

# A first attempt to model SWHAP á la IDEF-0

The diagram illustrates a data architecture with two main sections: **Collect + Curate** and **Present**.

**Collect + Curate Section:**

- Data Sources:** Dashboards, Crowded Promotes, and Good Sourcing feed into a **WAREHOUSE**.
- Warehouse:** Labeled "physical raw material collected (Legacy)". It "exposes" data to a **Catalog + Journal**.
- Catalog + Journal:** Receives "exposes" from the Warehouse and "exposes" to a **SW Depository**. It also receives "Crowdsourcing Dashboards" and "Crowded Promotes" (with a note "leads from physical material + log phenomena information").
- SW Depository:** A cylinder representing a data store. It "exposes" to a **Catalog + Journal** and "fetches" data from a **WORKBENCH**.
- WORKBENCH:** A table-like structure. It "fetches" data from the SW Depository and "exposes" to a **Journal**. It also receives "Interviews, articles, documents, notes, etc." and "clean up" actions (classify, tag, time stamp history, link into coherent VCS repository).
- Journal:** A document icon. It "exposes" to the WORKBENCH and "fetches" data from the SW Depository.
- HA L:** A dashed line labeled "possible variation archives" connects the SW Depository to the WORKBENCH.

**Present Section:**

- Processing:** **VEB** and **Applicator** feed into a **Processing Engine** (gear icon).
- Storage:** The Processing Engine "fetches" data from a **Stories** database (cylinder).
- Workbench:** A table-like structure. It "fetches" data from the Stories database and "exposes" to a **Wiki photo**.
- Wiki photo:** A box representing a photo/wiki entry. It "exposes" to a **Software the type**.
- Software the type:** A box representing a software type. It "exposes" to a **Wiki photo**.
- Interviews, articles:** A box representing content. It "exposes" to a **Wiki photo**.

Figure 2.1: Roberto's schema

## 2.1 Notation

- Le scatole sono attività, che trasformano documenti da destra verso sinistra.
- Quindi, le frecce verso destra rappresentano gli input dell'attività, e le frecce da sinistra gli output. Spesso gli output diventano input.

- Le frecce dal basso rappresentano i meccanismi caratteristici utilizzati dall'attività: sono *risorse*, fisiche o logiche, con le loro regole di utilizzazione.
- Le frecce dall'alto rappresentano i *ruoli* (leggi: insiemi di competenze) coinvolti nell'attività: i ruoli sono assegnati a persone (almeno in prima istanza, poi si potranno mettere automi più o meno intelligenti), e ovviamente la stessa persona può svolgere più ruoli, anche in attività diverse.
- Ho esteso la notazione per accogliere la nozione di entità astratta, denotata dall'avere il nome in corsivo (à la UML - sottolineato quando scritto a mano, secondo la tradizionale convenzione tipografica). Ad esempio, il processo *Collect* è astratto, come pure la *Notice* in input, lo strumento *Warehouse* e il ruolo *Collector*. Ho usato *X::Y* per indicare che *Y* è super-classe di *X*.
- Altra estensione: molteplicità di specializzazione. Un asterisco dopo il nome di una entità astratta indica che essa può essere specializzata in più modi diversi (ivi compresa la non specializzazione, ossia il non uso) per ogni sua occorrenza in un diagramma. Ad esempio, l'attività *Collect* di Fig. 2.2 ha un meccanismo astratto *Warehouse*, che in *CrawlCollect* (Fig. 2.4 non è usato).

## 2.2 Process elements

There is a lot to recover from section 1.1!

**AcSCnotice : Documento** Questo documento (*AcquirableSourceCode Notice*) riflette l'assunzione che il processo di acquisizione parte con una "notizia di reato": sto assumendo che ogni processo di acquisizione parta da un documento - una lettera di un donatore (esempio: la lettera di Attardi relativa al suo garbage collector, poi utilizzato da Google per Java), una call per raccogliere una certa categoria di SW (tipo quella che stiamo preparando per "5 software per 5 decenni"), una qualche dichiarazione d'interesse per (una parte di) un archivio SW - che in qualche modo identifica il SW da salvare.

**Collect : Attività** Questa si presenta concretamente in varie forme, una per tipo di SW da raccogliere. La figura 2.4 dettaglia (che parola!) quello per il SW archived.

**CrawlCollect : Attività** Questa attività, specializzazione dell'attività *Collect*, si ha quando il software che si vuole raccogliere è già disponibile in un repository moderno, tipo Git Lab. L'attività ha inizio con una specifica *Notice*, che contiene le coordinate necessarie per reperire il software nell'origine da cui prelevare i documenti. La figura 2.4 specifica solo gli elementi astratti di figura 2.2: la notizia è relativa a un SW Archived, la *Warehouse* non è coinvolta, e il *Collector* viene specializzato sia per gestire il processo di raccolta sia per gestire il Depository.

Il prossimo passo è di avere una descrizione del processo *CrawlCollect*, almeno a parole. Qui siamo a un livello di dettaglio per cui forse si può pensare a una notazione diversa (diagrammi di attività UML, Business Process Modeling Notation, BPMN,...).

**Collector : Ruolo (specializzazioni multiple)** Specifica le competenze necessarie per la raccolta del codice e la sua memorizzazione nel Depository, a futura memoria dello stato originale. Include, quando necessaria la trasformazione da un supporto obsoleto alla forma digitale adatta alla cura e al deposito nella Vault. Si sono individuati i seguenti sottotipi:

**CrawlEngineer :: Collector** : l'insieme delle competenze che permettono di gestire la parte di ricerca e trasferimento, in rete, di codice dall'origine al Depository.

**Curator : Ruolo**

**DepositEngineer :: Ruolo** : l'insieme delle competenze che permettono di facilitare l'accesso al Depository.

**Origin : Dato** Descrizione della collocazione originaria del codice, sia essa online o offline.

**PresentationDesigner: Ruolo**

**SC : Documento** : Ciò che si vuol salvare in primis (Source Code). Si presenta in diverse forme, che riflettono l'avanzamento del processo:

**OriginalSC** Il codice sorgente che si trova nell'Origin (vedi); è conveniente pensarlo di diversi tipi, per le conseguenze che ciò ha sul processo, sia rispetto al tipo di acquisizione

**pulled** in cui l'acquisizione parte da un centro, più o meno legato a SWH, interessato a raccogliere una data classe di SC;

**pushed** in cui l'iniziativa parte dalla periferia, sia individuale sia di gruppo,

sia rispetto all'accessibilità originale:

**online** già disponibile in forma digitale in rete

**offline** disponibile su altri supporti fisici, che necessitano di una fase di virtualizzazione.

Delle quattro combinazioni possibili dei due tipi, pulled/online è quella realizzata finora, come CrawledSC, da SWH. In 5S<sub>2</sub> (5softwarePer5stagioni) dovremmo sperimentare almeno anche qualcosa offline.

**DepositedSC** Ciò che si ha a disposizione una volta terminata la pertinente attività Collect, conservato a futura memoria per analisi approfondite - anche sulle scelte fatte dal Curator.

**CuratedSC** Questo è l'obiettivo primario di SWH: il codice sorgente di cui alla relativa Notice, a disposizione del pubblico nella Vault, con la sua evoluzione ricostruita a partire dalle informazioni originali, sotto la responsabilità del Curator.

**PresentableSC** Questo è l'obiettivo secondario di SWH, ma primario per 5S<sub>2</sub>, realizzato per caratteristiche particolari del codice (importanza dello stesso nella storia del SW) o per pressioni particolari di un committente (salvataggio fatto in occasioni particolari,

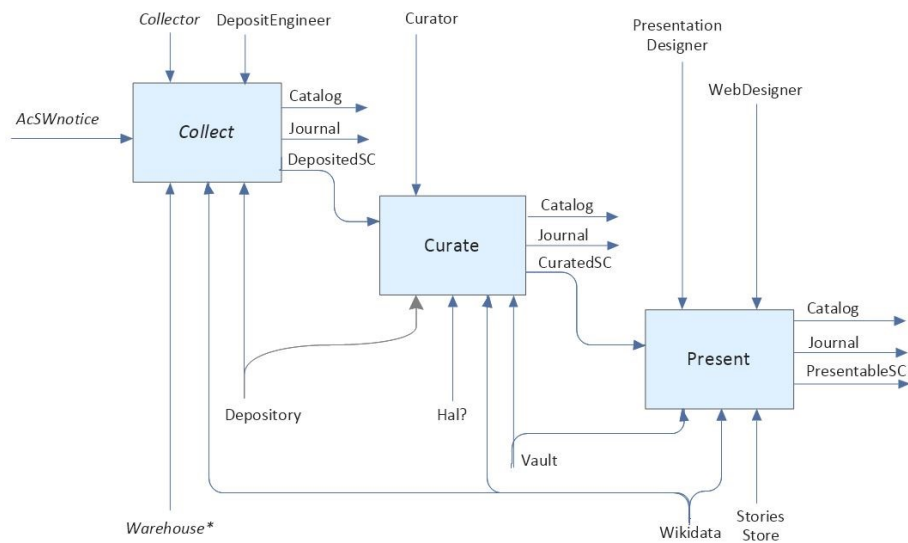


Figure 2.2: The overall process

come il 50-nario del corso di laurea in Scienze dell'Informazione): al codice sorgente salvato nel caveau come CuratedSW, si associa una pagina (un sito web?) che raccoglie documenti storici e illustrativi, guide d'uso ecc.

**Warehouse : Meccanismo** Spazio fisico in cui raccogliere, se possibile, in maniera controllata e durevole, il materiale fisico (tabulati, nastri, ecc) originale da salvare nella Vault. Tipicamente, è la sezione di un museo, con le sue regole di deposito e catalogazione.

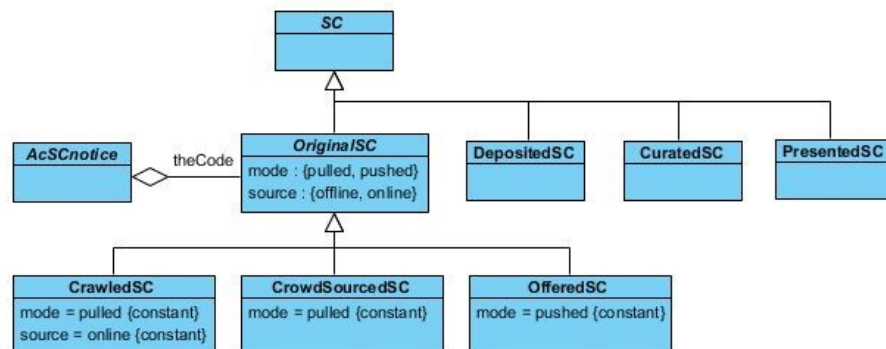


Figure 2.3: Class diagram - SC types

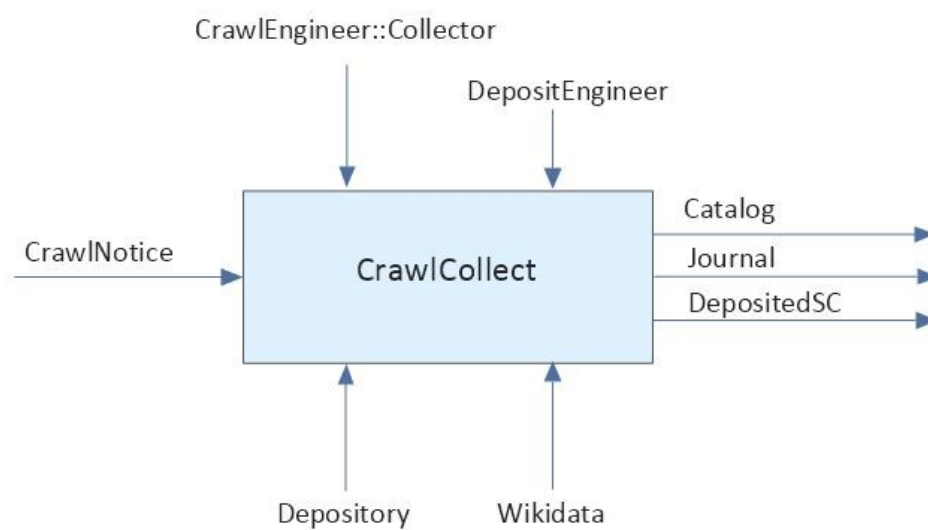


Figure 2.4: The CrawlCollect activity