedule must precede the assessment. Reporting the local gap is sufficient; it is not necessary to assign a numerical loss to the entire identity or claim that the incident erased the history.

Editorial interpretation

The example illustrates a distinction: recovering a relationship at the end does not prove that it was available at the required intermediate milestone. Nor does it establish memory loss when the necessary data merely ceased to reach the observer. The contract must separate preserved information, ability to access it and available evidence. Section 7.8 formalizes these commitments; its Proposition 7.5 depends on its own assumptions about record recovery and semantic transport.

Source of the excerpt: section 7.7 of the complete work, Worked example 7.4. The explanations surrounding the example were added in this demonstration draft.

Which conclusion does the evidence support?

EDITORIAL CONTEXT FOR TABLE 7.1

Section 7.8 fixes a nonempty domain H of admissible histories, the milestones and the contract. $C^{+}_{H}(h)$ is the destination commitment: it combines global adequacy of memory to the questions with the identity and integrity obligations for history h. It is a logical predicate, not a numerical score.

An observation V produces evidence e. H_e collects the histories in H compatible with it. In the table, $C^{+}H$ abbreviates $C^{+}_{H}(h)$. Reducing H_e does not establish global memory adequacy that failed in domain H.

Cases compatible with the evidence	Conclusion within the model
H_e nonempty; all histories fulfil $C^{+}H$	Fulfilment supported within the chosen scope.
H_e nonempty; all histories violate $C^{+}H$	Violation supported within the chosen scope.
H_e contains fulfilment and violation	Indeterminacy: the observation does not decide.
H_e empty	No compatible case: review the domain, observation model or evidence.

Table 7.1 — Assessment states, keeping the domain and commitment fixed. “Indeterminate” is not a third truth value of the actual property; an empty set does not authorize concluding fulfilment by vacuous universal satisfaction.

Scope of the classification

The property in a complete history is true or false; indeterminacy describes the assessment. Applying the table to the real world requires justifying the domain and fidelity of the observation. Having no compatible history requires reviewing the model or evidence. The judgement is limited to the declared milestones; it guarantees neither unobserved intervals nor a timely response.

Source of the excerpt: section 7.8 of the complete work, Table 7.1. Its function is to distinguish truth, evidence and limits of inference. This bridge does not replace the definitions, Proposition 7.5 or its proof.

References for this demonstration edition

Sources actually cited in the selected excerpts. The author's repository publications, institutional documents, technical literature and product documentation differ in nature. A DOI identifies a record and does not constitute proof or independent validation.

[MAN] VOIGT, Rafael Felippe. Manifesto da Engenharia da Continuidade Estrutural: A Fundação da Engenharia da Continuidade Computacional. Version 1.0. Brazil, 2026. Author's manifesto.
https://www.voigtcore.com.br/downloads/documentos/manifesto-engenharia-continuidade-estrutural.pdf

[L02] LAMPORT, Leslie. Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers. Addison-Wesley Professional, 2002. Technical book; reference for states, actions, invariants and composition. Accessed 29 October 2026.
https://lamport.azurewebsites.net/tla/book.html

[L78] LAMPORT, Leslie. Time, Clocks, and the Ordering of Events in a Distributed System. Communications of the ACM, v.21, n.7, p.558-565, 1978. Accessed 29 October 2026.
https://www.microsoft.com/en-us/research/publication/time-clocks-ordering-events-distributed-system/

[DYN11] MIT OPENCOURSEWARE. Dynamic Systems and Control. 6.241J, 2011. Emilio Frazzoli and Munther Dahleh. Accessed 29 October 2026.
https://ocw.mit.edu/courses/6-241j-dynamic-systems-and-control-spring-2011/

[TDB00] SNODGRASS, Richard T. Developing Time-Oriented Database Applications in SQL. Morgan Kaufmann, 2000. Accessed 29 October 2026.
https://www2.cs.arizona.edu/~rts/tdbbook.pdf

[PROV13] W3C. PROV-DM: The PROV Data Model. Recommendation, 2013. Accessed 29 October 2026.
https://www.w3.org/TR/prov-dm/

[OTEL] OPENTELEMETRY. Observability primer. Official documentation. Accessed 29 October 2026.
https://opentelemetry.io/docs/concepts/observability-primer/

[NASA16] NASA. Systems Engineering Handbook. NASA/SP-2016-6105 Rev. 2. Section 5, Product Realization. Accessed 29 October 2026.
https://www.nasa.gov/reference/5-0-product-realization/

[NREL] NIST/SEMATECH. e-Handbook of Statistical Methods. Section 8.1.3.1, Censoring. Accessed 29 October 2026.
https://www.itl.nist.gov/div898/handbook/apr/section1/apr131.htm

[ANOM09] CHANDOLA, Varun; BANERJEE, Arindam; KUMAR, Vipin. Anomaly Detection: A Survey. ACM Computing Surveys, 41(3), art. 15, 2009. DOI 10.1145/1541880.1541882. Accessed 29 October 2026.
https://arindam.cs.illinois.edu/papers/09/anomaly.pdf

[GA23] PARK, Joon Sung et al. Generative Agents: Interactive Simulacra of Human Behavior. UIST 2023. DOI 10.1145/3586183.3606763. Accessed 29 October 2026.
https://research.google/pubs/generative-agents-interactive-simulacra-of-human-behavior/

[TSC] VOIGT, Rafael Felippe. TEMPORAL STRUCTURAL COMPUTING - Foundational Manifesto of a New Computational Category. Zenodo, 2026. Author's repository publication.
https://doi.org/10.5281/zenodo.20114016

[PRINC] VOIGT, Rafael Felippe. Fundamental Principles of Temporal Structural Computing - Establishing the Scientific Foundations of a New Computational Discipline. Zenodo, 2026. Author's repository publication.
https://doi.org/10.5281/zenodo.21268037

[ESC] VOIGT, Rafael Felippe. Foundations of the Voigt School of Temporal Structural Computing. Zenodo, 2026. Author's repository publication.
https://doi.org/10.5281/zenodo.20208382

[IT16] POLYANSKIY, Yury; WU, Yihong. Lecture Notes on Information Theory. MIT OpenCourseWare, 6.441, 2016. Entropy and conditional distributions. Accessed 29 October 2026.
https://ocw.mit.edu/courses/6-441-information-theory-spring-2016/pages/lecture-notes/

[VIM] JCGM. International vocabulary of metrology - Basic and general concepts and associated terms (VIM). 3rd ed. JCGM 200:2012. Entries 1.1 and 1.26. Accessed 29 October 2026.
https://doi.org/10.59161/JCGM200-2012

[HA21] HYNDMAN, Rob J.; ATHANASOPOULOS, George. Forecasting: Principles and Practice. 3rd ed. OTexts, 2021. Section 5.10, Time series cross-validation. Reference for assessment in temporal order; the data and alert matching in this work are for teaching. Accessed 29 October 2026.
https://otexts.com/fpp3/toolbox.html#time-series-cross-validation

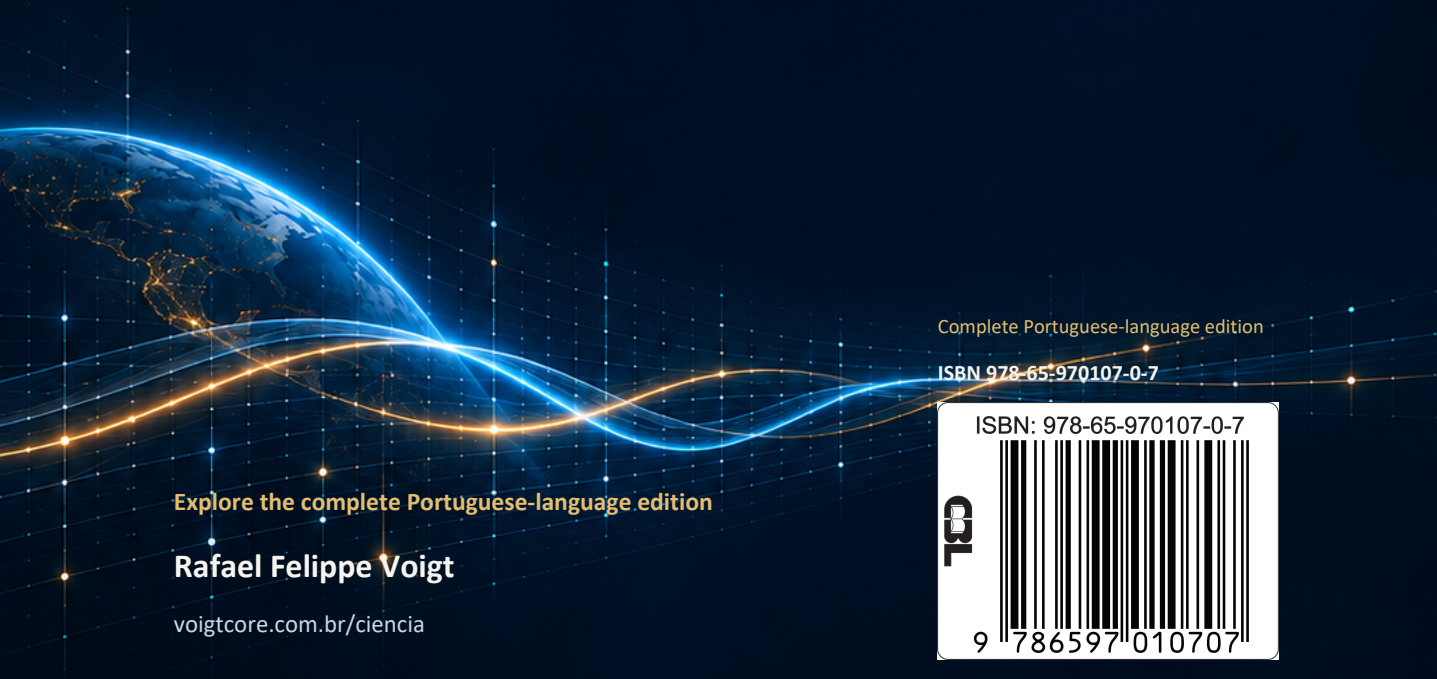
FREE DEMONSTRATION SAMPLE | 2026

What must endure when a system changes?

A system can keep running while losing relationships essential to its history. How can this loss be recognized? What must endure when components, rules and models change?

This work presents the proposal of Structural Continuity Engineering and its foundations in Temporal Structural Computing. It relates states, events and trajectories to the preservation of memory, identity, integrity and obligations.

Definitions, propositions, proofs, worked examples and exercises provide a path for researchers, engineers, teachers and students. The studies address computational systems, infrastructure and artificial intelligence, including Accumulated Digital Capital. Readers learn to examine what has been preserved, what has been violated and what the evidence does not yet allow them to conclude.



Explore the complete Portuguese-language edition

Rafael Felippe Voigt

voigtcore.com.br/ciencia

Complete Portuguese-language edition

ISBN 978-65-970107-0-7

ISBN: 978-65-970107-0-7

9 786597 010707