THE EUROPEAN SOCIETY OF MEDICINE
Medical Research Archives, Volume 14 Issue 3

RESEARCH ARTICLE

Transforming Legacy Clinical Trial Schedules of Activities into Interoperable Digital Formats

Mark A. Kramer

OPEN ACCESS

PUBLISHED
31 March 2026

CITATION
Kramer, M., 2026. Transforming Legacy Clinical Trial Schedules of Activities into Interoperable Digital Formats. Medical Research Archives, [online] 14(3).

ISSN
2375-1924

ABSTRACT

Objective: Clinical trial protocols contain Schedule(s) of Activities that specify patient visit timing and procedures. Extracting SoAs from legacy protocols into interoperable standards such as Fast Healthcare Interoperability Resources (FHIR) and the Unified Study Definitions Model (USDM) is limited by widely varying table formats and complex trial activities. ProtocolMiner automatically extracts Schedule(s) of Activities from protocols, interprets their semantics, and outputs standards-compliant representations.

Materials and Methods: Semantic analysis was applied to determine visit timing, procedures, cycles, and exceptions in 29 diverse protocols. The methodology maps to FHIR PlanDefinition and ActivityDefinition resources, as well as USDM classes including ScheduleTimeline, Encounter, and Activity.

Results: ProtocolMiner created accurate, detailed timelines for 22 of 29 SoAs, with minor errors in seven. FHIR successfully represented complex timing relationships including relative timing with windows, open-ended repetitions with stopping criteria, and conditional execution. USDM mapping enables integration with Clinical Data Interchange Standards Consortium-compliant study design workflows.

Conclusion: Automated Schedule of Activities extraction from legacy clinical trial protocols and transformation into FHIR and USDM is feasible with high fidelity. ProtocolMiner bridges the gap between legacy protocols and emerging digital standards, enabling computational protocol management and cross-system interoperability in clinical research.

Keywords: Clinical trials, Schedule of Activities, FHIR, USDM, healthcare interoperability, protocol digitalization, ICH M11, CDISC

1. Introduction

Evidence-based medicine depends on clinical trials, and within each trial protocol, the Schedule of Activities (SoA) plays a pivotal role by enumerating study procedures, assessments, and planned data-capture time points. The SoA is the primary bridge between a human-readable protocol and executable workflows used by sites, contract research organizations, and sponsors. Yet SoAs vary widely in structure, semantics, and completeness, contributing to operational inefficiencies and the risk of procedural and timing deviations.[1] Even when SoA content is conceptually similar across studies, differences in layout, timing expressions, and footnote conventions hinder automation of study startup, scheduling, and downstream data flows.[2]

This paper presents *ProtocolMiner*, an AI-assisted methodology for extracting SoAs from legacy clinical trial protocols, performing detailed semantic analysis of visits, procedures, and timing, and transforming the resulting model into interoperable, standard representations that can be consumed by downstream systems. In contrast to prior work that either assumes pre-structured inputs or uses AI to extract high-level protocol elements, ProtocolMiner targets the end-to-end pipeline from complex multi-page PDF SoA tables to validated, standards-compliant representations with outputs in two leading standards, Clinical Data Interchange Standards Consortium's (CDISC) Unified Study Definitions Model (USDM), and Health Level Seven International's (HL7) Fast Healthcare Interoperability Resources (FHIR).

2. Prior Work

To address the challenge of computable protocols, multiple standards efforts are advancing structured representations of SoAs. The USDM standard defines a shared, machine-readable study design model that captures key protocol elements, including study timelines, encounters, activities, and timing relationships.[3] It is designed to integrate with the broader CDISC ecosystem, including controlled terminology and biomedical concepts. This integration enables automated generation of downstream artifacts such as case report forms, Study Data Tabulation Model datasets, and analysis specifications. The model also supports regulatory submission requirements, making it particularly valuable for sponsors seeking to streamline their development processes.

TransCelerate's Digital Data Flow (DDF) initiative builds on USDM to support end-to-end digital protocol flows, including exchanging structured study definitions between sponsors, CROs, sites, and technology platforms.[4] The International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use (ICH) M11 standard complements these efforts by defining a semi-structured protocol document suitable for regulatory use, which can be mapped into USDM but currently does not prescribe a fully digital representation of SoAs.[5] Utilities for converting M11 to USDM have been provided by Data4Knowledge.[6]

Several lines of prior work have focused specifically on formalizing SoAs using FHIR resources. The HL7 Vulcan Accelerator[7] has published an implementation guide that defines how FHIR can be used to represent SoAs in a computable form.[8] Richardson demonstrated that FHIR could describe SoA requirements across diverse study designs including simple linear schedules, cyclical patterns typical in oncology trials, and event-driven sequences common in vaccine studies.[9] Building on this foundation, Richardson and colleagues proposed a definitional-resource–based approach in which SoA semantics are supported by graph-based structures and targeted extensions to handle intricate temporal dependencies and reuse across activities and visits.[10] Genyn and co-authors further explored FHIR-based "activity libraries" that encapsulate reusable activity definitions and scheduling patterns to support clinical trial design and execution, emphasizing modularity and interoperability between authoring tools and operational systems.[11] These studies show that SoAs can be represented in FHIR, but generally assume that the underlying schedule content is already available in a structured or semi-structured form rather than being extracted from legacy documents.

There has also been progress on creating structured protocols directly, using specialized authoring tools. Shin and colleagues described an automated protocol template system in which key design parameters are specified in structured form and SoA tables are generated programmatically, reducing manual effort and improving internal consistency.[12] Several open-source and vendor tools now support authoring clinical trial designs and exporting them as structured USDM, including OpenStudyBuilder[13] and Faro Health.[14] These approaches highlight the long-term direction of protocol authoring but do not address the large corpus of legacy protocols.

More recently, AI-assisted methods have begun to tackle protocol information extraction from unstructured documents. Kestemont and Rogiers[15] took a retrieval-augmented generation approach, embedding protocols into a vector database for information extraction, though the completeness of SoA extraction into USDM was unclear. Babaeipour and colleagues also reported a retrieval-augmented generation approach, showing that large language models can improve extraction accuracy and auditability relative to purely rule-based pipelines.[16] Both works underscore the promise and difficulty of using AI to recover structured protocol data, particularly when tables span multiple pages, use merged cells, or encode complex timing semantics in footnotes. These systems provide coarse schedule summaries and do not attempt a full, standards-aligned transformation into FHIR and USDM that preserves the detailed temporal and conditional logic required for interoperable execution.

3. Semantic Analysis of SoAs

3.1 STARTING POINT: THE CELL LIST

The starting point for ProtocolMiner's semantic analysis is the *cell list*, a cell-by-cell representation of the SoA table. The cell list must include the entire SoA, even if it spans multiple physical pages.

Every cell is described by the properties: Row_Start, Row_End, Col_Start, Col_End, Cell_Content, Referred_Content. Four coordinates are required to capture the extent of each cell, since merged cells are common and have semantic significance. The cell content is divided into the text in the cell itself and any content referenced by the cell's footnotes or other text (e.g., *see section x.y*). Formats such as font, text rotation, and alignment are also collected by ProtocolMiner for rendering purposes, but are not required for semantic analysis.

Extracting the cell list from Portable Document Format (PDF) is complex, but outside the scope of this paper. It involves analyzing each subtable using image analysis and/or PDF extraction, and then combining the subtables. Pitfalls in this process have been described elsewhere.[17] In a previous study, generic PDF table extraction methods were put to the test, and performed poorly, either omitting rows or columns or incorrectly aligning or merging cells.[18] Several flagship AI systems (Claude Sonnet 4, Gemini 2.5 Pro, and GPT-4.1) also fared poorly. In a related unpublished test, ProtocolMiner successfully processed protocols containing 1 to 21 subtables, achieving high accuracy with only 2 minor errors across 30 protocols tested. To the best of the author's knowledge, ProtocolMiner is the only PDF extraction tool with the ability to combine multiple subtables.

Since the time of that analysis (August 2025), AI capabilities have advanced considerably. The author recently put the same cell list creation problem to Claude Opus 4.5 and OpenAI GPT 5.2 with extended thinking enabled. Instead of directly reading and interpreting the PDF, both proceeded to write python code. Claude's algorithm employed *pdfplumber*,[19] one of the underlying libraries used by ProtocolMiner. However, it still missed and misaligned several cells. A different approach was used by GPT 5.2, which converted the PDF to an image, then analyzed it using libraries such as *OpenCV*.[20] It arrived at the correct structure, but missed the majority of footnote superscripts. Interestingly, ProtocolMiner uses a combination of these two techniques, with special techniques to handle complications such as watermarks, grid misalignments, and footnote

identification, as well as several methods for combining tables across page breaks.

3.2 INTERNAL SCHEMA

ProtocolMiner mirrors how a human reads an SoA table: identify timing columns, then rows, then cell intersections. It uses an internal schema closely aligned to the semantic extraction task, rather than directly mapping to FHIR and USDM, which would impose more complexity. The conceptual model can be summarized as follows:

- An SoA is composed of one or more timelines,
- Each timeline contains one or more visits ("visit" is used to describe timeline events, whether or not the event involves direct contact between a patient and practitioner),
- Each visit is defined by a visit plan,
- Each visit plan contains one or more procedures (the term "procedure" is used for medication administrations, observations, measurements, imaging, and other similar actions), and
- Each procedure is defined by a procedure plan.

The schema used to represent this conceptual model is shown in Figure 1. The Plan class can represent any of the following:

- The plan for a sequence of visits (a timeline), either the main timeline or an auxiliary timeline triggered by an unscheduled event,
- The plan of activities for a single visit, or
- A single procedure, observation, or measurement (a Plan with no sub-activities).

The Activity class represents an instance of a visit or procedure that takes place during a plan. It has specific timing information with respect to the Plan that contains it. For a timeline, the contained Activities are visits. For visits, the contained Activities represent procedures carried out during that visit. These relationships are shown on the right side of Figure 1.

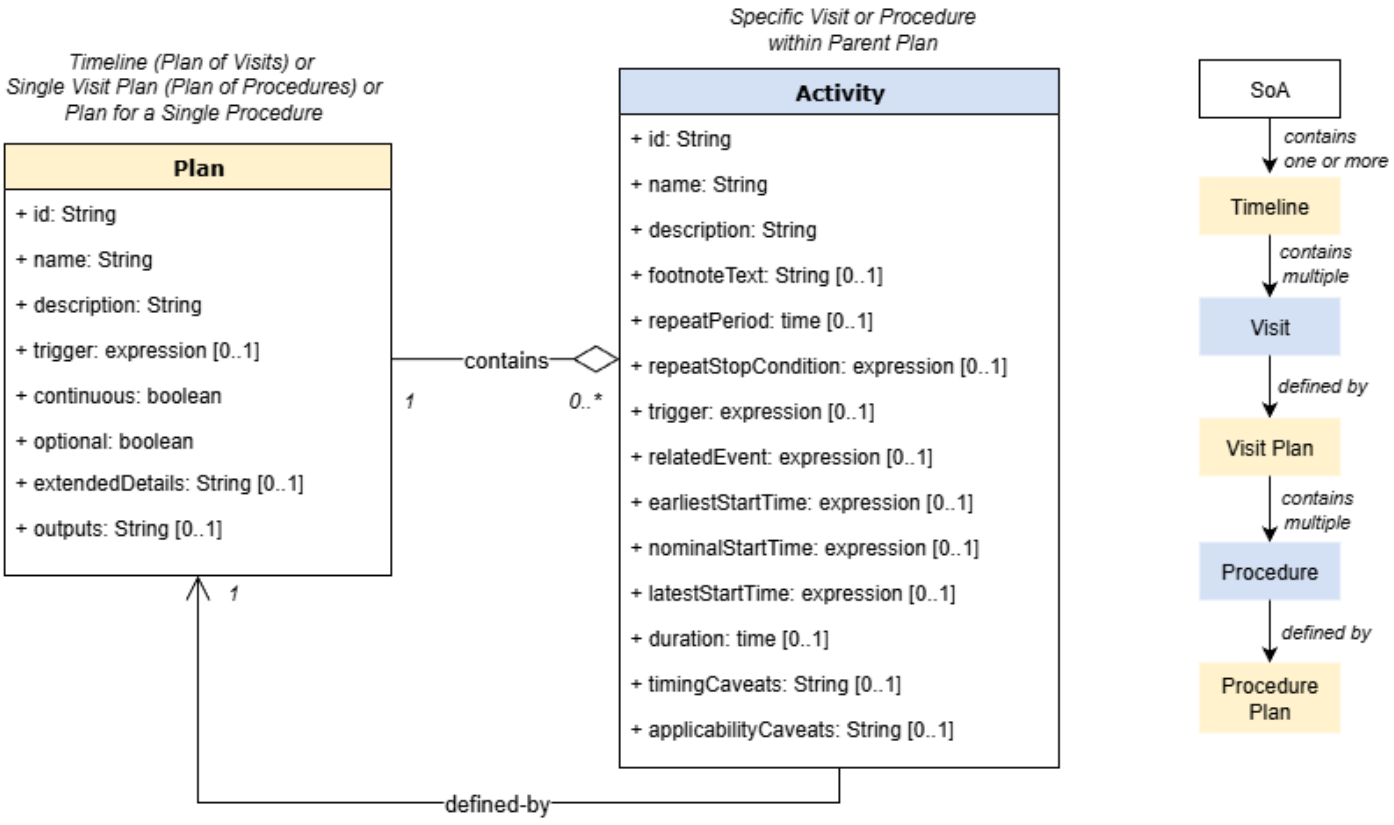

Figure 1. SoA model used in ProtocolMiner

3.3 TIMING REPRESENTATION

The start of an activity can be described in two ways: by a triggering event, or by a temporal relationship to the beginning or end of another activity. For relative timing, the key information is captured in the Related Event and the Nominal, Earliest, and Latest Start Times. The Related Event is the point in time that the start time of the current activity is connected to. The Related Event element is one of the following expressions (vertical bar = 'or'):

- A phrase in the form "[start | end] of [Activity ID]" where ID is the ID of a different Activity,
- The name of a point-in-time event such as "baseline" or "early termination of treatment", or
- "None" if there is no related event tied to the start of the activity.

The Related Event is always a point in time, not a time interval.

The Nominal, Earliest, and Latest Start Times specify when the activity should take place, relative to the Related Event. The Nominal Start Time is the normal or target time for the start of the activity, and the Earliest and Latest Start Times capture the permissible time interval around the activity, if specified. If only one of these times is specified, it must be the Nominal time. If two times are specified, it must be the Earliest and Latest Start Time. All three Start Times share the following syntax:

- A phrase in the form " [number] [minutes | hours | days | weeks | years] [before | after]", for example: "10 days before" or "2 weeks after",
- The single word [before | at | after], or
- "Unspecified," if the relative time is not given.

Because the Related Event and the start of the Activity are both points in time, not time intervals, the words "before", "at", and "after" cover all possible temporal relationships.

The next timing element is Duration. Duration is specified as a unit-bearing time quantity and an optional inequality, either <= or >=, forming expressions such as "2 hours", "6 months", "<= 3 days", ">=10 minutes".

To illustrate how timing attributes work together, Table 1 provides examples of how to combine these elements.

Table 1. Examples of how the start time and related event determine the start of an activity.

Start Time	Start Time Value	Related Event	Then the Activity starts:
Earliest	5 days after	End of Treatment (EOT)	At the earliest, 5 days after EOT
Nominal	7 days after	start of C1D1	Nominally, 7 days after the start of C1D1
Latest	30 minutes before	start of Activity 2	At the latest, 30 minutes before the start of Activity 2
Nominal	28 days before	baseline	Nominally, 28 days before baseline
Earliest	at	start of Infusion	At the earliest, at the start of Infusion
Latest	before	end of Cycle 1	At the latest, before the end of Cycle 1

3.4 TABLE CELL CLASSIFICATION

The first step of semantic analysis uses AI to locate certain key rows and columns in the combined table, specifically:

- Timing row, the row in the column headers containing the most detailed time information,
- Time window row, containing tolerances (±) on the timings (not always present),
- Activity column, containing names of study activities,
- First data row and first data column (non-header), and
- Notes or additional information column(s), if applicable.

Information from the cell lists, arranged in rows or columns, is provided to AI. Prompts include appropriate guidelines, for example, this guidance for detecting the timing window row:

- Cannot be the same as the Timing Row
- Located near the top of the table (within the first 4-5 rows), above the first data row
- Dedicated specifically to defining time windows (time ranges) for study activities
- May be labeled "Window", "Allowed Variation", "Timing Window", or similar
- Contains numerical values with time units, such as hours, days, or months
- Contains plus/minus signs or the ± symbol
- Example: "Visit Window (Days) | -7 to 0 | 0 | ±3 | ±3 | ±3 | ±3 | ±3 | | 0 to +5"

In addition to the key rows/columns, every cell in the table is classified as either a column heading, row label, separator, note, notes heading, or data cell.

3.5 PROCEDURE AND VISIT IDENTIFICATION

Based on table cells identified as activities, the algorithm builds a table of all atomic activities that are part of the SoA (those without sub-activities). Once this list has been created, an AI system is employed to create detailed descriptions of each procedure. With the entire SoA document as input, the AI system is asked to summarize all information in the document regarding the timing of the procedure (e.g., whether before or after any other procedures and by how much), what takes place (e.g., carrying out procedures, making observations, collecting specimens, answering questionnaires, administering medications, etc.), where it occurs (home visit, at clinic, specialist office, via telemedicine, etc.), whether the procedure is continuous (for example, collection of adverse events), how the activity is carried out (the method or any significant execution details), how much (quantity of specimens, dose of medications, etc.), and how long it lasts (for example, a 15-minute exercise tolerance test). This process lessens the reliance on the SoA table and its footnotes, and creates a more comprehensive picture of the procedure.

A similar process is then carried out for visits, nominally found in the columns of the SoA table. At this point, the emphasis is on identifying distinct visit plans, rather than distinct visits. Because of the way the SoA is modeled, plans are reusable. For example, if the same visit plan is used in five treatment cycles, then at this stage one visit plan is identified, not five. Again, the AI system is asked to search the entire protocol document for further details, including information on the number and duration of cycles, visit timing, and the allowed time windows, if any. Finally, for each visit plan, ProtocolMiner builds the association between the visit and the procedures carried out at that visit, setting the stage for creating Activity instances.

3.6 VISIT TIMING DETERMINATION

Determination of visit timing proceeds in three distinct phases: visit occurrences, visit chronology, and relative timing.

3.6.1 Enumerating Visit Occurrences

A single column of the SoA table can represent multiple visits. This approach is particularly common in oncology trials with repeating treatment cycles, chronic disease studies with regular monitoring intervals, or any protocol where the same assessments

are systematically repeated at predictable time points. By consolidating these repetitive visits into a single column, the SoA becomes more concise and easier to read while ensuring that study teams understand that the marked procedures must be completed at each of the specified time points within the indicated range or cycle pattern.

To be explicit about the number and sequence of visits, a timeline must expand hierarchical and multiplicative combinations implied by the column headers. The AI system reads the headers to determine if a column implies multiple visits. Left to right, visits are not always arranged chronologically, as in this example:

Cycles 2-5	
Days 1, 21	Day 11

Here, days 1 and 21 share the same visit plan, repeated twice in each of four cycles. Overall, there are twelve distinct visits: C2D1 (Cycle 2 Day 1), C2D11, C2D21, C3D1, C3D11, C3D21, C4D1, C4D11, C4D21, C5D1, C5D11, C5D21.

The AI system also identifies cases where the number of repetitions cannot be predetermined. A column header might employ expressions like "… and beyond", "Every 14 days", "Monthly until…", or simply "+" (as in "Cycle 2+"). Two examples of open-ended repetition are included in the SoA of Figure 2. This flexibility is essential in trials where treatment duration depends on individual patient response, tolerability, or other clinical factors that cannot be predetermined during protocol development, allowing the SoA to accommodate variable study durations while maintaining standardized procedural requirements across all repeated visits. In these cases, the AI system identifies the visit repeat period and the stopping criteria for the repetition. Examples of stopping criteria are: study completion, treatment discontinuation, and disease progression.

Closed-ended and open-ended repetitions can also be combined in a single column, for example, "Every 8 weeks for the first 12 months; every 12 weeks thereafter." This situation is handled by expanding the closed-ended repetition into specific visit instances, followed by the open-ended cycle repeated until a stopping condition.

Evaluation	Screening (up to 28 days before Day 1)		Treatment Phase[a]				End of Treatment (EOT)	Post Treatment Follow-up Phase			Notes
			Cycle 1			Cycle 2 and Beyond		Safety follow-up Period		Survival follow-up	a A cycle is 21 days
	D-28 to D-15	D-14 to D-1	D1 (±1)	D8 (± 1)	D15 (±1)	D1 (± 2)	30 (±7) days after last IMPs admin	At 60 (±7) days after last IMPs admin	At 90 (±7) days after last IMPs admin	Every 90 days (±7) after last safety follow-up	
Blood Typing Interference Test		X				Cycle 2 Day 1 only					Section 10.3 Before each transfusion.
Serology HBV and HCV (for HCC only)	X										Section 10.3
Urinalysis (at baseline and if required) /urine dipstick		X	X	As clinically indicated			X	X	X		Section 10.3
Disease Assessment											
CT/MRI (for HCC, SCCHN, and EOC)	X					X (Weeks 9, 18, 27, and then every 12 weeks)	X (if necessary)		X (until PD is confirmed if no PD documented & confirmed)		Section 8.1
		X (within									Section 8.1

Figure 2. Examples of open-ended timing (Cycle 2 and Beyond, Survival follow-up every 90 days). (Sanofi S.A., Protocol ACT15377 Amendment 5, Nov. 2020, NCT03637764).

3.6.2 Creating Timelines

After visits have been enumerated, the next phase involves using AI to create timelines with visits in chronological order. This will always include the main timeline, the standard, planned progression for participants who complete the study as designed. For the main timeline, the chronological sequence is created by sorting the visits hierarchically by study phase (e.g., screening, treatment, follow-up), then by cycle number (if applicable), and then by day within cycle or study phase.

In addition to the main timeline, other visit sequences can be triggered by unscheduled events such as early termination of treatment, serious adverse events, disease progression, or strategic treatment interruptions (also known as medication holidays or treatment breaks), or end of treatment (EOT). For each alternative timeline, the AI system is asked to determine the triggering event that leads to the alternative pathway, the sequence of visits involved in the alternative timeline, and where the alternative pathway rejoins the main pathway (if it does). For example:

- After a strategic treatment interruption, treatment might resume at the start of the next full cycle, or
- If there is an early termination of treatment, the main path could be rejoined at the beginning of the follow-up period.

ProtocolMiner does not create timelines focused on each type of activity, for example, a timeline just for imaging studies. This type of timeline can be useful for resource scheduling and costing.

3.6.3 Visit Timing

The final phase of visit sequencing is the addition of relative timing information. To do this, the AI system is asked to determine which event the current activity is related to and/or the triggering event for the activity. In the first case, the AI system is prompted for the relative timing, returning the result using the syntax described in the section 3.3 TIMING . If there is information indicating the visit is optional or as-needed, the AI system will report that, as well. The input includes the full context for every activity in the timeline, including footnotes, notes, timing windows, and extended descriptions.

Example 1: The follow-up (f/u) period consists of quarterly checkups in the first year, and then yearly checkups for the next 4 years.

ID	Description	Related Event	Earliest Start Time	Nominal Start Time	Latest Start Time
V-1	F/u Month 3	EOT	--	3 months after	--
V-2	F/u Month 6	EOT	--	6 months after	--
V-3	F/u Month 9	EOT	--	9 months after	--
V-4	F/u Month 12	EOT	--	12 months after	--
V-5	F/u Month 24	EOT	--	24 months after	--
V-6	F/u Month 36	EOT	--	36 months after	--
V-7	F/u Month 48	EOT	--	48 months after	--
V-8	F/u Month 60	EOT	--	60 months after	--

Example 2: There are three 28-day cycles. The start of the second cycle has a window of ±4 days and the start of the third cycle has a window of ±3 days. Each cycle has two visits, day 1 and day 15±1 day.

ID	Description	Related Event	Earliest Start Time	Nominal Start Time	Latest Start Time
V-1	Cycle 1 Day 1	BASELINE	--	0 days after	--
V-2	Cycle 1 Day 15	start of V-1	13 days after	14 days after	15 days after
V-3	Cycle 2 Day 1	start of V-1	24 days after	28 days after	32 days after
V-4	Cycle 2 Day 15	start of V-3	13 days after	14 days after	15 days after
V-5	Cycle 3 Day 1	start of V-3	25 days after	28 days after	31 days after
V-6	Cycle 3 Day 15	start of V-5	13 days after	14 days after	15 days after

3.6.4 Procedure Timing

Determining the timing of procedures within visits starts with the previously-determined mapping of procedures to visits. Nominally, this mapping corresponds to the 'X' marks. However, exceptions to the occurrence of an event may be documented in a footnote or as a note appearing in a data cell. For example, in Figure 2, the blood typing interference test is indicated in the column "Cycle 2 and Beyond", the text in the cell indicates "Cycle 2 Day 1 only", which negates the activity in Cycle 3 and beyond. This is captured as a timing caveat attached to that activity in the plan that represents the Cycle 2 and Beyond Day 1 visits. Similarly, conditions may be attached to an activity, such as urinalysis is noted as "As clinically indicated" in Figure 2. This will appear as an applicability condition, i.e., to be performed only if clinically indicated.

Activities that span multiple times (horizontally merged cells in the data area) are tagged as continuous activities. Activities can also be tagged as optional, if indicated in the SoA table, footnotes, or other sources.

Example 3: A clinic visit has five activities: filling out a questionnaire, a physical exam, a 20-minute Exercise Test, an electrocardiogram (ECG), and an optional counseling session. The physical exam must be before the exercise test, which is followed by an ECG, 5 minutes to 15 minutes after the exercise test. The questionnaire can be filled out any time during the visit. This is expressed by the following timing (selected columns shown):

ID	Description	Related Event	Nominal Start Time	Earliest Start Time	Latest Start Time	Duration	Optional
A-1	Physical Exam	--	--	--	--	--	False
A-2	Exercise Test	end of A-1	after	--	--	20 min	False
A-3	ECG after Exercise	end of A-2	--	5 min after	15 min after	--	False
A-4	Questionnaire	--	--	--	--	--	False
A-5	Counseling	--	--	--	--	--	True

4. Mapping to Standards

4.1 CONVERSION TO FHIR

Two primary FHIR resources are used to describe SoAs.[21] PlanDefinition represents the overall structure of a study schedule, including its timelines, visit sequences, and the relationships between activities. Each action within a PlanDefinition can specify timing relationships, conditions for execution, and references to detailed activity definitions. ActivityDefinition represents individual procedures, observations, or measurements that occur during study visits. These resources work together to create a complete, computable representation of what should happen during a clinical trial and when. Mappings to FHIR are given in Table 2 and Table 3. In addition, the Group resource is used to represent study cohorts.

To facilitate the translation, ProtocolMiner uses FHIR Shorthand[22] as an intermediate representation. FHIR Shorthand, an HL7 standard, is a widely-used method of specifying FHIR profiles, instances, and implementation guides. It was used to create the instances of PlanDefinition and ActivityDefinition that represent the SoA, as well as a Group resource representing the study population. The conversion from FSH to FHIR is accomplished automatically using SUSHI, the open-source FSH compiler.[23] For convenient data sharing, ProtocolMiner also generates a single FHIR Bundle containing all these resources.

Table 2. Key mappings of timelines to FHIR's PlanDefinition

Timeline Attribute	FHIR PlanDefinition Element
ID	id, url
Name	name
Description	description
Trigger	trigger
Subactivity:	action
ID	action.id
Name	action.title
Description	action.description
Trigger	action.trigger
Optional	action.requiredBehavior
Continuous	action.type
Defining ID	action.definitionCanonical
Duration	action.timingDuration
Repeat Period	action.timingTiming.repeat.period
Stopping Criteria	action.condition (kind = stop)
Timing Caveat	action.condition (kind = start)
Applicability Caveat	action.condition (kind = applicability)
Relative Timing:	action.relatedAction
Related Event	action.relatedAction.actionId, action.relatedAction.relationship
Nominal Start Time	action.relatedAction.offsetDuration
Earliest Start Time	action.relatedAction.extension.valueRange.low
Latest Start Time	action.relatedAction.extension.valueRange.high

Table 3. Mappings to FHIR's ActivityDefinition

Activity Attribute	FHIR ActivityDefinition Element
ID	id, url
Name	name, code.text
Description	description
Activity Output	productReference (This is a partial mapping because ObservationDefinition is not a permitted data type for productReference)
Optional	*Included in timeline*
Continuous	*Included in timeline*

4.1.1 Activity Timing in FHIR

In FHIR, activities can be triggered in two ways: when a logical condition becomes true (action.trigger) or due to a timing relationship to the beginning or end of another action, the related event (action.relatedEvent). Each action that is ready to execute is also subject to additional conditions (action.condition) that control whether the action should start or not. According to the FHIR documentation, the timing elements (trigger and relatedEvent) specify *when* the action should take place, and action.condition specifies *whether* the action should take place. There are three types of conditions, start, stop, and applicability. An example of an applicability condition is pregnancy testing applicable only to females of child-bearing potential. A stop condition halts an executing activity (including any repeats specified in action.timingTiming) as soon as it becomes true.

Triggers and timed execution are to some extent interchangeable. For example, an activity to take place nominally 7 days after another activity could have a time-based trigger based on the date being a week after the first activity. However, a trigger event does not allow for time windows, so it is preferable to express actions with timing relationships. Triggers are reserved for starting unscheduled event timelines.

PlanDefinition.action.relatedAction.relationship expresses the timing of the action relative to the relatedAction. The FHIR representation is not one-to-one with ProtocolMiner's internal representation of relative timing. The translation of these relationships is shown in Table 4. The values in the first column come from the phrases that describe the Earliest, Nominal, or Latest time. Those in the second column are part of the phrase "start of" or "end of" that describe the related event.

For activities taking place during a visit, the timing of an activity is often unspecified (last row of Table 4). This means the activity takes place *during* the visit. FHIR R4 and R5 cannot express "during", as distinct from "concurrent" (meaning at the same time). However, starting in FHIR Release 6 (R6), "during" can be expressed by setting relatedAction.relationship to "after-start" and relatedAction.endRelationship to "before-end".

Indefinitely repeating activities are represented using the timing element of an action by specifying the repeat period in action.timingTiming.repeat.period. The stopping criteria for the action are represented with a stop condition in action.condition. This approach enables representation of schedules such as "every 3 weeks until disease progression" or "every 28 days until unacceptable toxicity."

Table 4. Mapping of internal timing representations to FHIR.

ProtocolMiner Start Time modifier	ProtocolMiner RelatedEvent modifier	FHIR relatedAction.relationship	FHIR relatedAction.endRelationship (R6 only)
before	start of (or none*)	before	--
before	end of	before-end	--
at	start of	concurrent-with-start	--
at	end of	concurrent-with-end	--
after	start of	after-start	--
after	end of (or none*)	after	--
at	none*	concurrent	--
unspecified	unspecified	after-start	before-end

* Only for point-in-time events

4.2 MAPPING TO USDM

The USDM approach to schedule representation centers on several key classes. ScheduleTimeline represents a sequence of encounters or visits, analogous to FHIR's PlanDefinition. Each timeline contains ScheduledActivityInstance objects that define when specific activities occur. The Encounter class represents individual study visits, while Activity and Procedure classes define the specific actions performed. The Timing class in USDM provides a flexible mechanism for specifying when activities should occur relative to other events or absolute time points.

Conversion to USDM uses the FHIR resources as input, chosen because this approach is more likely to be of general use. The essential mappings are as follows: PlanDefinition maps to ScheduleTimeline, Encounter, and StudyEpoch; PlanDefinition.action maps to ScheduledActivityInstance; ActivityDefinition maps to Activity and Procedure; and timing relationships map to Timing.

The main PlanDefinition becomes the main ScheduleTimeline, and each action in the PlanDefinition becomes an Encounter and a ScheduledActivityInstance. Timing information from relatedActions is mapped to the Timing class in USDM. Each ActivityDefinition is converted to a USDM Activity, and a corresponding Procedure is created for each Activity. New universal unique identifiers are generated for all USDM elements, and references between objects are maintained using these identifiers.

5. Validation Methodology

To validate semantic extraction and timeline construction by ProtocolMiner, timelines were created for 29 protocols that have been used as test cases by CDISC, TransCelerate, and HL7. In the test, the only inputs to ProtocolMiner were the protocol PDF file and the page numbers of the SoA table(s), footnotes, and abbreviations.

Among the protocols examined, 83% had row-wise page breaks, 20% had column-wise page breaks, 40% had timing hierarchies deeper than 2 levels, 37% had explicit timing window rows, 20% had notes columns, 10% had closed-ended repetitions, 43% had open-ended repetitions, and 20% had multiple visits per column.

ProtocolMiner is designed to use multiple AI models. The test was conducted using claude-sonnet-4-20250514 and gemini-2.5-pro. The Gemini model was used when analyzing the entire protocol document, because of its large (1M token) context window. For other steps, Claude models, which outperformed OpenAI GPT models, were used.

After running ProtocolMiner, each SoA was analyzed to determine the accuracy of the SoA timelines. The timeline accuracy evaluation included the main timeline and unscheduled event timelines. Timeline activities must account for all start, stop, and applicability conditions, number of repeats, repeat periods, and all relative timing attributes (related event, earliest, nominal, and latest times).

6. Results

Results are summarized in Table 5. Overall, the SoAs extracted were extremely accurate with only minor deviations. Semantic analysis achieved high accuracy with 22 of 29 SoAs (77%) extracted with 100% accuracy. Among the errors identified: one protocol missed a ±3-day window on cycle visits because it was not mentioned in table entries or footnotes; one missed the earliest start time of the screening period; one produced ±2 weeks instead of ±10 days and failed to represent certain cycles as optional; one interpreted ±10-day windows incorrectly.

Table 5. Results on 29 protocols.

	Company, Identifier	Date, Rev.	National Clinical Trial #	Timeline Accuracy
1	Eli Lilly H2Q-MC-LZZT	2006	--	100%
2	Takeda C25003	March 2015, Amend. 7	01712490	Missed ±3-day window on cycle visits because it is not mentioned in table entries or footnotes.
3	Tesaro 213356	Jan 2019, v8.0	01847274	100%
4	Bayer BAY 88-8223	April 2018, V5.0	02043678	100%
5	Eli Lilly I3Y-MC-JPBL	April 2014, first version	02107703	100%
6	Alexion ALXN1007-GIGVHD-201	Feb. 2016, V8.0	02245412	100%
7	Amgen 20140254	Oct. 2016, Amend. 4	02575833	Missed earliest start time of the screening period.
8	AstraZeneca D0816C00020	Oct. 2018, V3.0	03402841	100%
9	Eli Lilly I8R-JE-IGBJ	Dec 2017, Amend. (a)	03421379	100%
10	Novo Nordisk NN9536-4373	March 2019, V1.0	03548935	100%
11	Novo Nordisk NN9536-4376	June 2020, V1.0	03548987	100%
12	Pfizer C3671002	Aug 2019, Amend. 3	03572062	100%
13	Takeda Ponatinib-3001	Oct. 2021, Amend. 10	03589326	100%
14	Sanofi ACT15377	June 2019, Amend. 04	03637764	Disease assessment visits were not properly represented because their timing was defined in the data area and not aligned to cycle visits in headers
15	Roche BP40657	Feb. 2023, V7	03735121	Open-ended repeat cycle with different time windows on different cycles were not captured.
16	Roche MO40597	Sept. 2019, V4	03817853	± 10-day time window came out as ± 2 weeks. Did not represent cycles 7/8 as optional.
17	Pfizer ARRAY-818-201	Jan. 2021, V4	03911869	100%

	Company, Identifier	Date, Rev.	National Clinical Trial #	Timeline Accuracy
18	MolMed IPR/33.C	Jan. 2020, VC	04097301	100%
19	Eli Lilly I8F-MC-GPHK(b)	Oct. 2021, Amend. b	04184622	100%
20	Roche WA42380	June 2020, V3	04320615	100%
21	NHLBI ORCHID	June 2020, V4.0	04332991	100%
22	Alexion ALXN1840-WD-204	March 2022, Amend. 3.1	04573309	100%
23	AstraZeneca D8851C00001	June 2021 V7.0	04723394	100%
24	Bristol-Myers Squibb CA045-020	Nov. 2021, Amend. 01	04730349	100%
25	Merck MK-7902-017-03	July 2022, Amend. 03	04776148	Inaccurate timing windows on post-treatment visits at QW8 and Q12W, both expressed in days, indicating difficulty working with mixed time units.
26	Vertex VX20-121-103	Aug. 2021, V3.0	05076149	100%
27	Sanofi LTS17352	June 2021, V1	05132127	100%
28	KalVista KVD900-301	May 2023, V4.1	05259917	Missed that treatment period visits are event-triggered by disease flare-ups
29	BeiGene BGB-dinutuximab beta-101	June 2023, Amend. 1.0	05373901	100%

6.1 FHIR OUTPUT VALIDATION

All generated FHIR resources passed validation against the FHIR Release 4 (R4) specification and the Vulcan Schedule of Activities Implementation Guide. The PlanDefinition resources correctly represented the hierarchical structure of timelines, visits, and procedures. Timing relationships were accurately captured using relatedAction elements with appropriate relationship codes, offset durations, and window extensions.

For a follow-up period with quarterly checkups in the first year and yearly checkups for the next 4 years, ProtocolMiner correctly produced a PlanDefinition with actions at 3, 6, 9, 12, 24, 36, 48, and 60 months after end of treatment, each with specified nominal start times and relatedAction references to the end-of-treatment activity.

For a protocol with three 28-day cycles where the second cycle has a ±4 day window and the third has a ±3 day window, with visits on day 1 and day 15±1 of each cycle, ProtocolMiner correctly produced PlanDefinition.action elements with appropriate relatedAction.offsetDuration values and extension elements capturing earliest and latest start times.

For clinic visits with ordered procedures, the algorithm correctly produced ActivityDefinition instances for each procedure with appropriate timing relationships, durations, and optional flags encoded in requiredBehavior.

6.2 USDM OUTPUT VALIDATION

Generated USDM output was validated against the USDM schema. The ScheduleTimeline, Encounter, ScheduledActivityInstance, Activity, and Procedure objects correctly represented the protocol structure. Timing information was appropriately mapped to USDM Timing class instances. Cross-references between USDM elements were maintained using generated universal unique identifiers, enabling navigation of the complete study design model.

7. Discussion

7.1 IMPLICATIONS FOR HEALTHCARE INTEROPERABILITY

ProtocolMiner addresses a critical gap in clinical trial interoperability. While FHIR and USDM provide destination formats for digital protocol representation, thousands of existing protocols remain in legacy PDF format. Accurate extraction and subsequent transformation of SoA content makes retrospective digitization feasible, which in turn allows the material to be integrated into contemporary clinical trial management systems and to be repurposed for secondary aims, including cross-trial analyses and meta-research.

The dual-output approach, generating both FHIR and USDM, maximizes the utility of the extraction process. Organizations using FHIR-based clinical systems can directly integrate protocol requirements into site workflows, scheduling systems, and electronic health records. Organizations aligned with CDISC standards can leverage USDM output for study setup, case report form generation, and regulatory submission preparation.

While FHIR and USDM serve overlapping purposes, they address different use cases and stakeholder needs. FHIR excels at healthcare system integration, enabling direct connection between protocol definitions and clinical care systems including electronic health records, laboratory information systems, and scheduling applications. This makes FHIR particularly valuable for sites conducting clinical trials, where protocol requirements must integrate with existing clinical workflows. USDM, by contrast, is optimized for study design and regulatory workflows. Its integration with CDISC standards makes it ideal for sponsors and contract research organizations managing the full lifecycle of study development. The detailed class model supports automation of study setup tasks and ensures consistency across the various systems used in trial execution.

7.2 FHIR REPRESENTATION CONSIDERATIONS

The mapping to FHIR revealed both the power and limitations of current FHIR versions. The PlanDefinition and ActivityDefinition resources provide a comprehensive framework for representing study schedules, with particularly strong support for timing relationships through the relatedAction mechanism. The ability to specify conditions for start, stop, and applicability enables representation of the complex conditional logic found in real-world protocols.

However, certain SoA patterns require workarounds in FHIR Release 4 and Release 5. The inability to express "during" relationships—as opposed to "concurrent" or sequential relationships—means that activities occurring within a visit but without specified order cannot be precisely represented. Release 6 addresses this limitation with the addition of endRelationship, enabling expression of activities that start after another activity starts and end before that activity ends.

Time windows present another challenge. While FHIR provides offsetDuration for nominal timing, earliest and latest times require extensions. ProtocolMiner uses extension elements with valueRange to capture these windows, following the approach recommended by the Vulcan implementation guide.

7.3 USDM INTEGRATION CONSIDERATIONS

The FHIR-to-USDM conversion approach enables ProtocolMiner to leverage FHIR as a well-validated intermediate representation before generating USDM output. Any timing or structural issues are identified and resolved during FHIR validation before propagating to USDM.

Some advanced USDM features are not yet implemented in ProtocolMiner. Biomedical concepts, which link activities to standardized terminology and enable downstream automation, require vocabulary mapping that is beyond the current scope. Similarly, detailed integration with CDISC controlled terminology would require additional mapping tables and validation logic.

7.4 ROLE OF AI IN PROTOCOL DIGITALIZATION

This work demonstrates that AI systems, while not capable of standalone table extraction, are highly effective for semantic interpretation when provided with accurately extracted table structures. The various tasks present different demands. Some steps benefit from using advanced reasoning models while others require large context windows. The combination of algorithmic table extraction with AI-powered semantic analysis leverages the strengths of both approaches.

The use of multiple AI models for different tasks reflects the current landscape where no single model excels at all tasks. This modular approach provides flexibility as model capabilities evolve.

The field of AI has evolved considerably since these tests were performed in June 2025, and improved results may be anticipated, using the same semantic interpretation algorithm.

7.5 LIMITATIONS

ProtocolMiner has the following limitations:

- The algorithm currently does not handle constraints on the end time of an activity, although the start time of an activity can be tied to the end of another activity.
- No attempt has been made to assign vocabulary codes to activities.
- There is currently no way to represent conditional visit offsets, for example, if an early termination visit should be performed within 5 days if the patient withdraws consent, but within 15 days if disease progression is detected.
- ProtocolMiner timelines will not include visits defined outside of the SoA table, i.e., in at-large text or footnotes.
- Start, stop, and applicability conditions for activities are presented as descriptive phrases rather than in formal logic. For example, "only applies to women of child-bearing potential" is presented as a phrase, not executable logic.

Specific limitations include the inability in FHIR Release 4 and 5 to express "during" relationships for activities occurring within a visit. ProtocolMiner's USDM lack biomedical concept integration and controlled terminology mapping.

7.6 FUTURE DIRECTIONS

Several extensions would enhance ProtocolMiner. Vocabulary coding using standard terminologies would improve semantic interoperability and enable USDM biomedical concept integration. Formal logic representations for conditions would enable computational verification of protocol adherence.

The methodology could also create activity-focused timelines useful for resource scheduling and costing. Integration with electronic data capture systems could enable automated case report form generation. Extension to FHIR Release 6 features would enable more precise timing relationship representation.

8. Conclusions

The field of *ab initio* clinical trial protocol structuring is advancing, driven by the need for increased efficiency, data interoperability, and improved patient outcomes. Initiatives from ICH, CDISC, HL7, TransCelerate, European Medicines Agency, and the US Food and Drug Administration are pushing this evolution. However, it will take time before SoAs are routinely authored in computable formats. This work presented ProtocolMiner, a practical solution for transforming legacy clinical trial Schedules of Activities into interoperable digital formats with specific mappings to both FHIR and USDM standards. The approach of structured AI-powered semantic analysis and standards-based transformation achieved high accuracy across 29 SoAs, with 100% accuracy on 22 cases and minor errors on the remaining seven.

The approach provides a useful bridge from legacy protocols and M11 to the digital models of the future, which promise to accelerate collaboration among stakeholders, streamline trial execution processes, reduce manual effort, and free up resources for other critical trial activities. By supporting both FHIR and USDM, ProtocolMiner enables organizations to integrate legacy protocol content into whichever ecosystem best suits their needs. Healthcare systems can leverage FHIR output for clinical workflow integration, while sponsors and contract research organizations can use USDM output for study design automation and regulatory submissions.

Conflicts of Interest Statement:

The author declares no conflict of interest in relation to this work.

Funding Statement:

Funding for this work was provided by MITRE Corporation.

Acknowledgments:

The author is grateful to Kayla Williams of Takeda Pharmaceuticals for insights and suggestions that significantly improved this work; Dave Iberson-Hurst and Kerstin Forsberg at Data4Knowledge for their openness and collaborative spirit; Geoff Low for suggestions on open-ended recurrence and FHIR representations; Hugh Glover for Vulcan leadership; and colleagues for their continuous support, suggestions, and feedback.

References:

1. Faro Health. Clinical Data Management Insights: Using Digital SoA to Solve Modern Clinical Trial Challenges. Published September 24, 2023. Accessed February 19, 2026. https://farohealth.com/blog/clinical-data-management-insights-using-digital-soa-to-solve-modern-clinical-trial-challenges

2. TransCelerate BioPharma Inc. Digitizing the clinical protocol: small steps for seismic change. Clinical Leader. Published January 30, 2024. Accessed March 3, 2026. https://www.paconsulting.com/newsroom/clinical-leader-digitizing-the-clinical-protocol-30-january-2024

3. CDISC. Unified Study Definitions Model (USDM). Accessed July 2025. https://www.cdisc.org/ddf

4. TransCelerate BioPharma Inc. Digital Data Flow Initiative. Accessed July 2025. https://www.transceleratebiopharmainc.com/initiatives/digital-data-flow/

5. International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. ICH M11 Template. Accessed July 2025. https://database.ich.org/

6. Data4Knowledge. USDM Protocol [GitHub repository]. Accessed July 2025. https://github.com/data4knowledge/usdm_protocol

7. Health Level Seven Vulcan Accelerator. Accessed July 2025. https://www.hl7vulcan.org/

8. Health Level Seven International. FHIR Implementation Guide: Vulcan Schedule of Activities. Accessed July 2025. https://hl7.org/fhir/uv/vulcan-schedule/

9. Richardson, A. Representing Clinical Study Schedule of Activities as FHIR Resources: required characteristic attributes. J Soc Clin Data Manag. 2024;4(2). doi:https://doi.org/10.47912/jscdm.266

10. Richardson A, Genyn J, Choi S, et al. Clinical trial Schedule of Activities specification using Fast Healthcare Interoperability Resources definitional resources: mixed methods study. JMIR Med Inform. 2025;13:e71430. doi:10.2196/71430.

11. Genyn J, Choi S, Richardson A, et al. Developing FHIR-based activity libraries to support clinical trial design and execution. J Soc Clin Data Manag. 2025;4(3):e423.

12. Shin K, Han S, Choe H, et al. Automated protocol templates with efficient Schedule of Activities table generation for healthy volunteer trials. Contemp Clin Trials Commun. 2025;46:101498. doi: 10.1016/j.conctc.2025.101498

13. OpenStudyBuilder. OpenStudyBuilder: an open-source project for clinical study specifications. OpenStudyBuilder website. https://www.openstudybuilder.com. Accessed March 3, 2026.

14. Faro Health. Faro: the AI platform for clinical development, including digital study design and protocol document authoring. Faro Health website. https://farohealth.com. Accessed March 3, 2026.

15. Kestemont J, Rogiers S. Development of USDM through translation of human-readable protocols. Presented at: CDISC + TMF Europe Interchange; April 2024.

16. Babaeipour F, Maleki F, Chlipala M, et al. AI-assisted protocol information extraction for improved accuracy and efficiency in clinical trial workflows. arXiv. Preprint published January 18, 2026. doi:10.48550/arXiv.2602.00052.

17. Kramer, M. The hidden challenge of digitizing clinical trials: why converting legacy protocols is harder than you think. Medium. Published July 25, 2025. https://medium.com/@kramermark/the-hidden-challenge-of-digitizing-clinical-trials-why-converting-legacy-protocols-is-harder-than-296e5088a6ff

18. Kramer, M. I tested 12 "best-in-class" PDF table extraction tools, and the results were appalling. Medium. Published August 2, 2025. https://medium.com/@kramermark/i-tested-12-best-in-class-pdf-table-extraction-tools-and-the-results-were-appalling-f8a9991d972e

19. Singer-Vine, J., pdfplumber contributors. pdfplumber (version 0.11.9) [computer software]. 2026. GitHub.

https://github.com/jsvine/pdfplumber

20. OpenCV Team. opencv-python (version 4.13.0.92) [computer software]. Python Package Index. https://pypi.org/project/opencv-python/

21. Health Level Seven International. FHIR Implementation Guide: Vulcan Schedule of Activities. Accessed July 2025. https://hl7.org/fhir/uv/vulcan-schedule/

22. HL7 International. FHIR Shorthand (FSH). Accessed July 2025. https://hl7.org/fhir/uv/shorthand/

23. FHIR Foundation. SUSHI: FSH Compiler [GitHub repository]. Accessed July 2025. https://github.com/FHIR/sushi

