TABLES AND ATTRIBUTES IN 3P3 THEORY

Technical-Operational Document for FileMaker Implementation

PREMISE: CLARIFYING THE PRACTICAL IMPLEMENTATION OF 3P3 THEORY

The implementation of information systems based on 3P3 theory requires clarity on fundamental principles. Analysis of concrete implementations often reveals **partial understanding** of 3P3 ontology, offering the opportunity to refine basic concepts.

Typical Fields in 3P3 Implementations: Task ID, Client, Responsible, TextJoin, MainProcess, Process, SubProcess, Process ID, CLI_Code, CLI_Type, TSK_Code

The Recurring Problem: Confusion between ontological level (theory) and practical manifestation (software implementation).

---

1. FUNDAMENTAL PRINCIPLE: "PROCESS = TABLE"

1.1 The Ontological Essence

Every process is a table. Fields are the process attributes. Elementary action = information insertion in table field.

1.2 The Implementation Challenge

Pure Theory: 1 table = 1 process → 150+ tables **Elegant Solution:** 1 table = 1 macroprocess

Macroprocess Architecture:

- **TASK:** Management of all elementary processes
- **PROJECT:** Management of all composite processes
- **REQUEST:** Management of all client requests
- **RESOURCE:** Management of all company resources

Each macroprocess maintains **3P3 ontological purity** through specific fields and hierarchical DNA, solving implementation complexity without compromising theory.

---

2. CLARIFICATION: TASK vs LOG vs PROCESS

2.1 Correct Definitions

TASK (Template Process)

- **Nature:** Macro process containing task typologies

- **Function:** Replicable model with standard characteristics
- **Example:** "Customer Call Management" with default responsible, estimated lead time
- **DNA:** TSK25001 (unique process code)

LOG (Registration Process)

- **Nature:** Specific process that records every executed activity
- **Function:** Total traceability without additional effort
- **Example:** "Phone call executed by Luca to client X on 07/08/25 at 11:00"
- **DNA:** LOG25002 linked to TSK25001

PROCESS (Ontological Entity)

- **Nature:** Unifying structure of Task and Log
- **Function:** Ontological container for operational instances
- **DNA:** Unique hierarchical coding

2.2 The Fundamental Relationship

```
PROCESS (Ontology)
├── TASK (Template/Model)
└── LOG (Instance/Execution)
```

---

3. CONFUSION TASK_ID vs PRX_ID: ONTOLOGICAL CORRECTION

3.1 The Identified Problem

Typical implementations create conceptual duplication:

- TASK_id and PRX_id represent the same ontological entity
- Lack of DNA coherence from beginning to end

3.2 3P3 Solution

Unifying Principle: Every process instance has unique hierarchical DNA

- **TSK25001** = specific instance of Task process
- **Elimination** of terminological duplications
- **Maintenance** of ontological coherence throughout the application

---

4. THE ELEMENTARY BUILDING BLOCK: INFORMATION

4.1 Information as Manageable Entity

3P3 FUNDAMENTAL PRINCIPLE: In our conceptual system, the elementary building block we want to control is **INFORMATION**. We don't manage "things" or "objects" - we manage **information that becomes entities**.

Elementary information represents the most granular level of systemic control. It is the atomic unit from which all organizational complexity emerges.

4.2 RED: The Elementary Instance

Concrete Example:

- **COLOR** = field/attribute (color management process)
- **RED** = information instance (specific elementary data)
- **CELL** = ontological container (database + interface)

RED is not simply data - it is an **informative entity** that possesses:

- **Unique identity** (DNA: CLR\RED\001)
- **Relations** with other color information
- **Evolutionary history** (who created it, when, why)
- **Transformative potential** (can become part of more complex processes)

4.3 Fractal Principle of Information

INFINITE SCALABILITY: Elementary information aggregates following the fractal principle:

```
INFORMATION (RED)
   ↓ aggregates with other information
DATA ROW (color + hue + saturation)
   ↓ aggregates with other rows
INFORMATION SHEET (complete color range)
   ↓ aggregates with other sheets
COMPLEX FILE (final color chart)
   ↓ aggregates with other files
CORPORATE SYSTEM (complete coloring management)
```

Relating to the world of processes:

- **Elementary actions** aggregate into **simple processes**
- **Simple processes** aggregate into **composite processes**
- **Composite processes** aggregate into **parent processes**
- **Parent processes** aggregate into **corporate systems**

4.4 STR-REL-ETY Applied to Information

EVERY ELEMENTARY INFORMATION manifests the ontological triad:

STRUCTURE (STR)

- Intrinsic form of information (RED has specific properties)
- Constitutive attributes (hue, saturation, brightness)
- Composition rules (how it combines with other information)

RELATIONS (REL)

- Input connections (where RED information comes from)
- Output connections (where it goes and how it transforms)
- Interdependencies (relations with other color information)

ENTITY (ETY)

- Concrete manifestation of information in the system
- Transformative capacity (RED can become strand, then chart)
- Ontological persistence (maintains identity through transformations)

4.5 Practical Implementation of the Building Block

In a 3P3 information system:

- **Database field** = container for elementary information
- **Data entry** = elementary action of information creation
- **Record** = aggregation of correlated elementary information
- **Table** = template to manage information typologies
- **Database** = ecosystem of interconnected information

KEY INNOVATION: Elementary information is not passive - it is **self-aware:**

- Knows where it comes from (parent process)
- Knows where it can go (potential transformations)
- Knows how to relate (relationship rules)
- Knows when to evolve (trigger conditions)

4.6 Granular Control

COMPETITIVE ADVANTAGE: By controlling information at elementary level, the system can:

- **Trace every micro-transformation** without additional effort
- **Predict emergent behaviors** from information composition

- **Automatically optimize** processes based on informative patterns
- **Guarantee coherence** from elementary information to overall system

Example: When we insert "RED" in the system, automatically:

- Unique DNA is generated for that informative instance
- Relations with correlated information are activated
- The elementary action is registered in LOG
- Successive possible transformations are enabled

NOTE ON USER GRANULARITY: While maintaining theoretical control over elementary information, the significant entity for the end user is the **PROCESS INSTANCE** (a database row). This balancing avoids informative overload ("too much information equals no information") while maintaining 3P3 traceability power. The user works with significant processes, the system automatically traces at granular level.

---

5. COMPLETE LIFECYCLE: FROM THEORY TO REALITY

5.1 From Customer Need to Finished Product

CONCRETE APPLICATION of 3P3 theory: The path from customer need to finished product demonstrates how elementary information transforms into business value through intelligent concatenation of processes.

Zero Point: Customer Need

- **Customer:** Wants to launch new coloring line
- **Trigger:** Request to commercial technician
- **Origin DNA:** RCH25001 (first generated processual entity)

Automatic Processual Cascade:

```
RCH (Customer Request)
   ↓ automatically generates
PRJ (Development Project)
   ↓ activates specific processes
SWA (Strand Sampling)
   ↓ if approved generates
JOB (Production Order)
   ↓ culminates in
DEL (Customer Delivery)
```

5.2 The Strand: Concrete Example of Informative Evolution

The Evolution of RED Information:

PHASE 1: Elementary Information

- **Input:** Customer shows desired color reference
- **Process:** Technical color analysis → RED as coded information
- **DNA:** CLR\RED\REF\001

PHASE 2: Processed Information

- **Input:** RED + technical parameters (base hair, pigment percentages)
- **Process:** SWA (sampling) → specific formulas to obtain RED
- **DNA:** SWA\CLR\RED\FORMULA\001

PHASE 3: Materialized Information

- **Input:** RED formula + production process
- **Process:** Physical creation of strand with RED color
- **DNA:** SWA\CLR\RED\STRAND\001

PHASE 4: Validated Information

- **Input:** RED strand + customer approval
- **Process:** Specific confirmation for series production
- **DNA:** APR\CLR\RED\APPROVED\001

PHASE 5: Produced Information

- **Input:** Approved RED specifications + order 100 charts
- **Process:** Production 1000 RED strands (10 per chart)
- **DNA:** PRD\CLR\RED\BATCH\001

5.3 Automatic and Intelligent Concatenation

3P3 AUTOMATISM: Every completed process **automatically generates** the necessary successive processes:

SWA Completed → Automatic Triggers:

- **If approved:** Automatically generates JOB (production order)
- **If modifications requested:** Reactivates SWA with new parameters
- **If rejected:** Generates alternatives or closes project

Preventive vs Actual Control:

- **SWA Preventive:** 3 days, 1 technician, 5 estimated attempts

- **SWA Actual:** 2.5 days, 1 technician, 3 real attempts
- **Learning:** System updates future estimates for similar colors

5.4 Infinite Traceability and Feedback Loop

DNA that Follows the Product:

Every RED strand in the chart maintains complete traceability:

```
Customer Chart ABC → Position 15 → RED Strand
Complete DNA: RCH25001\PRJ25045\SWA25123\CLR\RED\STRAND\015
```

The strand "remembers":

- Who requested it (client ABC)
- When it was developed (07/08/25)
- Who created it (technician Marco)
- How many attempts were needed (3)
- Exact formula used
- Real costs sustained

Post-Delivery Lifecycle:

- **Hairdressers:** Use strands for client consultations
- **Feedback:** Satisfaction/problems return to the system
- **Evolution:** New requests based on RED success
- **Maintenance:** Possible corrections/updates

Emergent Intelligence:

The system automatically learns:

- **Success patterns:** Which colors work better
- **Optimizations:** How to reduce development times
- **Predictions:** Emerging color trends
- **Quality:** Correlations between process and final result

5.5 The Value of Complete Traceability

CONCRETE ADVANTAGES:

For Business:

- Real costs of every developed color

- Effective vs estimated times for continuous improvement
- Specific ROI for every client project
- Knowledge base that grows automatically

For Quality:

- Guaranteed repeatability of approved colors
- Rapid troubleshooting if post-delivery problems
- Batch control to guarantee consistency
- Continuous improvement based on real data

For Innovation:

- Complete repertoire of tested solutions
- Scientific base for new developments
- Rapid personalization for new clients
- Product evolution guided by real feedback

RESULT: Elementary information "RED" transforms into **structured business knowledge** that generates continuative value over time.

---

6. EXISTENTIAL ATTRIBUTES: COMMON PROCESS CHARACTERISTICS

6.1 The Universal Ontological Matrix

3P3 CORE INNOVATION: Every process/table shares universal existential attributes that define the ontological nature of the entity. This standard matrix represents the **informative DNA** that makes every process traceable, relational and self-evolutionary.

IMPLEMENTATION NOTE: This matrix represents **3P3 theoretical completeness**. In practical implementation, start with essential attributes and gradually add other attributes based on real operational needs. Organic growth maintains usability while preserving future scalability.

6.2 Existential Attribute Categories

IDENTITY MANAGEMENT

Identity attributes guarantee uniqueness and infinite traceability:

- **ety_id:** Unique entity code (auto-generated)
- **parent_id:** Generator process (hierarchical relation)
- **sequence:** Position in structure (ontological ordering)
- **DNA_code:** Complete hierarchical coding (PRJ25001\RCH25045\TSK2617)

- **speaking_code:** Intuitive code for users
- **qr_code:** Automatic physical identification

CLASSIFICATION MANAGEMENT

Classification attributes define processual nature:

- **entity_type:** Process typology (TSK, PRJ, RCH, OFC, etc.)
- **category:** Primary functional categorization
- **sub_category:** Secondary specialized categorization
- **process_nature:** 3P3 nature (STR-REL-ETY)
- **complexity_level:** Level of processual complexity
- **automation_degree:** Degree of possible automation

COMMUNICATION MANAGEMENT

Communication attributes manage informative flow:

- **input_source:** Origin of processed information
- **output_destination:** Destination of produced information
- **communication_channel:** Transmission medium used
- **content_type:** Nature of content (data, documents, decisions)
- **information_format:** Standard exchange format
- **translation_rules:** Format conversion rules

RESPONSIBILITY MANAGEMENT

Responsibility attributes define processual ownership:

- **creator_id:** Who created the process (origin traceability)
- **responsible_id:** Who is responsible for execution
- **approver_id:** Who approves the final result
- **controller_id:** Who verifies conformity
- **stakeholder_list:** List of process stakeholders
- **escalation_rules:** Automatic escalation rules

TEMPORAL MANAGEMENT

Temporal attributes control processual evolution:

- **created_at, updated_at, deleted_at:** Existential timestamps
- **date_start_forecast, date_end_forecast:** Temporal planning

- **date_start_real, date_end_real:** Effective temporal execution
- **cycle_time:** Optimal time necessary for execution
- **lead_time:** Standard crossing time
- **deadline_hard, deadline_soft:** Critical and flexible deadlines

STATE MANAGEMENT

State attributes control processual evolution:

- **status_entity:** Current state (TO_DO → IN_PROGRESS → DONE)
- **trigger_entity:** Rules for state transitions
- **completion_percentage:** Process advancement percentage
- **quality_score:** Execution quality evaluation
- **priority_level:** Dynamic priority level
- **blocking_factors:** Factors that prevent advancement

PERFORMANCE MANAGEMENT

Performance attributes measure processual effectiveness:

- **resource_consumption:** Resources effectively consumed
- **efficiency_ratio:** Expected/real performance ratio
- **error_count:** Number of detected errors
- **rework_instances:** Number of necessary reworks
- **customer_satisfaction:** Stakeholder satisfaction evaluation
- **improvement_suggestions:** Automatic improvement suggestions

6.3 Self-Evolution Principle

PATENTABLE CHARACTERISTIC: Existential attributes are not static but **self-evolutionary**. The system learns from usage patterns and automatically suggests:

- **Time optimization** based on execution history
- **Block prediction** based on recurring patterns
- **Responsibility suggestions** based on demonstrated competencies
- **Progressive automation** of standardizable repetitive actions

6.4 Relationship Matrix

RELATIONAL INNOVATION: Every attribute can relate with homologous attributes of other processes, creating a **self-reinforcing ontological network:**

- **Structural relations:** Parent-child, sibling, dependency
- **Temporal relations:** Before-after, concurrent, conditional
- **Functional relations:** Input-output, trigger-response, feedback
- **Semantic relations:** Similar, opposite, complementary

---

7. LOG SYSTEM: AUTOMATIC TRACEABILITY

7.1 Automatic Functioning

Every user click = traced elementary action

- User logs in → essential questions: what must I do? how? when? with which resources?
- Every field entry → automatically registered in LOG
- End activity → comparison expected vs obtained

7.2 Automatic Daily Report

Objective: Every operator gets activity report without additional effort

- **Digital activities:** Automatically traced
- **External activities:** Registered during/after execution
- **Performance control:** EXPECTED TIME / TIME USED

7.3 Example of Processual LOG

```
LOG25003: Red Color Management
- Responsible: Simona (Designer)
- DNA: CLR\RED\VALIDATION\001
- Action: Color approval for sampling
- Trigger: SAS completed → generates strand detail
- Parent: PRJ25351\PROTO\PRT001
- Timestamp: 07/08/25 14:30:15
```

---

8. IMPLEMENTATION PRAGMATISM: BALANCING THEORY AND PRACTICE

8.1 Macroprocess Architecture

Macroprocess Solution:

- **Complexity reduction** from 150+ to <10 main tables
- **Ontological purity maintenance** through specific fields
- **Hierarchical DNA preserved** for every instance

- **Optimized performance** for relational database systems

8.2 Maintaining 3P3 Compass

Irrenunciable Principles:

1. **Hierarchical DNA** for every entity
2. **Standard existential attributes**
3. **Automatic LOG traceability**
4. **Ontological coherence** from beginning to end

Acceptable Compromises:

1. **Macroprocess aggregation** maintaining ontological coherence
2. **Relationship optimization** for database performance
3. **Pragmatic interfaces** without losing 3P3 essence

---

9. CONCLUSIONS: "HOLY" DOCUMENT FOR 3P3

This document establishes the **fundamental principles** of 3P3 theory applied to information systems, balancing **ontological purity** with **implementation pragmatism**.

Theoretical Value: A reference document that maintains 3P3 essence while addressing practical challenges of the real world.

Practical Value: Guide for concrete implementations that respect fundamental ontological principles.

Future Value: Solid bases for developing next-generation native 3P3 systems.

"3P3 is an essential, very simple concept, but when applied to complex reality, if we don't always stay attentive to the compass of principles, we get lost." - Luca Meggiolaro