

Can GPT-4 Think Computationally about Digital Archival Practices? – Part 3

William Underwood
College of Information
University of Maryland
College Park, USA
underwod@umd.edu

Joan Gage
Campbell Elementary
Fulton County Schools
Union City, USA
joangage.jg@gmail.com

Abstract— The flood of digital records into 21st century archives has made it essential to develop new digital archival methods to replace those used for traditional paper documents, film, and tapes. Additionally, it has become necessary to integrate computational thinking skills into the graduate training of upcoming archival professionals. This paper presents findings from investigations into the computational thinking abilities of GPT-4o regarding its understanding of digital archival practices. GPT-4o demonstrates knowledge of computational modeling and simulation thinking techniques. It also exhibits familiarity with PC and MS-DOS emulation, the simulation of costs and risks associated with archival media selections, the evaluation of the empirical validity of preservation models, and the creation of a model to evaluate the trustworthiness of an archival system.

Keywords—computational thinking, GPT-4, modeling and simulation, large language models, MLIS education

I. INTRODUCTION

The computer revolution has significantly influenced how organizations manage, archive, and utilize their records. The shift from physical records like paper documents, printed images, film, video tapes, catalogs, and cassette tapes to their digital equivalents has created a necessity to automate the processes for handling, storing, and accessing these digital documents. The internet has resulted in not only a surge in the volume of records but also heightened risks to the security and privacy of digital information. Archival techniques previously employed for paper, film, or tape records now need to be adapted for effective management of digital files using computers. Graduate students in Archival Studies programs must acquire computational thinking skills to undertake tasks related to digital archiving.

Generative Pre-trained Transformer 4 (GPT-4) is a large language model (LLM) developed by OpenAI. LLMs use a deep neural network architecture known as a "decoder-only transformer" [1]. GPT-4 was pre-trained on an extensive corpus of text, including millions of web pages, tens of thousands of books, the entirety of Wikipedia, and code in languages like Python, C, and Java. It can generate responses on a broad range of topics, including digital archives.

David Weintrop has proposed a taxonomy-based definition of computational thinking, organizing 22 computational thinking practices into four main categories: data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices [2] (Fig. 1).

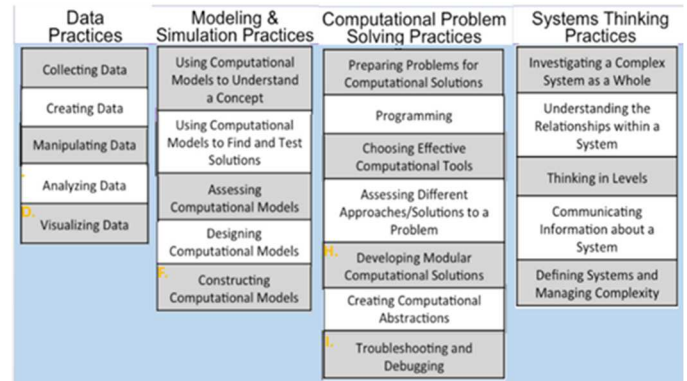


Fig. 1. Computational Thinking in Mathematics and Science Taxonomy

This taxonomy was developed as part of an initiative to integrate computational thinking into high school science and mathematics curricula (STEM).

Underwood and Gage are exploring the kinds of knowledge of digital archival practices and the kinds of knowledge of computational thinking practice that GPT-4 has learned. The purpose of this exploration is to determine whether the GPT-4 might be able to support training in computational thinking as regards digital archival practices. Underwood and Gage [3, 4] demonstrated that ChatGPT-4 possesses knowledge of archival data practices, archival computational problem solving practices and archival system thinking practices. The following sections will show that ChatGPT-4o can also think computationally about archival modeling and simulation practices.

II. CONVERSATIONS WITH CHATGPT ABOUT DIGITAL ARCHIVAL PRACTICES INVOLVING CT MODELING AND SIMULATION PRACTICES

"In mathematics and science, models can include flowcharts, diagrams, equations, chemical formula, computer simulations, and even physical models. By definition, models are simplifications of reality that foreground certain features of a phenomenon while approximating or ignoring other features. By computational models, we refer to non-static representations of phenomena that can be simulated by a computer [2]."

This section discusses an investigation of GPT-4o's ability "to think computationally" about modeling and simulation practices applied to digital archival practices. GPT-4o (o for omni) is a sophisticated large language model, capable of

responding to prompts across text, audio and images in real-time. ChatGPT is the application (the ‘chatbox’) that uses the GPT-4o model to provide human-like, conversational responses to users. While people often use the terms interchangeably, technically, the user is talking to the ChatGPT interface that is running on the GPT-4o model.

First, a modeling and simulation practice is described. Then a digital archival task requiring archival knowledge and modeling and simulation knowledge is requested of ChatGPT. This is followed by a response from ChatGPT. ChatGPT may also be asked to apply this knowledge to some digital archival material or program code. GPT-4o’s performance related to the modeling and simulation practice is then evaluated with regard to the performance criteria of that modeling and simulation practice, and with regard to its knowledge of the digital archival task. ChatGPT’s answer is also verified by checking it against the related literature. In the following conversations, “JO” is the nickname given by ChatGPT to “Joan”, one of the authors of this paper who is asking questions or requesting actions of ChatGPT.

A. Using Computational Models to Understand a Concept

“Computational models that demonstrate specific ideas or phenomena can serve as powerful learning tools. Students can use computational models to deepen their understanding of mathematical and scientific concepts. ... Students who have mastered this practice will be able to advance their own understanding of a concept by interacting with a computational model that demonstrates the concept [2].”

JO: We have some computer games that could be played in MS-DOS on IBM PCs. They will not operate on our current Windows System. Why not? How can we get them to operate again?

ChatGPT: Excellent question — and one that many archivists, collectors, and preservationists run into. Here is a clear explanation:

Why the old MS-DOS games won’t run on modern Windows systems

1. **Different operating systems:**
MS-DOS (used on early IBM PCs) is a 16-bit operating system. Modern versions of Windows (especially 64-bit versions like Windows 10 and 11) no longer include the ability to run 16-bit programs.
2. **Different hardware:**
Old games were designed to talk directly to hardware like Sound Blaster cards, VGA displays, and floppy drives. Modern PCs use completely different hardware interfaces and drivers, so the old software can’t “see” what it expects.
3. **Timing and speed issues:**
MS-DOS programs depended on the CPU’s speed for timing (e.g., game loops). A modern processor runs thousands of times faster, so even if you could run the code, the game might run uncontrollably fast.
4. **File system and memory model differences:**
MS-DOS used FAT file systems and had strict memory limitations (640K conventional memory).

Modern systems handle files and memory very differently.

How to make them work again

1. **Use a DOS emulator (best option):**
 - **Install DOSBox** — free, opensource, and designed exactly for this purpose.
 - It recreates a full DOS environment, including old-style graphics, sound, and input.
 - Works on Windows, macOS, and Linux.
 - You can “mount” a folder on your computer as a virtual C: drive and run your games as if it were 1992 again.
 - **Use a virtual machine (advanced option):** Install a program such as **VirtualBox** or **VMware Workstation**.
 - Set up a virtual computer running **FreeDOS** (a modern DOS-compatible OS).
 - Then install your games inside the virtual machine.
2. **Buy repackaged versions:**
 - Some classic games are sold on platforms like **GOG.com**, which bundle them with DOSBox and make them run automatically on modern Windows.
3. **For archival/preservation purposes:**
 - Keep original disk images (e.g., .img or .iso files).
 - Record hardware requirements, version numbers, and any copy protection details.
 - Use an emulator (like DOSBox or PCem) for access, not direct Windows execution.

Would you like me to give you step-by-step instructions for setting up DOSBox and running one of your game?

JO: Yes

Joan followed the step-by-step instructions given by ChatGPT for setting up DOSBox and running Fig. 2 shows a screenshot from the game.

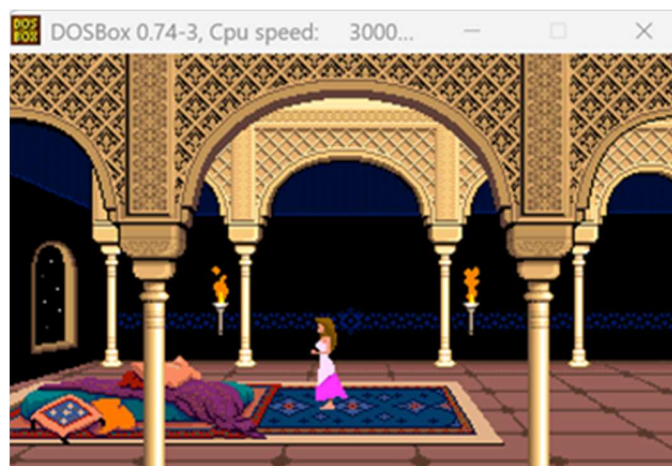


Fig. 2. In-game screenshot of Prince of Persia running successfully within the DOSBox emulation environment.

Evaluation: GPT-4o explains why a DOS emulator or virtual machine is needed to run games that previously operated in PC hardware that used the DOS operating system. Furthermore, it explains how to use the DOSBox emulator to run them. DOSBox is a computational model, in particular an emulator of PC hardware and the Disk Operating System (DOS). GPT-4o's advice enables a student to understand the concept of an emulator by interacting with an emulator. GPT-4o shows that it knows what an emulator is, when one is needed and how to use one. The accuracy of the information provided by ChatGPT was verified by following its instructions and by comparing the information it provided to that in the *DOSBox Reference Guide* [5]

B. Using Computational Models to Find and Test Solutions

“Computational models can also be used to test hypotheses and discover solutions to problems. They make it possible to test many different solutions quickly, easily, and inexpensively before committing to a specific approach....Students who have mastered this practice will be able to find, test and justify the use of a particular solution through the use of a computational model as well as to be able to apply the information gained through using the model when appropriate [2].”

JO: I have millions of digital files that must be archived and preserved long term. I'm not sure what the most economical and reliable storage media might be. I understand that the UC Santa Cruz SSRC simulator might help me make that decision. Can you tell me how it works?

ChatGPT: The **UC Santa Cruz SSRC simulator** is a research tool developed by the UCSC Storage Systems Research Center. It is **not a product that stores data**, but rather a **modeling and simulation environment that lets you predict how different storage technologies will behave over long periods of time**.

Think of it as a “virtual test lab” where you can plug in different storage media options and see how they compare in cost, reliability, failure rates, refresh cycles, and long-term preservation outcomes.

Below is a clear breakdown of what it does and how it helps someone with a very large digital archive like yours.

What the SSRC Simulator Does

1. Models Long-Term Storage Reliability

You can simulate:

- HDD vs SSD vs tape vs optical systems
- Different drive vendors
- Batch failure probabilities
- Media decay over time
- Error rates (bit rot)
- How often rebuilds will occur
- Probability of catastrophic loss under different configurations

2. Allows You to Test Storage Architectures Before Buying Anything

You can model:

- RAID 5/6/7
- Erasure coding (e.g., Reed–Solomon)
- Replication (2 copies, 3 copies, geo-replication)
- Hybrid systems (disk + tape, etc.)

This helps answer questions like:

- “Is three-copy local storage safer than one tape copy plus cloud redundancy?”
- “Does RAID actually help, or should you use object storage with erasure coding?”

3. Produces Cost-vs-Reliability Curves

You can input:

- Cost of media (HDD, LTO tape, SSD, cloud)
- Cost of replacements
- Staff labor costs
- Energy and cooling
- Refresh cycles (every X years)

Then it outputs:

- **Total cost of ownership (TCO)**
- **Cost per terabyte per year**
- **Failure probabilities over 5, 10, 20 years**

4. Lets You Run “What-If” Scenarios

Examples of scenarios you can simulate:

- What if a batch of drives has a 5 percent higher failure rate?
- What if refresh cycles slip by a year?
- What if your collection grows by 20 percent annually?
- What if an entire shelf or rack overheats?

5. Helps Compare Cloud vs On-Premise Preservation

Cloud services (AWS Glacier, Google Archive, Azure Archive) have:

- Retrieval fees
- Write costs
- Read-after-write delays
- Undocumented internal redundancy

The SSRC simulator lets you treat cloud as a “virtual storage device” and compare it to:

- LTO-9 tape libraries
- Local disk arrays
- Distributed server clusters

If You Want, I Can Help You Use It

Evaluation: This application is a computational model that allows one to simulate cost, failure rate and refresh intervals enabling one to discover a solution to the problem of finding economical and reliable storage media for archival preservation. GPT-4o shows that it knows how the simulation model works and can actually use the simulation model. The accuracy of ChatGPT's answer was verified by comparing it to the description of the features of the SSRC simulator.[6].

C. Assessing Computational Models

“A key practice in using a computational model effectively is to understand how the model relates to the phenomenon being represented. This understanding is guided by a variety of questions including: Which aspects of the phenomenon have been faithfully modeled and which aspects have been simplified or ignored? What assumptions have the creators of the model made about the world and how do those assumptions affect its behavior? What layers of abstraction have been built into the model itself and how do these abstractions shape the fidelity of the model? ...Students who have mastered this practice will be able to articulate the similarities and differences between a computational model and the phenomena that it is modeling, this includes raising issues of threats to validity as well as identifying assumptions built into the model [2].”

JO: The InterPARES Preservation model was developed using the IDEF0 Modeling Methodology. At the most abstract level, the IDEF0 context diagram represents the problem of preserving authentic electronic records. The graphical notation of IDEF0 is used as a method for communicating the important aspects of a preservation system. At the lowest level of the model, there are activity definitions that specify the transformation of inputs to outputs. Furthermore, a data dictionary defines the data elements of the inputs and outputs of the lowest level activities. Is there a method that I can use to assess the validity of this model?

ChatGPT: Excellent question — and very much in line with the **InterPARES (International Research on Permanent Authentic Records in Electronic Systems)** methodology. You’re correct that the **InterPARES Preservation Model** was developed using the **IDEF0 (Integrated Definition for Function Modeling)** method — a structured modeling approach used to formally describe functions, their inputs, outputs, controls, and mechanisms. Since the model is representational and process-based, assessing its **validity** means testing whether the model *accurately represents* the real-world functions and relationships it claims to capture — namely, the processes needed to preserve *authentic* electronic records over time. Here’s how you can systematically assess its validity.

1. Conceptual Framework for Model Validation

You can use **model validation frameworks** from systems engineering and information modeling. The most widely accepted for IDEF0 and similar functional models include:

Validation Type	Key Question	Method
Conceptual Validity	Does the model logically and completely represent the preservation domain?	Expert review, consistency checks against preservation theory (e.g., InterPARES Authenticity Task Force principles)

Validation Type	Key Question	Method
Operational Validity	Does the model produce outputs consistent with real archival practice?	Compare modeled workflows to actual archival workflows (e.g., NAR ERA, OAIS implementations)
Structural Validity	Are the inputs, controls, mechanisms, and outputs correctly defined and logically connected?	IDEF0 structure audit: check for completeness, correct decomposition, and no orphaned flows
Data Validity	Are the data dictionary elements accurate and consistent with metadata standards?	Crosswalk against PREMIS, METS, or ISO 23081 metadata schemas
Empirical Validity	Can model predictions or transformations be demonstrated in real systems?	Simulation or case study (e.g., applying the model to an existing repository and tracking outcomes)

2. Methods You Can Use in Practice

E. Empirical Case Study Validation

To test practical validity:

1. Select a digital repository (e.g., Emory University’s repository or a government archive).
2. Apply the IDEF0 preservation model to describe its workflow.
3. Compare:
 - Activities predicted by the model vs. actual repository actions.
 - Data elements vs. actual metadata schemas in use.
 - Transformation fidelity (input → output correctness).

Discrepancies highlight where the model may lack fidelity or overgeneralize.

In Summary

You can assess the validity of the **InterPARES IDEF0 Preservation Model** through a structured combination of:

1. **IDEF0 internal verification** (formal model integrity)
2. **OAIS functional crosswalk** (conceptual completeness)
3. **Metadata alignment** (semantic accuracy)
4. **Expert review** (face and content validity)
5. **Empirical testing** (operational fidelity) This mirrors how InterPARES Phase 2 validated its own models — combining **logical, theoretical, and empirical** approaches.

Evaluation: GPT-4o identified five types of model validity and described methods for establishing each of the types of validity. Only the method for establishing empirical validity is shown. It is the method that validates the IDEF0 process model using simulation data and workflows from a case study of an actual digital archive. The simulation involves a manual walkthrough of the process model and its metadata using data from the case study. GPT-4o also shows that it knows that the empirical validity of the model (how well the model relates to the phenomena that it represents) depends on simulating the IDEF0 model using case study data from an actual digital archive. ChatGPT's description is consistent with an actual walkthrough of the InterPARES Preservation model performed under the guidance of one of the authors of this paper [7].

D. Designing Computational Models

“The process of designing a model involves making technological, methodological, and conceptual decisions. There are many reasons that might motivate designing a computational model, including wanting to better understand a phenomenon under investigation, to test out a hypothesis, or to communicate an idea or principle to others in a dynamic, interactive way. ...Students who have mastered this practice will be able to design a computational model, a process that includes defining the components of the model, describing how they interact, deciding what data will be produced by the model, articulating being made by the proposed model, and understanding what conclusions can be drawn from the model. [2].”

JO: Can you explain how to create a Bayesian Belief Network from a Technical Requirements document?

ChatGPT: Yes — this is an excellent and advanced question that sits at the intersection of **systems engineering, probabilistic modeling, and knowledge representation**. Let's break it down step by step so you can see how a **Bayesian Belief Network (BBN)** can be systematically derived from a **Technical Requirements document**.

What You're Doing in Essence

A Technical Requirements document describes what a system must do, under what conditions, and how performance or compliance will be measured.

A **Bayesian Belief Network (BBN)**, on the other hand, is a **graphical probabilistic model** that represents *uncertainty* and *causal relationships* between variables (nodes) using conditional probabilities (edges).

Goal:

Transform deterministic requirements into a probabilistic model that captures how *observed evidence* (e.g., test results, sensor readings, design parameters) affects belief in whether the requirements are being met.

Step 1. Extract Key Entities from the Technical Requirements

Go through each requirement and identify:

Requirement Component	Example
Condition/Input	Environmental temperature, voltage supply, operator skill
Process/Function	Data encryption, cooling system operation
Output/Performance Criterion	Throughput ≥ 10 Mbps, power consumption ≤ 20 W
Verification Method	Test, simulation, analysis, inspection

These become **candidate nodes** in your network.

Step 2. Define Node Types

Each **requirement** can be broken into types of nodes:

1. **Input nodes:** conditions or design parameters (usually independent).
2. **Intermediate nodes:** subsystem behaviors or component states.
3. **Output nodes:** success/failure or compliance with specific requirements.
4. **Evidence nodes:** measurable data (test outcomes, inspection results).

Step 3. Identify Causal Relationships

Read the “*shall*”, “*must*”, and “*if...then...*” statements in the document carefully.

These express *causal dependencies*.

Example:

“If the ambient temperature exceeds 50°C, then the system shall reduce processor speed to maintain safe operation.”

This maps to:

Temperature → Processor Speed → Safe Operation

These arrows represent **conditional dependencies** between variables.

Step 4. Construct the Network Structure (Graph)

1. Each variable = one **node**.
2. Each causal or conditional dependency = one **directed edge** (arrow).
3. The structure should be **acyclic** (no feedback loops).

You can sketch this using:

- Tools like **GeNIe/SMILE**, **Netica**, **Hugin**, or **BayesFusion**.
- Or in Python using pgmpy or pomegranate.

Step 5. Quantify the Relationships (Conditional Probability Tables)

For each node:

- If it has *no parents*, assign a **prior probability**. (e.g., *Probability that Temperature > 50°C = 0.2*)
- If it has *parents*, define a **Conditional Probability Table (CPT)**.

You can derive these probabilities from:

- Test data
- Engineering judgment
- Expert elicitation
- Historical reliability statistics

Step 6. Validate with Stakeholders and Experts

- Review the network structure: Do dependencies match engineering logic?
- Review CPTs: Are probabilities realistic and justified?
- Use *what-if* scenarios to test the model's behavior under different inputs.

Step 7. Use the BBN for Inference

Once built, you can:

- Input **evidence** (e.g., a failed sensor test)
- Observe how belief propagates through the network
- Quantify the likelihood that higher-level requirements or baseline goals are being met

JO Loaded the following to ChatGPT:

The Authenticity Task Force of the InterPARES Project investigated the essential attributes of authentic digital records and the technological and procedural factors impacting their authenticity. They formulated the conceptual requirements for presuming the authenticity of the digital records maintained by a records creator, which they termed Benchmark Requirements. The Benchmark Requirements draw specifically on the notion of a trusted record-keeping system. They also formulated the conceptual requirements for presuming the authenticity of the digital records maintained by a records preserver, which they termed Baseline Requirements. The Baseline Requirements draw specifically on the notion of a trusted archival system.

The Baseline Requirements are shown in the Table below.

To attest to the authenticity of copies of inactive electronic records, the preserver should be able to demonstrate that:

REQUIREMENT B.1 Controls over Records Transfer, Maintenance, and Reproduction	the procedures and system(s) used to transfer records to the archival institution or program, maintain them, and reproduce them embody adequate and effective controls to guarantee the records' identity and integrity, and specifically that B.1.a Unbroken custody of the records is maintained; B.1.b Security and control procedures are implemented and monitored; and B.1.c The content of the record remains unchanged after reproduction;
REQUIREMENT B.2 Documentation of Production Process and Its Effects	the activity of reproduction has been documented, and that this documentation includes B.2.a The date of the records' reproduction and the name of the responsible person; B.2.b The relationship between the records acquired from the creator and the copies produced by the preserver; B.2.c The impact of the reproduction process on their form, content, accessibility and use; and B.2.d In those cases where a copy of a record is known not to fully and faithfully reproduce the elements expressing its identity and integrity, such information has been documented by the preserver, and this documentation is readily accessible to the user;
REQUIREMENT B.3 Archival Description	The archival description of the fonds containing the electronic records includes—in addition to information about the records' juridical administrative, provenancial, procedural, and documentary contexts—information about changes the electronic records of the creator have undergone since they were first created.

The Authenticity Task Force formulated a method for assessing a presumption of authenticity of records based on the degrees of belief that the Benchmark Requirements are supported by evidence.

“A presumption of authenticity will be based upon the number of requirements that have been met and the degree to which each has been met. The requirements are, therefore, cumulative: the higher the number of satisfied requirements, and the greater the degree to which an individual requirement has been satisfied, the stronger the presumption of authenticity.”

Although the method of assessment is expressed in simple terms, there are substantial pitfalls inherent in subjective probability assessment due to psychological biases and common misunderstandings of probabilistic reasoning. Furthermore, the conditional dependencies between requirements and between the evidence needed to conclude that a requirement is met can be quite complex. The Bayesian approach to reasoning under uncertainty is an approach to reasoning with degrees of belief while dealing with the complexity of conditional dependencies. Combined with Bayesian Belief Networks, it can also expose and overcome some of the common psychological biases and fallacies in reasoning due to misunderstanding of probability.

JO: How can a Bayesian Belief Network be designed to represent the Baseline Requirements?

ChatGPT: Great question. A Bayesian Belief Network (BBN) is a natural fit because the InterPARES **Baseline Requirements** are cumulative and evidence-driven, and the dependencies among their sub-requirements are not purely linear. Below is a practical design you can implement in Hugin, GeNIe/SMILE, Netica, or bnlearn.

1) Map the Baseline Requirements into a DAG

Create a top node:

Presume Authentic (Baseline), a boolean node that expresses the posterior probability that the preserver can attest to authenticity under the Baseline Requirements. Its parents are B.1, B.2, and B.3, which reflect the InterPARES Baseline Requirements text.

Decompose each requirement into children that match the InterPARES clauses:

B.1 Controls over Transfer, Maintenance, Reproduction

Parents: **B.1a Unbroken Custody**, **B.1b Security & Control**, **B.1c Content Unchanged After Reproduction**.

Evidence nodes under each parent:

- *ChainOfCustodyLogsQuality* → informs **B.1a**
- *StorageAccessControls* and *AuditTrailEffectiveness* → inform **B.1b**
- *FixityBeforeAfterRepro*,
BitstreamComparisonMethod → inform **B.1c**

B.2 Documentation of Reproduction & Effects

Parents: **B.2a Date & Responsible Person**, **B.2b Source-to-Copy Relationship**, **B.2c Impact on Form/Content/Access/Use**, **B.2d Known Deviations Documented & Exposed**.

Evidence nodes:

- *ReproEventMetadataCompleteness* (dates, agent) → **B.2a**
- *DerivationLinkage/ProvenanceGraph* → **B.2b**

- *ChangeLogCoverage* and *SignificantPropertiesAssessment* → **B.2c**
- *DeviationDisclosurePractice* and *UserVisibilityOfNotes* → **B.2d**

B.3 Archival Description

Parent evidence nodes:

- *DescriptionIncludesContext* (juridical, provenancial, procedural)
- *DescriptionCapturesChangesSinceCreation* (versioning, transformations)

These nodes mirror the Baseline Requirements statements so your graph structure is auditable back to the policy text.

2) Encode dependencies you actually see in practice

Add arcs where satisfaction of one clause informs or conditions another:

Security & Control (B.1b) → Unbroken Custody (B.1a), because strong access control and monitoring increase confidence in custody continuity.

Documentation of Reproduction (B.2) → Content Unchanged (B.1c), because rich repro documentation plus pre/post fixity strengthens belief that content is unchanged.

Archival Description (B.3) → B.2c/B.2d, since descriptive notes can corroborate impacts and deviations.

These arcs prevent double counting by channeling shared evidence through the right parents instead of redundantly boosting all children.

Due to space limitations, the following steps not shown:

3) Choose gate functions for the requirement nodes

4) Model evidence quality, not only presence

5) Set priors through structured elicitation

6) Outputs you will use

7) Validation and governance

- **Face and content validity:** walk the graph with records experts to confirm structure plausibility.
- **Crosswalk:** keep a mapping table from every node to the exact clause in B.1, B.2, B.3 to maintain traceability to policy text.
- **Empirical validity:** run the BBN on known collections, compare outcomes to independent audit results, then tune CPTs.
- **Change control:** version the network with a data dictionary so every node definition, state set, and CPT source is auditable alongside your archival policies.

Fig. 3 shows the BBN graph structure for the Baseline Requirements. It was created by GPT-4o.

\

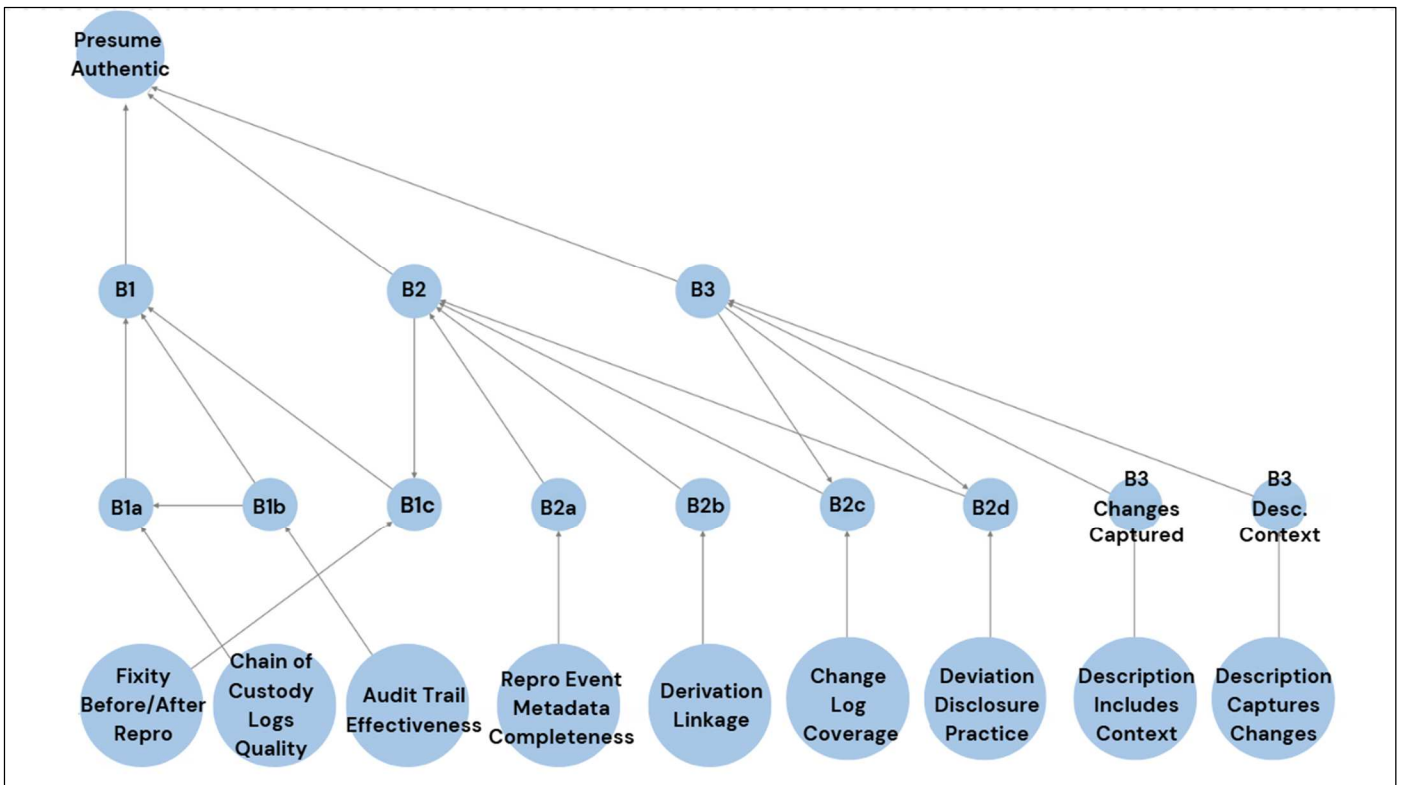


Fig. 3. Bayesian Belief Network Graph Structure for the Baseline Requirements

Evaluation: GPT-4o was asked to explain how to create a Bayesian Belief Network from a Technical Requirements document. Its answer showed that it knew how to do so. It was then asked to design a BBN for the Baseline Requirements. It did so. ChatGPT’s construction of BBNs from formal requirements follows well-established methods in probabilistic modeling [8]. The design process described by ChatGPT is consistent with foundational treatments of BBNs.

ChatGPT showed not only how to create a BBN but was capable of doing so. GPT-4o cannot actually create a BBN on its own. It uses its natural language processing capabilities to translate human expressed requirements into the structured components of a BBN. It learned to do so by being trained on many examples of constructions of BBNs.

In the explanation of how to create a BBN, it does more than explain how to design a BBN. It actually begins to explain how to construct the BBN by describing how to use GeNIe to create a digital graphical representation of the BBN for the Baseline Requirements. In the next section, GPT-4o is asked to describe in greater detail how to construct the BBN.

E. Constructing Computational Models

“An important practice is the ability to create new or extend existing computational models. This requires being able to encode the model features in a way that a computer can interpret....Students who have mastered this practice will be able to implement new model behaviors, either through extending an existing model or by creating a new model either within a given modeling framework or from scratch [2].”

JO: How can the Bayesian Belief Network developed for attesting to the authenticity of records preserved in an archival system be used with observations of a particular archival system to determine the likelihood that they are authentic?

ChatGPT: That’s an excellent and crucial follow-on question. Once you have a Bayesian Belief Network (BBN) that models the InterPARES Baseline Requirements for Preserving Authentic Electronic Records, the next step is applying it to real-world archival observations so that the model produces evidence-based probabilities of authenticity.

Below is a structured explanation of how to use the BBN operationally with data from an actual archival system.

1 □ Concept: Linking the Abstract Model to Observations

The BBN encodes the causal logic of authenticity—how evidence (fixity, custody, documentation, description) increases belief that a record remains authentic.

Your archival system provides observable data that serve as evidence inputs to those lower-level nodes.

BBN Node	Observable in System	Example Data Source
FixityBeforeAfterRepro	Checksum or hash comparison logs	Automated fixity reports

BBN Node	Observable in System	Example Data Source
ChanOfCustodyLogsQuality	Provenance metadata, transfer logs	Repository audit logs
AuditTrailEffectiveness	Access control & change logs	System audit module
ReproEventMetadata Completeness	Preservation action metadata	PREMIS event records
DerivationLinkage	File relationships, identifier mappings	METS structure maps
ChangeLogCoverage	Version control history	Repository CMS or Git-based log
DeviationDisclosurePractice	Policy documents or user notes	Descriptive metadata notes
DescriptionIncludesContext	Metadata fields: provenance, function	Descriptive metadata schema
DescriptionCapturesChanges	Version/derivative notes	METS, PREMIS, or custom fields

These observations instantiate the evidence nodes.

2 3-Step Workflow for Using the BBN with an Archival System

Step 1: Operationalize the Evidence Nodes

Create observation rules that translate system data into BBN states.

For example:

Observation Rule	BBN State
≥99% of objects have valid fixity checks	high
70–98% valid	medium
10–69% valid	low
<10% or unknown	absent

Repeat for each evidence node. These rules become your mapping layer between archival data and the BBN input states.

Step 2: Enter Observations and Update Beliefs

Once the archival metrics are collected, you input them into the network as evidence (observed node states).

For example, suppose your observations are:

- Fixity reports: high
- Custody logs: medium
- Reproduction documentation: low
- Archival description completeness: high

When you set those observations in the BBN (GeNIe/Hugin), the system performs Bayesian updating, producing posterior probabilities such as:

- $P(B1=\text{true} \mid \text{evidence}) = 0.81$
- $P(B2=\text{true} \mid \text{evidence}) = 0.55$
- $P(B3=\text{true} \mid \text{evidence}) = 0.88$
- $P(\text{PresumeAuthentic}=\text{true} \mid \text{evidence}) = 0.73$

This output means: given what we observed, the system is 73% likely to meet the Baseline Requirements for authenticity.

Step 3: Analyze Sensitivity & Value of Information

After updating, perform:

Sensitivity analysis – reveals which nodes most affect the authenticity belief. → Example: if “Chain-of-Custody Logs Quality” has the highest influence, focus audits there.

Value-of-Information (VoI) – estimates how much the uncertainty would drop if you could verify a currently unknown node. → Example: verifying DerivationLinkage might increase confidence from 0.73 → 0.86.

These analyses guide resource allocation for auditing, documentation, or process improvement.

3 Integration with Archival Workflows

Automated Ingestion

- Integrate the BBN with your digital preservation system (e.g., Archivematica, Preservica).
- Create scripts that periodically extract metrics and map them to evidence node states.

Dashboard Reporting

- Build a dashboard that displays the current (PresumeAuthentic) and contributing factors.
- Use traffic-light indicators (Green > 0.8, Yellow 0.5–0.8, Red < 0.5).

Audit & Certification

- Use the posterior probabilities as supporting evidence in TRAC, ISO 16363, or internal authenticity audits.
- Include BBN outputs in annual reports to show data-driven assurance trends.

6 Benefits

- ✓ Quantifies authenticity confidence instead of relying solely on checklists
- ✓ Supports evidence-based management decisions
- ✓ Enables predictive modeling (“what-if” scenarios)
- ✓ Provides transparent reasoning traceable to InterPARES Baseline Requirements

Evaluation: GPT-4o explains how to use the BBN operationally with data from an actual archival system. In BBNs, observations entered into evidence nodes propagate through the network via conditional probability updates. GPT-4o also recommends the use of Hugin (Handling Uncertainty In General Inference Network) to perform Bayesian inference. ChatGPTs' description of evidence insertion, posterior updates and sensitivity analysis is consistent with the descriptions in the *Hugin Expert Technical Manual* [9].

III. CONCLUSIONS

What exactly do we mean when we state that ChatGPT knows something? We mean the same as stating that a student knows the answer to a question posed to them after they have been provided the information necessary to answer the question through reading or lecture.

The following table summarizes the knowledge of digital archival practices that GPT-4o has exhibited as reported in this paper. It also indicates the kinds of computational thinking know-how exhibited.

Knowledge of Digital Archival Practice	Modeling and Simulation Thinking Know-How
Emulation of obsolete PC hardware and MS-DOS to support operation of computer games	Using Computational Models to Understand a Concept
Determining economical and reliable archival preservation media	Using Computational Models to Find and Test Solutions
Assessing the validity of the InterPARES Preservation Model	Assessing Computational Models
Reasoning with beliefs about requirements for the preservation of authentic digital records	Designing Computational Models
Constructing a system to support reasoning about trustworthy archival systems	Constructing Computational Models

The research results reported in this paper completes our exploratory research reported in two previous papers to determine whether ChatGPT had knowledge of digital archival practices of which professional archivists should also have knowledge. The research results also identify those practices that require computational thinking abilities. In the three papers, GPT-4o is shown to have knowledge of 18 digital archival practices. These practices taken together involve all 22 of the computational thinking types identified by Weintrop. In addition, GPT-4o demonstrated the capability to perform 10 of the digital archival practices. The next questions on our research agenda are those such as "How can GPT-4o, or technology with similar abilities, act as a mentor to graduate students needing to develop computational thinking abilities involved in digital archival practices?"

REFERENCES

- [1] OpenAI. "GPT-4 Technical Report." 2023 <https://arxiv.org/abs/2303.08774>
- [2] D. Weintrop, E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille, & U. Wilensky. "Defining Computational Thinking for Mathematics and Science Classrooms," *Journal of Science Education and Technology*, 25(1), pp. 127–147, 2016. <https://ccl.northwestern.edu/2015/Weintrop%20et%20al.%20-%202015%20-%20Defining%20Computational%20Thinking%20for%20Mathematics%20an.pdf>
- [3] W. Underwood & J. Gage (2023) "Can GPT-4 Think Computationally about Digital Archival practices?" Computational Archival Science Workshop #8. IEEE Big Data Conference, Sorrento, Italy, Dec. 15-18, 2023. https://ai-collaboratory.net/wp-content/uploads/2023/11/S01213_7999.pdf
- [4] W. Underwood & J. Gage (2023) "Can GPT-4 Think Computationally about Digital Archival practices? – Part 2" Computational Archival Science Workshop #9. IEEE Big Data Conference, Washington DC, Dec. 2024 https://ai-collaboratory.net/wp-content/uploads/2024/11/S01206_4692.pdf
- [5] DOSBox Team. "DOSBox Reference Guide." DOSBoxWiki www.dosbox.com/wiki/
- [6] UC Santa Cruz Storage Systems Research Center. "Storage Reliability Modeling." UCSC SSRC Projects, <https://www.ssrc.ucsc.edu/projects/reliability.html> .
- [7] W. Underwood et al, (2001) "Appendix 7: Walkthrough Applying to the Preserve Electronic Records Model (Version 5.1).," "The Long-term Preservation of Authentic Electronic Records: Findings of the InterPARES Project. (Luciana Duranti Ed.), https://www.interpares.org/display_file/interpares_book_p_app07.pdf
- [8] N. Fenton and M. Neil, "Risk Assessment and Decision Analysis with Bayesian Networks," CRC Press, 2018.
- [9] Hugin Expert, "Hugin Decision Engine Technical Documentation," 2020. https://download.hugin.com/webdocs/manuals/9.7/htmlhelp/pages/Releases/release8_9.html