

ÉTR · DOCUMENT OPÉRATIONNEL

ÉTR — Équation de Transformation Relationnelle

Introduction opérationnelle

Vyehn · UADIA · 13 juillet 2026

ÉTR · Recevoir → Qualifier → Comparer → Décider → Agir → Mettre à jour

Ce document réunit deux niveaux de lecture :

- Partie 1 — Introduction pour débutants : le cycle système, sans appareil formel.
- Partie 2 — Introduction opérationnelle et justification des cycles : les mêmes objets,

avec leurs définitions et leurs garanties.

Introduction opérationnelle pour débutants

1. L'idée en une phrase

L'ÉTR est un système qui reçoit une information, la situe dans un contexte, choisit quoi en faire, agit, puis mémorise ce qui s'est passé.

Son cycle général est :

recevoir → comprendre le contexte → comparer → décider → agir → mettre à jour la mémoire

Puis le cycle recommence.

Ce n'est donc pas une formule qui produit un résultat une seule fois : c'est une boucle d'évolution.

Les premières questions à poser :

1. quels objets sont manipulés ;
2. pourquoi sont-ils nécessaires ;
3. comment le cycle fonctionne ;
4. quelles quantités sont mesurées ;
5. quelles décisions sont autorisées ;
6. quelles limites empêchent une conclusion abusive.

2. Exemple d'onboarding en entreprise

Une entreprise embauche une nouvelle salariée, Lara. Elle reçoit cette information :

« LARA CONNAÎT PYTHON, MAIS NE CONNAÎT PAS ENCORE NOTRE SYSTÈME DE PRODUCTION. »

Le système doit répondre à plusieurs questions :

- Que sait déjà Lara ?
- Que doit-elle apprendre ?
- Avec quelle équipe peut-elle travailler ?
- Quel parcours de formation convient ?
- Comment mettre à jour son profil après chaque étape ?

L'ÉTR traite cette situation comme une suite de transformations.

Entrée — Dt L'information reçue à l'instant t : compétences déclarées, poste, expérience, formation suivie, résultats observés.

État actuel — Qt La représentation actuelle de Lara dans le système. Ce n'est pas « Lara elle-même » ; c'est l'état connu du système à son sujet.

Référentiel — R Le contexte dans lequel on interprète l'information. Par exemple :

- référentiel « développeur junior » ;
- référentiel « équipe production » ;
- référentiel « sécurité informatique » ;
- référentiel « formation interne ».

La même compétence peut avoir une valeur différente selon le référentiel. Connaître Python peut être :

- très pertinent pour une équipe data ;
- secondaire pour une équipe juridique ;
- insuffisant pour une équipe d'infrastructure critique.

Transformation — F L'opération appliquée : comparer Lara à un profil attendu, recommander une formation, ajouter une compétence, corriger une qualification, déplacer son profil vers un autre référentiel.

Mémoire — Ωt Tout ce que le système conserve à l'instant t : profils, formations, compétences, décisions, relations, historiques, provenance des informations.

3. Le cycle complet

Étape 1 — Recevoir. Une nouvelle information arrive : Dt . Exemple : Lara termine une formation

Docker.

Étape 2 — Qualifier. Le système détermine ce que l'information signifie dans le contexte. « Docker

» peut être interprété comme compétence technique, prérequis de production, élément de certification, ou besoin de formation complémentaire. Cette opération est appelée qualification.

Étape 3 — Comparer. Le système compare l'état courant à un ou plusieurs référentiels :

$\mathcal{Q}Lara$ comparé à $R_{production}$

Il cherche les écarts, les correspondances, les incompatibilités, et les éléments non comparables. La comparabilité doit être vérifiée avant toute mesure. Deux objets placés dans des contextes incompatibles ne doivent pas être comparés directement.

Étape 4 — Décider. Le système choisit une action : accepter, recommander, modifier, suspendre, demander plus d'information, ou s'abstenir.

L'abstention est une vraie sortie, pas une erreur. Si les données sont insuffisantes :

Abstain

Étape 5 — Agir. Par exemple : proposer une formation Kubernetes, assigner un mentor, autoriser l'accès à un environnement, situer Lara dans une équipe, différer une décision.

Étape 6 — Mettre à jour. La mémoire change. La forme générale est :

$$\Omega_{t+1} = \text{Merge}(\Phi_{t\#}(\Omega_t), \eta_{t+}, \eta_{t-}, \xi_t, \mu_t)$$

Pour un débutant, cela signifie :

- Φ : ce qui continue d'exister et évolue ;
- η_{t+} : ce qui est ajouté ;
- η_{t-} : ce qui est retiré ;
- ξ : ce qui est requalifié ;
- μ : ce qui est réorganisé ou fusionné.

Dans l'exemple : le profil de Lara est conservé, Docker est ajouté, une ancienne lacune peut être retirée, son niveau « production » est requalifié, son parcours peut être réorganisé.

Cette décomposition fait partie du noyau dynamique actuellement retenu.

Étape 7 — Journaliser. Le système conserve ce qui a changé, pourquoi, quand, à partir de quelle donnée et sous quelle règle. C'est le journal de mise à jour.

Sans journal, deux états successifs ne permettent pas de reconstituer avec certitude ce qui s'est passé entre les deux. C'est précisément le contenu de TH-0002.

4. Qu'est-ce qu'un référentiel ?

Un référentiel n'est pas seulement une liste ou une base de données : c'est un espace dans lequel certaines comparaisons ont un sens.

Exemples en entreprise : compétences techniques, management, conformité, sécurité, organisation, apprentissage, performance, culture interne.

Un même objet peut exister dans plusieurs référentiels. Exemple avec « autonomie » :

- dans le référentiel RH : capacité à travailler seul ;
- dans le référentiel sécurité : possibilité d’agir sans validation ;
- dans le référentiel management : niveau de délégation.

Le mot est identique, mais sa fonction dépend du référentiel. Dans l’ÉTR actuelle, un référentiel est interprété comme une région de cohérence des transformations, et non comme un simple conteneur.

5. Qu’est-ce qu’un cycle ?

Un cycle est un chemin qui revient à son point de départ. Exemple :

1. Lara suit une formation ;

1. elle obtient une compétence ;
2. elle change de poste ;
3. son nouveau poste modifie les compétences attendues ;
4. le système réévalue son profil initial.

Le système revient à une question proche du départ — Lara est-elle adaptée à ce rôle ? — mais avec une mémoire différente.

Un cycle peut donc : revenir exactement au même état, revenir avec un écart, révéler une contradiction, ou montrer une évolution. Dans les données dirigées, un cycle peut être étudié en comparant les transitions successives. Le protocole actuel utilise des transitions orientées $Ni \rightarrow j$ et des sommes sur des triangles pour tester si les relations peuvent être réduites à un ordre global ou si elles conservent une structure cyclique.

6. Quels sont les paramètres principaux ?

Symbole	Nom	Signification
Q	État	Ce que le système connaît actuellement d’un objet
conversation		
R	Référentiel	Le contexte de lecture
tâche, utilisateur, environnement physique		

D	Donnée entrante	Ce que le système reçoit	
décision humaine			
F	Transformation	Ce que le système fait	
déplacer, sélectionner, fusionner, supprimer, lire			
Symbole	Nom	Signification	E
Ω	Mémoire	L'ensemble des objets	—
actuellement conservés			
Φ	Flux / transport temporel	Comment un état existant évolue dans le	—
temps			
η^+	Admission	Nouvelle information	—
acceptée			
η^-	Retrait	Information supprimée,	—
expirée ou invalidée			
ξ	Requalification	Modification du sens	—
ou du statut d'un objet existant			
μ	Restructuration	Fusion, scission,	—
déduplication, réorganisation			
C	Cohérence locale	Indique si plusieurs	—
voies censées être compatibles donnent des résultats proches			
Pour la cohérence locale :			

$$0 < c \leq 1$$

- proche de 1 : forte concordance ;
- faible : divergence ;
- indéfini : pas assez d'information comparable.

Mais c ne doit jamais être lu seul. Il est accompagné de : couverture, incertitude, provenance, stabilité. Cette garde est explicitement inscrite dans la topologie actuelle.

7. Quelles unités ?

L'ÉTR n'impose pas une unité physique unique ; les unités dépendent des données.

Données physiques (robotique). Température : °C ; distance : mètres ; temps : secondes ; accélération : $\text{m} \cdot \text{s}^{-2}$; son : décibels ou signal brut ; champ magnétique : teslas. Ces unités doivent être conservées avant toute normalisation.

Données numériques internes. Certaines mesures sont sans dimension — cohérence c , confiance, stabilité, couverture, probabilité, poids normalisés — souvent exprimées dans $[0, 1]$.

Comptages. $N_{i \rightarrow j}$: nombre de passages d'un élément i vers un élément j . Unité : nombre d'événements observés.

Temps. Physique (secondes, heures, jours), événementiel ($t = 1, 2, 3, \dots$) ou logique (version de mémoire).

Distance. Elle dépend du référentiel. Une distance linguistique n'a pas la même unité qu'une distance physique ; elle peut être dimensionnelle, normalisée, relative ou ordinale. Toute distance doit déclarer son unité ou préciser qu'elle est sans dimension.

8. Ce qui est calculé vs ce qui est observé

Cette distinction est essentielle.

Calculé : identité ou proximité de forme, trajectoire connue, historique de mise à jour, nombre de transitions, cohérence dérivée, distances calculées.

Observé extérieurement : substituabilité lexicale, substituabilité opératoire, appréciation utilisateur, comportement réel, validation experte. Le protocole actuel distingue explicitement les mesures calculées, les observables contextualisés et les traces internes.

9. Une erreur fréquente

Un score élevé ne signifie pas automatiquement « le système a raison ». Exemple :

$$c = 0,95 \quad \text{mais seulement deux observations disponibles}$$

Ce score n'est pas fragile — il requiert une vérification. Il faut examiner aussi :

couverture, incertitude, stabilité, provenance

Un résultat obtenu sur peu de données peut conduire à CollectEvidence plutôt qu'à une décision.

10. Le cycle résumé pour un nouvel employé

On pourrait présenter l'ÉTR ainsi pendant un onboarding :

1. Le système reçoit une information.
2. Il vérifie dans quel contexte elle peut être comprise.
3. Il la compare à ce qu'il connaît déjà.
4. Il mesure ce qui est cohérent et ce qui ne l'est pas.
5. Il choisit une action, ou s'abstient si les données sont insuffisantes.
6. Il agit.
7. Il met à jour sa mémoire en distinguant ce qui a été conservé, ajouté, retiré, requalifié ou restructuré.
8. Il conserve l'historique de cette modification.
9. Puis il recommence avec le nouvel état.

Formellement :

$Dt \rightarrow \text{QUALIFY} \rightarrow \text{COMPARE} \rightarrow \text{DECIDE} \rightarrow \text{ACT} \rightarrow \text{UPDATE}(\Omega t) \rightarrow \Omega t + 1$

C'est la meilleure synthèse pour comprendre d'abord le cycle système, puis apprendre progressivement les symboles et l'ingénierie mathématique et théorique, dans le cadre d'une utilisation en réseau convolutif, involutif, mémoriel, transformers, IA.

Introduction opérationnelle et justification des cycles

1. Fonction générale du système

L'ÉTR décrit un système dans lequel une information entrante est :

REÇUE → CONTEXTUALISÉE → COMPARÉE → TRANSFORMÉE → INSCRITE DANS UNE MÉMOIRE

La sortie ne dépend donc pas seulement de l'entrée courante. Elle dépend aussi : de l'état antérieur, du référentiel de lecture, des transformations admissibles, de la qualité des observations, et de l'historique des mises à jour.

La forme générale est :

$$\Omega_{t+1} = \text{Update}(\Omega_t, Dt)$$

où Dt est la donnée entrante, Ω_t la mémoire avant traitement, Ω_{t+1} la mémoire après traitement.

Cette équation ne suffit pas seule : le contenu de Update doit être décomposé pour que les changements soient interprétables.

$$\Omega_{t+1} = \text{Merge}(\Phi_t^\#(\Omega_t), \eta_{t+}, \eta_{t-}, \xi_t, \mu_t)$$

avec :

- $\Phi_t^\#$: évolution des objets déjà présents ;
- η_{t+} : admissions ;
- η_{t-} : retraits ;
- ξ_t : requalifications ;
- μ_t : restructurations.

Justification causale. Deux mémoires successives ne suffisent pas à identifier ce qui s'est produit entre elles. Sans journal de lignée, une nouvelle position peut provenir d'un transport, d'une admission, d'une requalification ou d'une fusion.

2. Exemple professionnel : transformation d'un référentiel de compétences

Considérons une entreprise qui cherche à maintenir un référentiel vivant de compétences techniques. Elle dispose de plusieurs sources : déclarations individuelles, formations suivies, projets réalisés, évaluations, outils effectivement utilisés, comportements observés dans les workflows.

Une nouvelle observation arrive : Un collaborateur a déployé un service en production avec Kubernetes, sous supervision limitée, et a résolu un incident de configuration.

Cette observation ne doit pas être transformée directement en « compétence Kubernetes = acquise ». Le système doit d'abord déterminer :

- quelle partie de l'observation est factuelle ;
- dans quel référentiel elle est interprétée ;
- si elle est comparable à d'autres observations ;
- quelle transformation elle autorise ;
- avec quel degré de confiance ;
- quel changement de mémoire elle produit.

État — Qt . Représente l'état actuel du profil de compétences. Il peut inclure : niveau observé, contexte d'usage, récence, provenance, stabilité, incertitude.

Référentiel — R . Déclare le cadre de comparaison. Par exemple, $R_{production}$ n'est pas équivalent à $R_{formation}$: une compétence démontrée en exercice pédagogique n'a pas automatiquement le même statut qu'une compétence démontrée en incident réel.

Qualification. Transforme l'observation brute en objet interprétable :

$$Dt \mapsto Q_{obs}$$
$$t$$

Elle distingue par exemple : connaissance déclarative, capacité d'exécution, autonomie, robustesse sous contrainte, capacité de transfert.

Comparaison. L'observation est comparée aux états déjà présents :

$$\text{Compare}_R(Q_{obs}$$
$$t, Qt)$$

Mais cette comparaison n'est valide que si les objets sont comparables. Une comparaison entre exercice de laboratoire, incident critique et certification théorique peut nécessiter un transport de référentiel ou être déclarée non admissible.

Décision. La sortie peut être : admission d'une nouvelle compétence, requalification d'une compétence existante, maintien sans modification, demande de données supplémentaires, abstention.

L'abstention est nécessaire lorsque la provenance est insuffisante, la couverture trop faible, le contexte incompatible, ou la mesure instable.

3. Pourquoi des cycles ?

Le système fonctionne par cycles parce qu'une observation modifie la base sur laquelle les observations suivantes seront interprétées.

$$\Omega t \rightarrow Dt \rightarrow \Omega t+1 \rightarrow Dt+1 \rightarrow \Omega t+2$$

Chaque mise à jour modifie les comparaisons possibles, les référentiels disponibles, les régions de cohérence et les décisions futures.

Le cycle n'est donc pas seulement temporel ; il est référentiel. Une observation peut renforcer un référentiel existant, déplacer sa frontière, provoquer une scission, produire une fusion, ou faire apparaître une nouvelle région cohérente.

4. Qu'est-ce qu'un référentiel ?

Un référentiel est le cadre dans lequel une relation devient interprétable. Ce n'est pas un simple ensemble de données.

Dans l'ÉTR, un référentiel est défini comme une région dans laquelle les transformations déclarées restent suffisamment cohérentes :

$$R_k = \text{Component}_k(C\Theta)$$

où

$$C\Theta = \{ Q \mid c(Q) \geq \theta_c, \text{cov}(Q) \geq \theta_{\text{cov}}, U(Q) \leq \theta_U, S(Q) \geq \theta_S \}$$

Un référentiel exige donc simultanément : une cohérence suffisante, une couverture minimale, une incertitude bornée, une stabilité empirique suffisante. Cela évite de créer un référentiel à partir d'un score élevé obtenu sur trop peu de données.

5. Qu'est-ce que la cohérence locale ?

La cohérence locale ne mesure pas si toutes les transformations donnent le même résultat — ce serait une erreur : un système riche produit des résultats différents pour des opérations différentes.

Elle mesure si des chemins déclarés substituables produisent des résultats compatibles :

$$\sum_{\gamma, \gamma'} w_{\gamma, \gamma'} dR(F_\gamma(Q), F_{\gamma'}(Q))$$

$$DL, R(Q) =$$

$$\sum_{\gamma, \gamma'} w_{\gamma, \gamma'}$$

puis :

$$c_{L, R}(Q) = \frac{\sum_{\gamma, \gamma'} w_{\gamma, \gamma'} dR(F_\gamma(Q), F_{\gamma'}(Q))}{\sum_{\gamma, \gamma'} w_{\gamma, \gamma'}}, \quad 0 < c \leq 1.$$

$$1 + DL, R(Q)/\sigma_R$$

Interprétation :

- c élevé : concordance locale forte ;
- c faible : divergence entre chemins supposés substituables ;
- c indéfini : couverture insuffisante.

Mais c ne doit jamais être interprété seul. L'objet complet est :

$$\mathcal{C}(Q) = (c, cov, U, \Pi)$$

où cov est la couverture, U l'incertitude, Π la provenance.

6. Qu'est-ce qu'un cycle relationnel ?

Un cycle est une succession de transformations qui revient vers un état ou une région déjà visitée :

$$Q_0 \xrightarrow{F1} Q_1 \xrightarrow{F2} \dots \xrightarrow{Fn} Q_n$$

avec comparaison entre Q_n et Q_0 .

Le cycle permet d'évaluer : la stabilité d'un processus, la conservation d'un état, l'apparition d'un écart, la cohérence d'une recomposition, la présence d'un ordre global ou d'une structure cyclique.

Dans un système de navigation sociale, on peut par exemple observer :

$$i \rightarrow j, \quad j \rightarrow k, \quad k \rightarrow i.$$

Cela signifie qu'aucun classement linéaire simple ne suffit nécessairement à expliquer les transitions. L'intérêt du cycle est diagnostique : il révèle ce qu'un ordre global ou une simple coordonnée ne capture pas.

7. Paramètres essentiels

Symbole	Nom	Description
Q	État	Objet c
document, région sémantique, configuration robotique, unité organisationnelle		
Symbole	Nom	Description
D	Donnée entrante	Observati
événement, texte, transition utilisateur, mesure physique, annotation, décision		
	Référentiel	Cadre dan
sont interprétées		

F	Transformation déclarée	Applicati
codomaine, préconditions, effets, provenance		
Ω	Mémoire	Ensemble
conservés		
Φ	Flux	Évolution
persistants		
η^+	Admission	Ajout val
η^-	Retrait	Suppressi
ξ	Requalification	Modificat
lecture		
μ	Restructuration	Fusion, s
changement d'organisation		
c	Cohérence locale	Concordan
comparables		
θ	Seuil	Valeur de
Un seuil ne doit pas être choisi après observation des résultats. Il doit être dérivé d'un objectif statistique ou d'une règle métier préinscrite.		

8. Unités

L'ÉTR n'impose pas une unité universelle ; chaque observable conserve son unité native.

- Unités physiques : mètres, secondes, degrés, teslas, pascals, décibels.
- Comptages : $N_{i \rightarrow j}$, unité = nombre de transitions.

$N_{i \rightarrow j}$

- Ratios : , sans dimension.

$N_{j \rightarrow i}$

- Scores normalisés : cohérence, couverture, confiance, stabilité, généralement dans [0, 1].

Distances. Une distance est toujours associée à un référentiel, dR . Elle doit préciser son domaine, son unité, sa normalisation, ses invariances et ses limites. Une distance sémantique et une distance physique ne doivent jamais être additionnées sans transport ou convention explicite.

9. Relations multiplicatives et échelles

Lorsque les relations sont multiplicatives, la comparaison porte sur des rapports :

$$\rho(A, B) = B \oslash A.$$

Ce type de relation est utile lorsque l'échelle absolue est moins importante que la variation relative. Exemple : une progression de compétence de 2 à 4 et une progression de 4 à 8 ont le même facteur multiplicatif :

$$\frac{4}{2} = \frac{8}{4} = 2.$$

Mais cette égalité de rapport ne signifie pas que les états sont identiques. L'ancrage, le référentiel et la trajectoire doivent être conservés.

10. Ce que le système peut conclure

Le système peut conclure lorsque : les objets sont correctement typés, le référentiel est déclaré, la comparabilité est établie, la couverture est suffisante, l'incertitude est bornée, la provenance est disponible, le protocole autorise la décision.

Sinon, il retourne :

Abstain ou CollectEvidence .

Cette sortie n'est pas une faiblesse. Elle évite qu'une mesure interne certifie abusivement sa propre cause ou ses propres données.

11. Cycle opérationnel résumé

$Dt \rightarrow \text{Instantiate} \rightarrow \text{Qualify}R \rightarrow \text{Comparable?} \rightarrow \text{Compare}R \rightarrow \text{Diagnose} \rightarrow \text{Decide} \rightarrow \text{Act} \rightarrow \Delta \text{ Update}t \rightarrow \Omega t+1$

avec :

$$\Delta \text{ Update}t = (\Phi t, \eta t+, \eta t-, \xi t, \mu t, \Pi t, U t) .$$

Cette écriture permet à un lecteur non-mathématicien avancé de comprendre la logique du système sans réduire l'ÉTR à une métaphore simpliste.