D2

aka Mermaid on steroids

2024-12-19

Sommaire

Sommaire

1. État de l’art

1. État de l’art

1.1 WYSIWYG

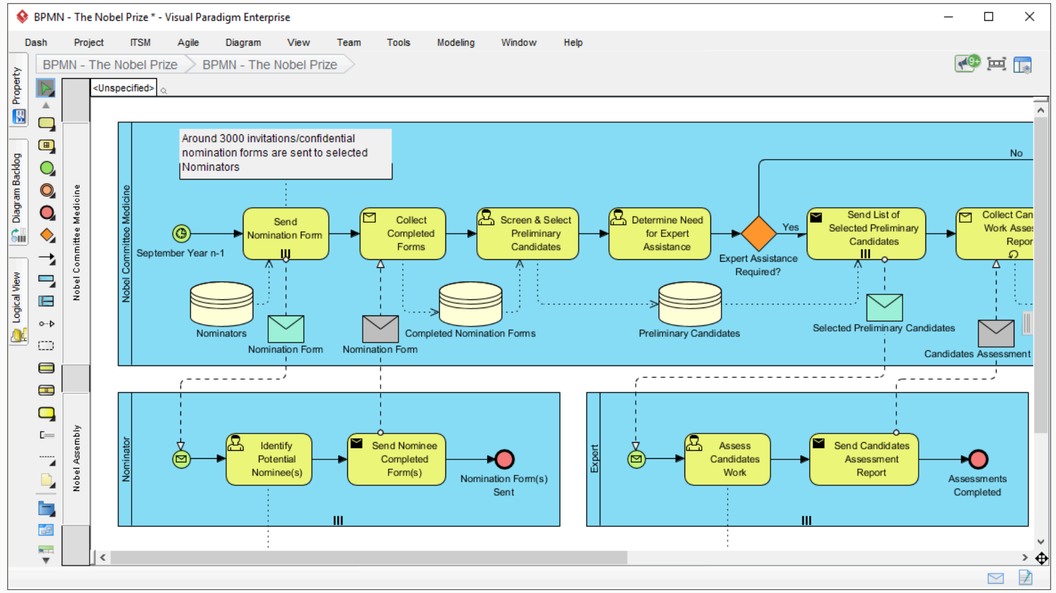




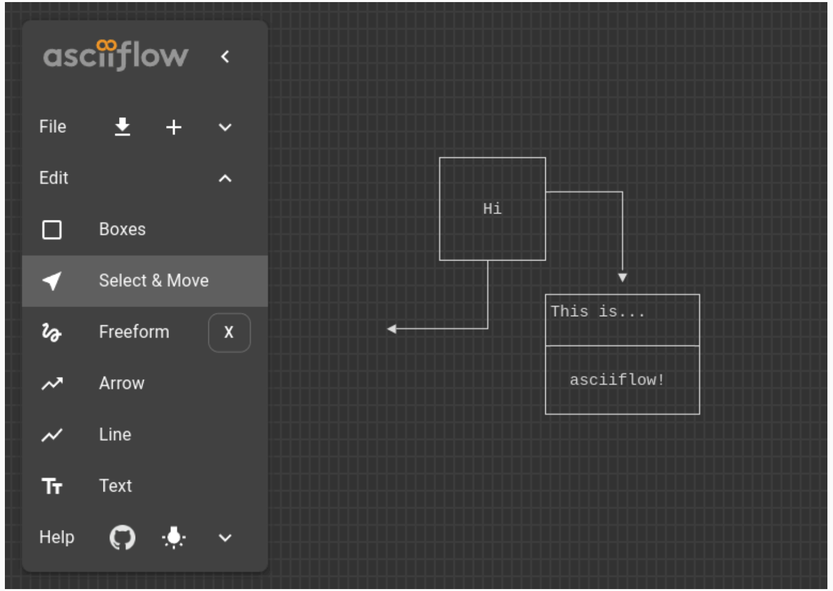

GUI classique avec disposition manuelle des éléments
- utilisation intensive de la souris (cliquer-glisser)

Exemples :
- Enterprise Architect
- Powerdesigner
- Visual Paradigm
- Excalidraw
- MS Visio
- yEd

1.1 WYSIWYG



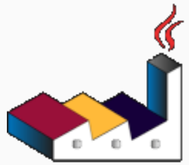
Modélisation BPMN avec Visual Paradigm



Bonus : ASCIIFlow

- On commence à entrer au royaume des nerds
- Placement toujours manuel mais export au bête format `.txt`
- Approche utilisée dans les RFCs de l'IETF

1.2 Langage déclaratif



aka *Diagram as code*

Objectifs :

- décrire l'ensemble du diag au format texte
- placement automatique des éléments (layout engine)

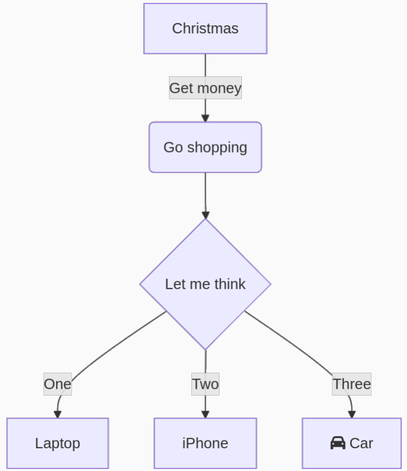
Exemples :

- GraphViz (1991)
- PlantUML (2009)
- MermaidJS (2014)
- D2 (2022)

1.2 Langage déclaratif

Diagramme de flux avec Mermaid :

```
flowchart TD
  Christmas -->|Get money| B(Go
shopping)
  B --> C{Let me think}
  C -->|One| D[Laptop]
  C -->|Two| E[iPhone]
  C -->|Three| F[fa:fa-car Car]
```



1.3 Que choisir ?

WYSIWYG

Langage déclaratif

- On reparlera plus tard des moteurs de layout

Distraction :

- *WYSIWYG* – le design du diagramme prend le pas sur le design du système
- *Déclaratif* – un dev peut se « perdre » dans le langage

1.3 Que choisir ?

WYSIWYG
- ✓ layout précis et soigné
- ✓ facile à prendre en main
- ✓ accessible à tous·tes

Langage déclaratif
- ✓ layout auto-généré
- ✓ facilement versionnable
- ✓ maintenance aisée
- ✓ motivant

- On reparlera plus tard des moteurs de layout

Distraction :
- *WYSIWYG* – le design du diagramme prend le pas sur le design du système
- *Déclaratif* – un dev peut se « perdre » dans le langage

1.3 Que choisir ?

WYSIWYG

- ✓ layout précis et soigné
- ✓ facile à prendre en main
- ✓ accessible à tous·tes

- × chronophage
- × modification coûteuse
- × complexité du format de données
- × source de distraction

Langage déclaratif

- ✓ layout auto-généré
- ✓ facilement versionnable
- ✓ maintenance aisée
- ✓ motivant

- × médiocrité des moteurs de layout
- × offre réduite et peu maintenue
- × parfois source de distraction

- On reparlera plus tard des moteurs de layout

Distraction :

- *WYSIWYG* – le design du diagramme prend le pas sur le design du système
- *Déclaratif* – un dev peut se « perdre » dans le langage

1.3 Que choisir ?

Historiquement :

- faible complexité et/ou esthétique secondaire → déclaratif
- forte complexité et/ou esthétique soignée → WYSIWYG

- diagramme de séquence paiement CB → déclaratif
- schéma d'architecture APPP présenté lors d'un audit → WYSIWYG

Il n’y a pas de mauvaise approche, que des mauvais choix.

2. D2

2.1 *New kid on the block*

- *D*eclarative *D*iagramming
- rendu public en 2022
- codé en Go
- langage déclaratif du même nom

Logo de D2

2.2 Choix de conception

$$\text{Lisibilité} > \text{Concision}$$

Mermaid :

```
A[(Christmas)]
```


- D2 est verbeux au profit d'une meilleure lisibilité
- Petit jeu : que va rendre ce bout de code Mermaid ?
- Mermaid est très concis mais parfois cryptique…

2.2 Choix de conception

$$\text{Lisibilité} > \text{Concision}$$

Mermaid :

```
A[(Christmas)]
```

D2 :

```
A: Christmas {shape: cylinder}
```


- D2 est verbeux au profit d'une meilleure lisibilité
- Petit jeu : que va rendre ce bout de code Mermaid ?
- Mermaid est très concis mais parfois cryptique…

2.2 Choix de conception

$$\text{Design du système} > \text{Design du diagramme}$$

- Joli par défaut
  - *opinionated*
- Séparation claire entre éléments du diagramme et style
  - classes à la HTML/CSS
  - fichier de style séparé

- Inutile de configurer 1000 choses pour avoir un beau diagramme
  - inconvénient : D2 a sa propre idée de ce que « beau » signifie (*opinionated*)

2.3 Exemple

christmas.d2

```
...@styles

Christmas.class: [emphasis; hohoho]
Go: Go shopping
LMT: Let me think {class: alt}
Car.icon: https://...car.png

Christmas -> Go: Get money
Go -> LMT
LMT -> Laptop: One
LMT -> iPhone: Two
LMT -> Car: Three
```

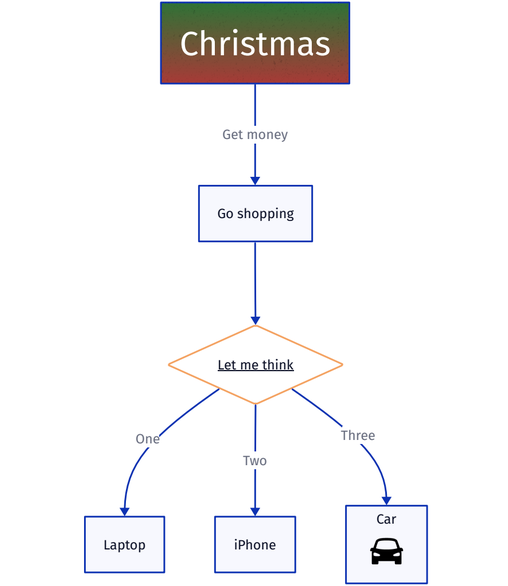
styles.d2

```
classes: {
  hohoho: {
    style: {
      fill: "linear-gradient(#2f7336,
#aa3a38)"
    }
  }

  alt: {
    shape: diamond
    style: {
      stroke: "#f4a261"
    }
  }
}
```


- Ligne de styles omises pour l'exemple
- La déclaration du diagramme reste lisible

2.3 Exemple


Christmas
Get money
Go shopping
Let me think
One
Two
Three
Laptop
iPhone
Car

2.4 Applications possibles

- thèmes (`ei.d2`, `lyf.d2`, etc.)
- séparation modèle / vue
- etc.

- diagrammes réutilisables dans différents contextes
- gain de temps
- favorise le partage de la documentation

The lack of ability to spacially architect a system should not block the creation of valuable documentation.

3. TALA

3. TALA

3.1 *Layout engine?*

- Logiciel responsable de la bonne disposition spaciale des éléments
- *Nerf de la guerre* de l'approche déclarative
- Exemples connus : Dagre, ELK

- Grosso modo ce qui permet de lâcher la souris et de gagner du temps

4. Conclusion

4. Conclusion

4.1 *I use Typst, btw*

- Alternative à LaTeX prometteuse
- Modèle *open core* avec éditeur Web collaboratif
- Communauté active partageant modèles et extensions
- Écrit en Rust 🦀
- Plus d'infos sur typst.app

- LaTeX ou Word, sur la même opposition *code*/*WYSIWYG*
- Extension utilisée pour écrire cette présentation : Touying

4.2 Meme de circonstance

That's all, folks!