GAP ANALYSIS - THE BRIDGE SPECIFICATION

Guide for Interface Validation and Documentation Update

INTRODUCTION

This document guides the validation of proposed interfaces and the update of technical documentation. Decisions made together will define the final implementation.

PHASE 1: INTERFACE VALIDATION - DECISIONS TO MAKE TOGETHER

HTML Prototypes Provided (PROPOSALS)

1. **process_manager_3p3.html** - Template definition and process structure
2. **instance_manager_3p3.html** - Runtime instance management
3. **pho-complete-walkthrough.md** - Complete PHO process example

These are WORKING PROPOSALS requiring validation for:

- Complete operational flow
- Necessary/superfluous fields
- Navigation logic
- Integration between modules

FUNDAMENTAL CLARIFICATIONS

1. PROCESS/INSTANCE MANAGER INTEGRATION The two modules are INTEGRATED:

- Process Manager: defines HOW the process presents itself
- Instance Manager: MANIFESTS the defined interface

IMPORTANT: In Process Manager we must define the PRESENTATION FORM:

- Complex processes → Tables with tabs and sub-tabs Example: "KOOL TOOL Management" (root) contains everything └── Product Management ├── BOM (Bill of Materials) ├── TDS (Technical Sheet) └── MAD (Technical Drawings)
- Simple processes → Single form Example: PHO (Phone Call Management)

2. THE 10 DOMAINS - Ontological Clarification The 10 domains are CATEGORIES of attributes, not different field types. Attribute management is ALWAYS THE SAME ontologically. Domains serve to:

- Classify the attribute
- Specialize its behavior
- NOT to create 10 different attribute types

Example: "caller_name" is always a TEXT attribute, but:

- Domain 1 (Identity): makes it searchable
- Domain 3 (Authorization): defines who can view it
- Domain 5 (Trigger): specifies what happens when it changes

3. WORKFLOW DESIGNER TO DECIDE WITH YOU: Depends on FileMaker capabilities

- Option A: Visual designer (if FileMaker supports it)
- Option B: Tabular configuration
- Consider: FileMaker is database + specialized interface manager

4. TEMPLATE VERSIONING QUESTION FOR YOU: How to manage template evolution?

- Do existing instances maintain original version?
- Automatic or manual migration?
- Is compatibility check necessary?

PHASE 2: TECHNICAL DOCUMENT UPDATE

After interface validation, update "The Bridge" with:

A. MANDATORY CORRECTIONS

1. **Nomenclature**: Processes→CMP, Entities→ETY, Logs→LOG
2. **DNA Format**: PRXYYNNNN (example: PHO250042, NOT PHO25001)
3. **CMP Structure**: Clarify is_template TRUE/FALSE usage

```sql
IF is_template = TRUE THEN use template_json
IF is_template = FALSE THEN use instance_json
```

B. NECESSARY INTEGRATIONS

1. Instance Manager Section (NEW)

- How interfaces defined in Process Manager manifest
- Management of simple forms vs complex tables
- Reference to HTML prototype for UI details

2. PHO Complete Example

- Integrate as Appendix A

- Verifiable example of complete system
- Shows CMP→ETY→LOG flow with triggers

3. Process Presentation Definition

- Hierarchical tree (KOOL TOOL Management → subsystems)
- Rules for forms vs tables
- Parent-child relationships

PHASE 3: DOCUMENTATION AND ROADMAP

Expected Final Document

"THE BRIDGE v2.0" must contain:

- 3P3 theoretical foundations
- Correct technical architecture
- Verifiable concrete examples
- Basis for future patent (not necessarily patent-ready)

Timeline and Priorities

REQUEST TO YOU:

- Define realistic timeline yourselves
- Identify implementation priorities
- Specify verifiable milestones
- Consider phases for MVP vs full system

PROPOSED WORK PROCESS

1. **YOU**: Validate HTML interfaces with us
2. **YOU**: Decide open points (workflow, versioning, etc.)
3. **YOU**: Update "The Bridge" document with decisions made
4. **US**: Verify updated document
5. **TOGETHER**: Proceed with implementation

EXPECTED OUTPUT

From your side:

1. Updated "The Bridge v2.0" document
2. Defined implementation timeline

3. Documented decisions on open points
4. Clear mapping: UI elements → Database operations

The updated document will undergo our final verification before proceeding with FileMaker implementation.

CRITICAL SUCCESS FACTORS

For Implementation

- Interfaces correctly map to CMP-ETY-LOG architecture
- DNA generation is collision-proof
- Workflow states are complete and reachable
- K coefficient calculation is accurate

For Documentation

- Clear enough for any FileMaker developer
- Complete enough for future patent filing
- Specific enough to avoid ambiguity
- Flexible enough for future extensions

OPEN QUESTIONS REQUIRING YOUR INPUT

1. **Technical Decisions**
   - FileMaker version and capabilities?
   - Server vs local deployment?
   - Multi-user concurrency strategy?
   - Backup and recovery procedures?
2. **Business Logic**
   - Which processes are MVP priority?
   - Required reporting capabilities?
   - Integration with existing systems?
   - User roles and permissions model?
3. **Documentation Preferences**
   - Preferred document format (Word/PDF/Markdown)?
   - Language (English/Italian/both)?
   - Level of technical detail needed?
   - Separate user vs developer documentation?

FINAL NOTES

- HTML interfaces are PROPOSALS to validate together
- Document must be clear IMPLEMENTATION BASE
- Future patent will require additional refinements
- Focus on: clarity, completeness, implementability

We await your feedback on:

- Points needing clarification
- Proposed timeline
- Next steps
- Any concerns or suggestions

Document Purpose: Introduction to the three deliverables:

1. process_manager_3p3.html
2. instance_manager_3p3.html
3. pho-complete-walkthrough.md

Next Action: Review these materials and schedule validation session.