SPECIFICATION

METHOD AND SYSTEM FOR MULTI-LEVEL AUTOMATED STATUS MANAGEMENT IN CLINICAL TRIAL WORKFLOWS

TECHNICAL FIELD

The present invention relates generally to computer-implemented systems for managing clinical trial workflows, and more specifically to methods and systems for automated multi-level status management with cross-entity validation in clinical trial environments.

BACKGROUND OF THE INVENTION

Clinical trials involve complex interactions between multiple entities including studies, protocol versions, investigational sites, and study participants. Each entity maintains independent lifecycle states but operates under interdependent business rules that must comply with regulatory requirements from agencies such as the Food and Drug Administration (FDA) and International Council for Harmonisation (ICH).

Traditional clinical trial management systems manage these entities in isolation, leading to several technical challenges:

1. Status Inconsistencies: Manual status updates create temporal gaps where related entities maintain inconsistent states, violating business rules and regulatory requirements.

2. Compliance Violations: Lack of automated validation allows invalid state transitions that violate FDA/ICH guidelines, creating regulatory risks.

3. Operational Inefficiencies: Manual coordination of multi-entity workflows creates bottlenecks and increases operational overhead.

4. Data Integrity Issues: Absence of real-time cross-entity validation can result in data inconsistencies that compromise trial integrity.

Existing systems such as Medidata Rave, Veeva Vault CTMS, and Oracle Clinical provide basic status management but lack integrated multi-level status computation with automated cross-entity validation. These limitations create significant operational challenges in large-scale, multi-site clinical trials.

There exists a need for a computer-implemented system that can automatically manage and coordinate status across multiple clinical trial entities while ensuring regulatory compliance and maintaining data integrity.

SUMMARY OF THE INVENTION

The present invention addresses the above-described problems by providing a computer-implemented system for automated multi-level status management in clinical trial workflows. The system includes a hierarchical status computation engine, cross-entity validation framework, event-driven state synchronization, and embedded compliance rule engine.

In accordance with one aspect of the invention, a computer-implemented method for automated status management in clinical trial systems comprises: maintaining a plurality of clinical trial entities in a database, each entity having an independent status lifecycle; establishing dependency relationships between entities based on clinical trial business rules; automatically computing entity status based on states of dependent entities using a hierarchical status computation algorithm; validating status transitions across entity boundaries using cross-entity validation rules; and synchronizing status changes across dependent entities in real-time using event-driven propagation.

In accordance with another aspect of the invention, the hierarchical status computation algorithm gathers current states of all dependent entities, applies priority-based computation rules specific to clinical trial workflows, and determines optimal status considering regulatory compliance requirements.

In accordance with yet another aspect of the invention, cross-entity validation includes constructing a real-time dependency graph of related entities, validating proposed status transitions against dependency constraints, and preventing transitions that would violate regulatory compliance rules.

The invention provides technical advantages including reduced system complexity through automated status management, enhanced data integrity via cross-entity validation, improved performance through event-driven architecture, and scalability supporting large-scale multi-site trials.

DETAILED DESCRIPTION OF THE INVENTION

Referring to the figures, the present invention will be described in detail.

The multi-level automated status management system comprises several interconnected components that work together to provide comprehensive clinical trial workflow management:

1. Hierarchical Status Computation Engine: This component automatically determines the status of clinical trial entities based on the current states of their dependent entities. The computation follows a hierarchical approach where higher-level entities (studies) derive their status from lower-level entities (sites, participants, protocol versions).

2. Cross-Entity Validation Framework: This component ensures that status transitions across different entity types maintain consistency and comply with regulatory requirements. It constructs real-time dependency graphs and validates proposed changes against business rules.

3. Event-Driven State Synchronization: This component propagates status changes across dependent entities in real-time, ensuring that the entire system maintains consistency without manual intervention.

4. Compliance Rule Engine: This component embeds FDA and ICH regulatory requirements directly into the system logic, preventing invalid status transitions that could compromise regulatory compliance.

The system operates through the following detailed process:

STEP 1: Entity State Management

Each clinical trial entity (Study, Protocol Version, Site, Participant) maintains its own lifecycle state in the database. States include but are not limited to: DRAFT, UNDER_REVIEW, APPROVED, ACTIVE, SUSPENDED, COMPLETED, TERMINATED.

STEP 2: Dependency Relationship Establishment

The system establishes dependency relationships between entities based on clinical trial business rules:

- Studies depend on Protocol Versions for regulatory approval status
- Studies depend on Sites for operational readiness
- Sites depend on Protocol Versions for activation capability

- Participants depend on Sites for enrollment capability

STEP 3: Hierarchical Status Computation

When computing entity status, the system:

a. Identifies all dependent entities that affect the target entity's status
b. Retrieves current states of all dependent entities
c. Applies priority-based computation rules specific to clinical trial workflows
d. Determines the optimal status considering regulatory compliance requirements

STEP 4: Cross-Entity Validation

Before allowing any status transition, the system:

a. Constructs a real-time dependency graph of all related entities
b. Validates the proposed status transition against dependency constraints
c. Checks regulatory compliance rules to ensure the transition is permitted
d. Prevents transitions that would violate FDA or ICH guidelines

STEP 5: Event-Driven Synchronization

a. When a status change is approved:
b. The system detects the status change event
c. Identifies all dependent entities that may be affected
d. Computes required status updates for dependent entities
e. Executes synchronized status updates across all affected entities

f. Maintains audit trails for regulatory reporting

FIGURES

FIG. 1 illustrates the overall system architecture of the multi-level automated status management system, showing the interconnection between clinical trial entities (Study, Protocol Version, Site) and the three main processing engines (Hierarchical Status Computation, Cross-Entity Validation, and Event-Driven State Synchronization).

FIG. 2 shows the detailed components of the hierarchical status computation engine, including the dependency resolver, priority-based rule engine with four priority levels (Regulatory Compliance, Safety Considerations, Operational Readiness, Administrative Completeness), and the status calculator algorithm.

FIG. 3 depicts the cross-entity validation framework workflow, illustrating the complete validation process from request input through dependency graph building, constraint validation, regulatory compliance checking, and final validation result output.

FIG. 4 illustrates the event-driven state synchronization process, showing how status change events are detected, analyzed for impact, and synchronized across multiple dependent entities using distributed transaction management with rollback capabilities.

FIG. 5 shows the compliance rule engine architecture, detailing the regulatory rule repository (FDA CFR and ICH guidelines), business rule repository (clinical trial workflow and entity relationship rules), and the rule execution engine with its four-stage processing pipeline.

These figures demonstrate the comprehensive technical implementation of the multi-level automated status management system, providing detailed architectural views of each major component and their interactions to ensure regulatory compliance and data integrity in clinical trial workflows.

DETAILED TECHNICAL ALGORITHMS

Algorithm 1: Hierarchical Status Computation

The hierarchical status computation algorithm operates through the following steps:

Step 1: Dependency Context Building

- ✓ Query database for all entities that affect the target entity's status
- ✓ Retrieve current lifecycle states of all dependent entities
- ✓ Build dependency relationship map with constraint definitions
- ✓ Establish priority hierarchy based on regulatory requirements

Step 2: Priority-Based Rule Application

- Level 1 (Highest): Regulatory compliance rules (FDA 21 CFR, ICH-GCP)
- Level 2: Safety consideration rules (adverse events, risk assessments)
- Level 3: Operational readiness rules (site activation, staff training)
- Level 4 (Lowest): Administrative completeness rules (documentation, contracts)

Step 3: Status Determination

- Evaluate each priority level sequentially from highest to lowest
- If all rules at a priority level are satisfied, proceed to next level
- Final status determined by highest achievable level with all rules satisfied
- Generate justification report with detailed rule evaluation results

Algorithm 2: Cross-Entity Validation

The cross-entity validation algorithm ensures status transition validity:

Step 1: Dependency Graph Construction

- Create nodes for all entities involved in the status transition
- Define edges representing dependency relationships and constraints
- Populate nodes with current entity state information
- Update graph with real-time status information

Step 2: Constraint Validation

- Evaluate business rules against proposed status transition

- Check entity relationship constraints (one-to-many, many-to-many)
- Validate workflow sequence requirements
- Assess impact on dependent entities

Step 3: Regulatory Compliance Verification

- Apply FDA 21 CFR regulations relevant to the transition

- Verify ICH-GCP guideline compliance

- Check institutional review board (IRB) requirements

- Validate data integrity and audit trail requirements

Step 4: Validation Result Generation

- Compile validation results with pass/fail status for each rule

- Generate warnings for non-critical violations

- Provide recommendations for issue resolution

- Create audit trail entry for regulatory reporting

Algorithm 3: Event-Driven State Synchronization

The synchronization algorithm manages multi-entity status updates:

Step 1: Event Detection and Analysis

- Monitor entity status change events in real-time

- Classify events by priority and impact scope

- Determine synchronization mode (synchronous/asynchronous)
- Validate event source and user authorization

Step 2: Impact Analysis

- Identify all entities affected by the status change
- Determine required status updates for each affected entity
- Calculate update order based on dependency hierarchy
- Estimate processing time and resource requirements

Step 3: Transaction Management

- Create distributed transaction context for all updates
- Acquire locks on all entities requiring modification
- Execute updates in dependency order with intermediate validation
- Implement rollback capability for failure recovery

Step 4: Synchronization Execution

- Apply status updates across all affected entities
- Validate each update against business and regulatory rules
- Generate audit trail entries for all changes
- Send notifications to relevant stakeholders
- Commit transaction and release entity locks