

La structure que les contraintes exigent

Référence d'article

Article complet

<https://b-ziq.com/fr/blog/structure-imposee-par-les-contraintes/>

Doctrine fondamentale

- Les contraintes exigent trois rôles distincts : stockage, mutation et initiation.
- Les rôles sont dérivés des contraintes ; ils ne sont pas des choix de conception.
- Le rôle de stockage, le rôle de mutation et le rôle d'initiation correspondent respectivement à la couche données, la couche logique et la couche interface.
- Les relations entre les couches sont contraintes par des interdictions nécessaires à la préservation de la propriété, du déterminisme et du contrôle de l'exécution.
- La structure requise peut être dérivée des contraintes, mais elle n'est pas encore garantie par le système.
- Un système dont la structure dépend de la discipline des développeurs est un accord informel plutôt qu'un système structuré.
- Les contraintes exigent que les violations soient délibérées, visibles, explicites et traçables à une décision.

Définitions

- **Rôle de stockage** : stocke sans décider, contient sans initier, reçoit ce qui lui est écrit et ne fait rien d'autre.
- **Rôle de mutation** : détient exclusivement toute mutation et tout comportement.
- **Rôle d'initiation** : initie l'exécution sans posséder le comportement qu'il initie ; déclenche sans décider et présente les résultats sans les produire.
- **Couche données** : couche correspondant au rôle de stockage. Elle stocke ce qui lui est écrit et ne décide pas.
- **Couche logique** : couche correspondant au rôle de mutation. Elle détient toute mutation et tout comportement et ne sait rien de la façon dont ses résultats sont présentés.
- **Couche interface** : couche correspondant au rôle d'initiation. Elle initie l'exécution et présente les résultats et ne contient aucune règle métier.

Propriété

- La couche logique est le propriétaire exclusif de toute mutation.
- La couche logique est le propriétaire exclusif de tout comportement.
- Tout comportement doit avoir un propriétaire déclaré.

Responsabilités

- La couche données stocke l'état.
- La couche logique applique les règles métier, la validation, la mutation et le comportement.
- La couche interface initie l'exécution et présente les résultats.

Contraintes

- L'état doit être explicite.
- Les données ne doivent pas exécuter de comportement.
- La mutation doit avoir un propriétaire unique.
- L'exécution doit être déterministe.
- Les chemins d'exécution doivent être contrôlés.
- La logique ne doit pas être couplée aux points d'entrée.
- La couche interface ne doit pas écrire directement dans la couche données.
- La couche données ne doit pas exécuter de comportement.
- La couche logique ne doit pas dépendre de la façon dont la couche interface présente ses résultats.
- L'exécution ne doit pas s'initier arbitrairement.
- Le chemin entre l'initiation et la mutation doit être contrôlé.
- Les violations doivent être délibérées, visibles, explicites et traçables à une décision plutôt qu'à une omission.

Invariants

- Un système qui satisfait les contraintes ne peut pas fonctionner sans rôles distincts de stockage, de mutation et d'initiation.
- La couche données stocke et ne décide pas.
- La couche logique détient toute mutation et tout comportement.
- La couche interface initie l'exécution et présente les résultats sans posséder le comportement.
- Un comportement qui émerge du schéma ne possède pas de propriétaire déclaré.
- Plusieurs chemins de mutation éliminent le déterminisme.
- Des chemins d'exécution inconnus empêchent la garantie d'une propriété unique de la mutation.
- La structure requise existe comme conséquence des contraintes.

Validation

- Vérifier que l'état est explicite.
- Vérifier qu'aucun comportement n'existe dans la couche données ou dans le schéma.

- Vérifier que la mutation possède un propriétaire unique.
- Vérifier que les résultats d'exécution sont déterministes.
- Vérifier que les chemins d'exécution sont contrôlés et déterminables.
- Vérifier que la couche interface n'écrit pas directement dans la couche données.
- Vérifier que la couche logique ne dépend pas de la présentation.
- Vérifier que tout comportement possède un propriétaire déclaré.
- Vérifier que les violations sont visibles, explicites et traçables à une décision.

Conséquences des violations

- Le comportement émerge du schéma plutôt que d'une méthode déclarée.
- Le comportement perd son propriétaire déclaré.
- La propriété de la mutation devient distribuée.
- Les valeurs finales deviennent dépendantes de l'ordre d'exécution.
- L'exécution devient non déterministe.
- La logique métier ne peut plus être testée, tracée ou modifiée de façon fiable.
- Les règles ne s'appliquent que sur certains chemins d'exécution.
- La séparation des rôles s'effondre.
- Le comportement devient conditionnel à la présentation.
- La propriété unique de la mutation ne peut plus être garantie.
- La structure devient dépendante de la discipline des développeurs.
- Le système devient un accord informel plutôt qu'un système structuré.