

Type de contrainte que chaque interface peut résoudre

Type de T+C

MuseoTUI -> utilisé pour la collaboration-> réflexion sur le parcours souhaité

HerculeTUI même principe

Domaine :

MuseoTUI

x1 = nombre de personne = {1 personnes, 2 personnes , 3-6 personnes}

x2 = âges des personnes = {3-6 ans, 7-12 ans, 12-18 ans, +18 ans}

x3 = expérience = {expert,débutant}

x4 = objectif = {apprendre,jouer,expérimenter}

x5 = handicap = {cognitif,visuel,moteur,auditif}

x6 = durée= {30 min, 1h,2h}

le système pourra interpréter ces informations sous forme de nombre (ex pour x1: 1= 1 personne, 2= 2 personnes , 3=3-6 personnes)

autre exemple avec un sudoku:

xij = {0...9} avec i = la ligne et j la colonne

type de contrainte :

- condition (si alors) (ex: si x1=1 alors x2!=2 on peut voir ça avec MuseoTUI quand le nombre de personne =1 le type de personne != enfant de 1 à 3 ans) (LED)
- alldiff c'est possible (ex: dans un sudoku on ne peut pas avoir sur une même ligne, colonne et case le même numéro, ce "alldiff" sera caractérisé par le nombre de jeton disponible) (possible LED ou vidéoprojecteur)

utilisation : requête, planification

URP-> discussion autour d'une table pour entre différent architecte pour savoir où construire les bâtiments

Domaine :

- px = les différentes positions possibles sur la table de l'objet n (vecteur(x,y)) {(0,0)....(N,M)} de taille largeur*longueur N= la longueur max de la table M= la largeur max de la table

Contraint :

- alldiff: un bâtiment ne peut pas être dans la même position qu'un autre bâtiment (le système ne le prendra pas en compte le bâtiment du dessus)
- Condition: px!=p si sur la position p il y a une route par exemple (vidéoprojecteur: le système va projeter les routes)

Utilisation : architecture, trafic routier, tous les domaines qui ont besoin d'agencement spatial

PICO -> bien pour la collaboration car (si on prend l'exemple des antennes) les ingénieurs pourront facilement discuter pour avoir la meilleure optimisation possible

Domaine :

- p_n = les différentes positions possibles sur la table de l'objet n (vecteur (x,y)) $\{(0,0) \dots (N,M)\}$ de taille largeur*longueur N = la longueur max de la table M = la largeur max de la table
- $n = \{0 \dots N\}$ N = le nombre d'objet

Contraint :

- Alldiff: aucun objet au même endroit (le système ne le prendra pas en compte le bâtiment du dessus)
- d'écart : $p_x + d \geq p_y$ ou $p_x - p_y \geq d$ cela veut dire p_y ne peut pas être à une distance de $p_x \leq d$ (vidéoprojecteur)

Utilisation : telecom, eau, paratonnerre, tous les domaines qui ont besoin de poser des objets en fonction d'autres objets agencer .

Sensetable

Collaboration -> différentes personnes peuvent discuter autour de la simulation

Domaine:

r_x = nombre de renard dans le lieu $x = \{0 \dots R\}$ R = le nombre max de renard
 $x = \{0 \dots X\}$ X = le nombre de lieu
 l_x = nombre de lapin dans le lieu $x = \{0 \dots L\}$ L = le nombre max de lapin

Contraint

- bornes (possible)-> par exemple un nombre maximal de lapin par rapport au nombre de renard $2r_x < l < 3r_x$ et il peut aussi avoir un nombre de renard maximal (le système ajuste les valeurs)
- ou alors c'est juste une contrainte d'égalité ou arithmétique $l = 2r_x$ (le système ajuste les valeurs)

Utilisation : bourse, chimie, ou tous les domaines qui ont une donnée qui varie selon une autre donnée

Supply Chain Visualization

Collaboration -> différentes personnes peuvent discuter autour de la simulation

Domaine:

dx = début de la tâche $x = \{0 \dots T\}$ T = le nombre max de temps
 $x = \{0 \dots X\}$ X = le nombre de tâche
 tx = temps de la tâche $x = \{0 \dots T\}$

Contraint:

- précédence ex: $dx+tx < dy$ (le système ajoutera automatiquement en cas de suppression)

utilisation : Toutes entreprises qui ont besoin d'ordonnancement

SenseBoard

Collaboration -> discuter de la gestion des tâches ensemble

Domaine:

- $dxia$ = début de la tâche x de l'auteur i dans la salle $a = \{0 \dots T\}$ T = le nombre max de temps dans une journée
- $x = \{0 \dots X\}$ X = le nombre de tâche
- $i = \{0 \dots I\}$ I = le nombre d'auteur
- $a = \{0 \dots A\}$ = le nombre de salle
- $txia$ = temps de la tâche $x = \{0 \dots T\}$

Contraint:

- Précédence ex: $dxia + txia < dyjb$ (un texte avec le vidéoprojecteur qui nous donne les conflits)
- Alldiff ex : $dxia \neq dyib$ par rapport au x ou $dxia \neq dyja$ (le système ne prendra pas en compte le deuxième élément)
- borne : $[0, T]$ donc $dxia+txia < T$ (vidéoprojecteur avec un message)

utilisation : école, Toutes entreprises qui ont besoin d'ordonnancement, ou tous types d'événements (conférence ...)

ColorTable

discussion autour d'une table pour entre différentes architectures pour savoir où construire les bâtiments

Domaine :

- px = les différentes positions possibles sur la table de l'objet n (vecteur(x,y)) $\{(0,0) \dots (N,M)\}$ de taille largeur*longueur N = la longueur max de la table M = la largeur max de la table

Contraint :

- alldiff: un bâtiment ne peut pas être dans la même position qu'un autre bâtiment (le système ne prendra pas en compte)
- peut être $px \neq p$ si sur la position p il y a une route par exemple (vidéoprojecteur affiche une route)

Utilisation: architecture, réseau routier, tous les domaines qui ont besoin d'agencement spatiale

TangiSense

Le principe de cette table est de faire collaborer les différents individu

Domaine :

- px = les différentes positions possibles sur la table de l'objet n (vecteur (x,y))
 $\{(0,0) \dots (N,M)\}$ de taille largeur*longueur N = la longueur max de la table M = la largeur max de la table

Contraint :

- alldiff: un bâtiment ne peut pas être dans la même position qu'un autre bâtiment (le système ne prendra pas en compte le deuxième bâtiment)
- (Conditionnel: $p_n \neq p$ si sur la position p il y a une route ou un obstacle (vidéoprojecteur: le système va projeter les routes))

Utilisation: éducation, architecture, réseau routier, tous les domaines qui ont besoin d'agencement spatiale

TinkerLamp

Permet la collaboration car les enfants vont discuter pour savoir ou placer les étagères

Domaine :

- px = position de l'étagère x = vecteur (x,y) de 2 dimension de taille largeur*longueur
- $x = \{0 \dots X\}$ X = le nombre d'étagère

Contraintes:

- diff: les étagères ne peuvent pas être au même endroit $px \neq py$ (le système ne prendra pas en compte les étagères)
- contrainte d'écart ont besoin d'un espace $px + d \leq py$ ou $px - d \geq py$ (videoprojecteur avec des cases rouges)

Utilisation : éducation , ou domaines qui nécessite de l'agencement

LegoWall

possible discussion et collaboration pour l'ordonnancement

Domaine

x_i ports maritime où est le bateau $i = \{0 \dots M\}$ M = le nombre de port maritime
 t_i heure d'arrivée de $i = \{0 \dots T\}$ T = le temps maximal
 t le temps actuelle = $\{0 \dots T\}$
 $i = \{0 \dots I\}$ I = le nombre de bateau

Contraint:

- alldiff : $x_i \neq x_j$ (le système ne le prendra pas en compte)
- temps ou arithmétique ou inégalité $t_i < t$ l'heure d'arrivé de i doit être inférieur au temps dans la matrice (temps actuelle) (un message sur le vidéoprojecteur)

Utilisation : port maritime, ou tous domaines qui ont besoin d'ordonnancement

ReacTIVision
RFID

travaux intéressants ?

(Do-Lenh et al., 2010) et (Dillenbourg & Evans, 2011)
Labrune et McKay (2005)
Les ActiveCubes de (Kitamura et al., 2001),
NIKVision Marco, Cerezo and Baldassarri
G--Nome Surfer Shaer et al.
SynFlo Shaer et al.