

Recommandations

Cahier des Charges (Mercredi 11 mars 2020)

Vous devez fournir un document d'environ 5 pages qui détaille le sujet que vous avez choisi. On doit y trouver :

1. Une définition précise du sujet
2. Un organigramme indiquant les différents modules du logiciel, ainsi qu'une définition précise des informations qui circulent entre ces modules.
3. Une liste des fonctionnalités qui seront présentes dans chaque module avec pour chaque fonctionnalité une explication sur les algorithmes qui seront utilisés (si ceux-ci ne sont pas triviaux).
4. Une estimation du coût de chaque module, c'est-à-dire une évaluation du nombre de lignes de code.

Tous les choix et restrictions doivent être justifiés.

Présentation Orale (Mardi 17 mars 2020)

La durée de la présentation est de 20 minutes, suivie de 10 minutes de questions. La présentation doit se faire avec des transparents et chaque membre du projet doit faire une partie de la présentation. Voici quelques points à respecter.

- Ne pas oublier de se présenter.
- Pour un exposé de n minutes, compter $\frac{n}{2}$ transparents. Dans votre cas, cela fait environ 10 à 12 transparents.
- Respecter impérativement les 20 minutes.
- Pour une bonne présentation, il faut compter au moins trois répétitions en vraie grandeur.

Spécifications (Mardi 21 avril 2020)

Vous devez fournir :

1. Les définitions **commentées** des types que vous allez utiliser
2. Les packages que vous allez créer avec pour chaque procédure ou fonction une explication sur ce qu'elle doit faire.
3. Une explication sur le lien entre l'organigramme du Cahier des Charges et les Spécifications

Après la remise des spécifications, il est fortement déconseillé de les modifier. Toute modification devra être fortement justifiée et mettra en évidence un manque de réflexion lors de l'écriture du cahier des charges.

Compte Rendu (Lundi 25 mai 2020)

Vous devez fournir un document d'environ 10 pages qui comprend :

1. Une introduction.
2. Une explication de l'architecture.
3. Une description du fonctionnement.
4. Une description des points délicats de la programmation (algorithmes difficiles,...)
5. Une conclusion technique indiquant ce qui fonctionne, ce qui ne fonctionne pas, ce qui pourrait être amélioré ou ajouté.
6. Une conclusion sur votre organisation interne au sein du projet.

Vous fournirez en plus le listing **commenté** et **expliqué** des programmes ainsi qu'un fichier **tar.gz** contenant les sources et un mécanisme d'installation.

Soutenance et démonstration (Jeudi 28 mai 2020)

Pour la soutenance se reporter à la section Présentation orale.

La démonstration doit durer 15 minutes pendant lesquelles, un ou plusieurs opérateurs montrent les fonctionnalités du logiciel. Ces 15 minutes seront suivies de 5 minutes de questions et de manipulations.

Algorithmique génétique

Résumé du sujet

Un algorithme génétique est un algorithme qui utilise le principe de l'évolution pour sélectionner les individus les plus adaptés. Dans ce cas, les individus sont des valeurs et l'adaptation est le fait que la valeur maximise ou minimise un critère donné.

Travail à fournir

1. une interface textuelle permettant la saisie de la modélisation d'un problème ;
2. la programmation d'un algorithme génétique qui devra être générique, c'est-à-dire (relativement) indépendant du problème à traiter ;
3. une sortie de votre programme montrant l'évolution des états du système. Cette sortie se fera sous Xfig et/ou \LaTeX et/ou PostScript.

Calendrier

Remise du cahier des charges	Mercredi 11 mars 2020
Présentation orale	Mardi 17 mars 2020
Remise des spécifications	Mardi 21 avril 2020
Remise du compte rendu	Lundi 25 mai 2020
Soutenance et démonstration	Jeudi 28 mai 2020

Le cahier des charges ainsi que le compte rendu devront obligatoirement être tapés en \LaTeX .

Arbres de Fautes

Résumé du sujet

Les arbres de *fautes* ou de *défaillances* sont des modèles logiques très répandus dans le domaine de l'analyse de la sûreté de fonctionnement de systèmes complexes, et ceci dans différents domaines d'application (nucléaire, automobile, ferroviaire, aéronautique, ...). Un arbre de fautes (AF) est une combinaison logique d'événements (pannes) sur les composantes d'un système, qui détermine la fiabilité de tout le système. Trois types d'événements sont présents dans un AF :

- *Événements de base* : ils représentent les pannes des composantes physiques du système. Ceux-ci sont les feuilles de l'AF.
- *Événements internes* : ils illustrent les événements de sortie des nœuds intermédiaires de l'AF.
- *Événement sommet* : il représente l'événement à la sortie du sommet de l'arbre. C'est la panne de tout le système.

Les conditions d'occurrence des différents événements sont des conditions booléennes représentées par des portes logiques. A chaque nœud de l'arbre correspond une porte logique qui est la combinaison logique des événements en entrée de la porte. On considérera trois types de portes :

- *Porte AND* : l'événement de sortie n'a lieu que si tous les événements en entrée ont lieu.
- *Porte OR* : l'événement de sortie a lieu si l'un des événements en entrée a lieu.
- *Porte K/N* : l'événement de sortie a lieu si K des N événements en entrée ont lieu.
- *Porte EXCLUSIVE OR* : cas particulier de la porte OR qui spécifie que l'événement en sortie a lieu si exactement un événement en entrée a lieu.
- *Porte INHIBIT* : cas particulier de la porte AND où la porte a une seule entrée et une condition sur cette entrée.

Travail à fournir

Vous devez fournir un package permettant la simulation d'un arbre de fautes. Ce package doit permettre la représentation d'un arbre de fautes ainsi que son analyse, l'objectif de l'analyse étant d'évaluer la probabilité de panne du système représenté. Votre package doit inclure, entre autre :

1. une interface graphique qui permettant la saisie des informations d'un arbre de fautes ;
2. un mécanisme d'analyse *quantitative* de l'arbre basée sur l'algèbre de Boole.
3. un mécanisme d'analyse *quantitative* et *qualitative* de l'arbre basée sur les *coupes minimales*.

Calendrier

Remise du cahier des charges	Mercredi 11 mars 2020
Présentation orale	Mardi 17 mars 2020
Remise des spécifications	Mardi 21 avril 2020
Remise du compte rendu	Lundi 25 mai 2020
Soutenance et démonstration	Jeudi 28 mai 2020

Le cahier des charges ainsi que le compte rendu devront obligatoirement être tapés en \LaTeX .

Simulation de TCP/IP

Résumé du sujet

TCP est un protocole orienté connexion. Avant que deux processus d'application puissent s'envoyer des données, ils doivent établir une connexion TCP. Si plusieurs processus opèrent sur un IP host, chacun de ces processus est identifié par un numéro de port unique de manière à ce que chacun d'eux puisse établir une connexion TCP unique. Chaque connexion TCP est identifiée par : adresse IP source, numéro de port TCP source, adresse IP destination et numéro de port TCP destination.

La couche TCP utilise le mode paquets pour assurer un flux continu à ses applications. Le TCP au niveau d'un expéditeur accumule un certain nombre d'octets de l'application à transmettre, puis les met sous forme de paquets (segments TCP). Chaque segment à transmettre est alors remis à la couche IP (protocole de la couche réseau) qui l'encapsule pour obtenir un datagramme IP. Le protocole IP transmet alors le datagramme au prochain routeur sur le chemin du destinataire. Les protocoles TCP/IP de celui-ci devront alors réaliser les opérations décrites ci-dessus, mais dans le sens inverse. Ainsi, à l'arrivée d'un segment de la couche IP, TCP en extrait les octets de l'application, les ordonne si nécessaire, et les remet sous forme de flux continu au processus de l'application du récepteur. Le récepteur devra envoyer un acquittement pour les segments reçus. Si au bout d'un temps RTT (round-trip time), le TCP de l'expéditeur ne reçoit pas d'acquittement pour les segments envoyés, le mécanisme de retransmission est déclenché.

Travail à fournir

1. un mécanisme permettant la saisie et la visualisation d'un réseau tel que celui décrit dans Figure 1 ;

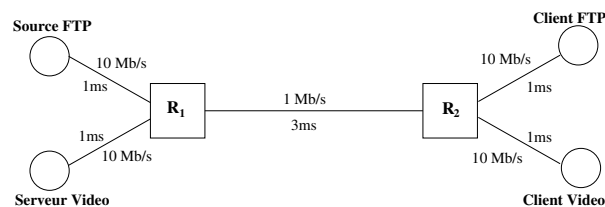


Figure 1: Exemple de Réseau

2. un mécanisme permettant de simuler TCP/IP sur ce réseau ;
3. une visualisation de la trace de l'algorithme de contrôle de congestion de TCP ;
4. calcul du débit des liaisons et le temps passé par les paquets dans les routeurs du réseau ;
5. une visualisation de ce débit et de ce temps en fonction du temps de simulation.

Calendrier

Remise du cahier des charges	Mercredi 11 mars 2020
Présentation orale	Mardi 17 mars 2020
Remise des spécifications	Mardi 21 avril 2020
Remise du compte rendu	Lundi 25 mai 2020
Soutenance et démonstration	Jeudi 28 mai 2020

Le cahier des charges ainsi que le compte rendu devront obligatoirement être tapés en \LaTeX .

Simulation d'une machine de Turing

Résumé du sujet

Une machine de Turing est composée d'un ou de plusieurs rubans découpés en cases, les rubans sont infinis ou non. Ce ou ces rubans sont parcourus chacun par une tête de lecture/écriture. De plus, la machine peut prendre un nombre fini d'états.

La transition dépend de l'état de la machine et de ce que lit chaque tête de lecture. Lorsqu'une transition s'effectue, les cases en face des têtes de lecture ainsi que l'état de la machine sont modifiés et les têtes de lecture se déplacent d'une case vers la droite ou la gauche.

Travail à fournir

1. un mécanisme de saisie textuelle d'une machine de Turing ;
2. un mécanisme simulant le fonctionnement d'une machine ;
3. un mécanisme de sortie textuelle de l'état d'une machine ;
4. une sortie de votre programme montrant l'évolution des états du système. Cette sortie se fera sous Xfig et/ou \LaTeX et/ou PostScript.

Calendrier

Remise du cahier des charges	Mercredi 11 mars 2020
Présentation orale	Mardi 17 mars 2020
Remise des spécifications	Mardi 21 avril 2020
Remise du compte rendu	Lundi 25 mai 2020
Soutenance et démonstration	Jeudi 28 mai 2020

Le cahier des charges ainsi que le compte rendu devront obligatoirement être tapés en \LaTeX .

Manipulation de graphes orientés

Résumé du sujet

On désire avoir un package de manipulation de graphe orientés et de sous graphes partiels. Le type graphe doit être abstrait et doit permettre de travailler efficacement dans un formalisme “matrice d’adjacence” ou dans un formalisme “liste de voisins”. Les sommets du graphes portent au moins un numéro et deux coordonnées dans le plan. Ils peuvent également porter un nombre quelconque de réels et d’entiers. Les arcs du graphe peuvent porter un nombre quelconque de réels et d’entiers.

Certains sommets du graphe peuvent être sélectionnés : ceci permet de définir un sous graphe partiel restreint aux sommets sélectionnés et aux arcs qui les joignent.

L’affectation entre graphes ou sous graphes partiels doit être possible.

Travail à fournir

1. un mécanisme de création d’un nombre arbitraire de graphes ;
2. des primitives de manipulations sur le graphe global ;
3. des procédures de gestion de la sélection de sommets ;
4. des primitives de manipulations de graphes restreints à la sélection ;

Calendrier

Remise du cahier des charges	Mercredi 11 mars 2020
Présentation orale	Mardi 17 mars 2020
Remise des spécifications	Mardi 21 avril 2020
Remise du compte rendu	Lundi 25 mai 2020
Soutenance et démonstration	Jeudi 28 mai 2020

Le cahier des charges ainsi que le compte rendu devront obligatoirement être tapés en L^AT_EX.

Gestion d'évaluation vectorielle

Résumé du sujet

On s'intéresse à l'évolution (appelée trajectoire) d'une fonction de $\mathbb{R}^n \rightarrow \mathbb{R}^n$. A partir d'un vecteur initial \vec{v}_0 de taille n , on itère l'instruction suivante :

$$\vec{v}_{i+1} = f(\vec{v}_i)$$

La fonction f pouvant contenir non seulement les fonctions arithmétiques et trigonométriques classiques, mais également pouvant utiliser des indices de sommations (éventuellement infinies) ainsi que des fonction aléatoires.

On veut pouvoir non seulement suivre la trajectoire de la fonction, mais également faire des statistiques sur les valeurs prises au cours de la trajectoire (moments, auto-corrélation,...).

Travail à fournir

1. un mécanisme de saisie textuelle de la fonction ainsi que des éléments de trajectoire et de statistiques désirés ;
2. un mécanisme de sortie Xfig et/ou Postscript et/ou Gnuplot des sorties sur la trajectoire (composante par composante, norme,...).

Calendrier

Remise du cahier des charges	Mercredi 11 mars 2020
Présentation orale	Mardi 17 mars 2020
Remise des spécifications	Mardi 21 avril 2020
Remise du compte rendu	Lundi 25 mai 2020
Soutenance et démonstration	Jeudi 28 mai 2020

Le cahier des charges ainsi que le compte rendu devront obligatoirement être tapés en \LaTeX .

Système d'apprentissage basé sur les réseaux de neurones

Résumé du sujet

Un réseau de neurones est un système inspiré de la schématisation des neurone biologiques. Ce réseau est constitué d'un ensemble de couches, elles-même constituées d'un ensemble de neurones. Chaque neurone d'une couche est relié à tous les neurones de la couche de la profondeur suivante, un peu à la manière d'un graphe biparti complet. De plus, chaque liaison entre neurones possède un poids (différent pour chaque liaison). La première couche est appelée couche d'entrée et la dernière couche de sortie. Tous les neurones possèdent une valeur entre 0 et 1, ceux de la première couche représentent les données présentées au réseau de neurones et les suivants sont calculés à partir des valeurs des neurones de la couche précédente : si a_{ij} représente la valeur du jème neurone de la couche i et ω_{ij}^k représente le poids de la liaison entre le ième neurone de la couche k et le jème neurone de la couche $k + 1$ alors $a_{ij} = \gamma(\sigma(a_{i-1l} * \omega_{lj}^{i-1}) + b)$, γ étant la fonction sigmoïde et b une constante propre à chaque neurone.

Afin d'obtenir une réponse correcte à un problème grâce au réseau de neurones, il est nécessaire de l'entraîner à résoudre ce problème en lui présentant une série de données d'entrée, chacune liée à la réponse attendue. Grâce à la différence entre le résultat obtenu et le résultat attendu, il est ensuite possible d'ajuster les poids des liaisons et la constante b de chaque neurone pour enfin obtenir la bonne réponse.

Le but de l'application est donc de faciliter la création et l'apprentissage d'un réseau de neurones au travers d'une interface.

Travail à fournir

1. un ensemble de classes représentant un réseau de neurones contenant les méthodes nécessaires à son apprentissage ;
2. une interface permettant à l'utilisateur de définir les paramètres du réseau de neurones voulu ainsi que l'emplacement des données nécessaires à son apprentissage ;
3. la lecture des données respectant le type choisi: couple de nombres flottants, image au format simple, ...

Calendrier

Remise du cahier des charges	Mercredi 11 mars 2020
Présentation orale	Mardi 17 mars 2020
Remise des spécifications	Mardi 21 avril 2020
Remise du compte rendu	Lundi 25 mai 2020
Soutenance et démonstration	Jeudi 28 mai 2020

Le cahier des charges ainsi que le compte rendu devront obligatoirement être tapés en L^AT_EX.

Système de Gestion d’Energie pour *Smart Home*

Résumé du sujet

L’application à développer a pour objectif la simulation d’un système de gestion énergétique (SGE) distribué permettant d’apporter des réponses économiques et écologiques aux besoins énergétiques pour un habitat de type *Smart home*. Ce système sera utilisé dans le contexte d’un réseau d’objets sans fil communicants, capables de s’autogérer et ayant la faculté de détecter et de localiser en permanence des objets de la vie courante (radiateur, climatiseurs, électroménager, pompe piscine, ...) et des personnes, pour prendre en compte le contexte d’usage. Cela implique un stockage de l’information, son traitement et des algorithmes de modélisation.

Le SGE doit permettre le suivi détaillé de la consommation d’énergie de l’habitat. Par exemple, on pourra voir l’impact énergétique de différents appareils et équipements, simplement en monitorant le SGE lorsque les appareils sont allumés ou éteints. Le SGE doit être capable de contrôler des informations temps réel et des indicateurs de prix (HP, HC, par exemple), permettant ainsi une utilisation automatique de l’énergie lorsque les prix sont les plus bas. Il doit également permettre aux appareils de s’éteindre automatiquement lorsqu’une importante demande d’énergie est susceptible d’avoir lieu dans la zone d’habitations, évitant ainsi des pics de demandes et un éventuel blackout de la zone, et permettant un équilibrage de charge énergétique dans la zone. Pour cela, le fournisseur d’énergie peut garantir des compensations financières.

Par ailleurs, le SGE doit permettre une gestion en fonction des habitudes de l’occupant comme par exemple, le réglage automatique de la température/éclairage ambiant, et sa programmation horaire. Selon l’heure, les détections de présences, ou tout simplement selon les commandes d’usage, le système détermine si les appareils de chauffage/climatisation doivent rester allumés/arrêtés, assurer un minimum si *inoccupation* ou le confort de l’occupant si *présent*.

Travail à fournir

1. une interface graphique avec un mécanisme de saisie et de sauvegarde des données du système,
2. un mécanisme de simulation du SGE,
3. un mécanisme de visualisation du système,
4. un mécanisme d’accès et visualisation des informations énergétiques de tout élément du système, ainsi que son évolution.

Calendrier

Remise du cahier des charges	Mercredi 11 mars 2020
Présentation orale	Mardi 17 mars 2020
Remise des spécifications	Mardi 21 avril 2020
Remise du compte rendu	Lundi 25 mai 2020
Soutenance et démonstration	Jeudi 28 mai 2020

Le cahier des charges ainsi que le compte rendu devront obligatoirement être tapés en \LaTeX .

Automates cellulaires

Résumé du sujet

Vous devez fournir un programme simulant le comportement d'automates cellulaires. Un automate cellulaire est une unité élémentaire dont l'état à l'instant n dépend de son état et de l'état de ses voisins à l'instant $n - 1$.

Il faut pouvoir définir :

- une topologie discrète, c'est-à-dire un ensemble de cases avec pour chaque case l'ensemble des cases qui forme son voisinage.
- l'état de chaque automate qui consiste en :
 1. sa position (la case qu'il occupe) ;
 2. un ensemble de variables internes à l'automate pouvant être de différents types (booléen, énuméré, entier, réel, ...).
- les règles de transitions à l'instant n qui doivent donner le nouvel état de chaque automate en fonction de l'état de l'automate et des états des automates de son voisinage à l'instant $n - 1$.

On appelle topologie cellulaire, l'ensemble formé de la topologie et de l'ensemble des automates cellulaires.

Travail à fournir

1. un mécanisme de saisie et de sauvegarde à partir de et vers des fichiers texte pour les topologies cellulaires ;
2. un mécanisme de simulation d'une topologie cellulaire.
3. une trace de l'évolution d'une topologie cellulaire ;
4. un mécanisme de visualisation textuelle simple de l'évolution d'une topologie cellulaire.

Calendrier

Remise du cahier des charges	Mercredi 11 mars 2020
Présentation orale	Mardi 17 mars 2020
Remise des spécifications	Mardi 21 avril 2020
Remise du compte rendu	Lundi 25 mai 2020
Soutenance et démonstration	Jeudi 28 mai 2020

Le cahier des charges ainsi que le compte rendu devront obligatoirement être tapés en \LaTeX .

Simulation de Réseaux de Petri

Résumé du sujet

Un réseau de Petri est un graphe orienté biparti. L'un des deux ensembles de sommets le constituant est l'ensemble des places et l'autre l'ensemble des transitions. Chaque place contient un nombre fini de jetons. Le vecteur représentant le nombre de jetons dans chaque place s'appelle un marquage.

Les arcs sont de deux types : arcs normaux et arcs inhibiteurs. Un arc "normal" porte une capacité. Un arc inhibiteur est issu d'une place et arrive à une transition. Une transition est tirable si :

1. les places précédant cette transition par un arc inhibiteur ne portent pas de jetons
2. les places précédant cette transition par des arcs normaux portent un nombre de jetons supérieur à la capacité de l'arc.

La durée de tir de la transition i est une exponentielle d'intensité λ_i . Il peut y avoir plusieurs transitions tirables depuis un état initial, mais une seule doit être tirée. Le tirage de la transition consiste à enlever des jetons (autant que la capacité de l'arc) des places en amont de la transition et à rajouter des jetons (autant que la capacité de l'arc) en aval de la transition. Attention, il n'y a pas nécessairement conservation du nombre de jetons.

On s'intéresse à la simulation à événements discrets de réseaux de Petri. La simulation consiste à faire évoluer le système depuis un marquage initial et de compter le temps passé dans chaque marquage. La simulation à événements discrets consiste à gérer un échéancier des événements se produisant dans le futur. On exécute le prochain événement, on génère les événements résultants et on modifie le marquage et l'échéancier.

Travail à fournir

1. un mécanisme permettant la saisie et la visualisation d'un réseau de Petri ;
2. un mécanisme permettant de le simuler ;
3. calculs de moments d'ordre quelconque sur la distribution empirique calculée ;
4. une visualisation de la trajectoire du marquage. Cette trajectoire doit pouvoir être visualisée sous forme graphique.

Calendrier

Remise du cahier des charges	Mercredi 11 mars 2020
Présentation orale	Mardi 17 mars 2020
Remise des spécifications	Mardi 21 avril 2020
Remise du compte rendu	Lundi 25 mai 2020
Soutenance et démonstration	Jeudi 28 mai 2020

Le cahier des charges ainsi que le compte rendu devront obligatoirement être tapés en L^AT_EX.

Autour de l'algorithme cryptographique RSA

Résumé du sujet

L'algorithme RSA, inventé par Ronald Rivest, Adi Shamir et Leonard Adleman, a été présenté publiquement pour la première fois dans le numéro d'août 1977 de la revue *Scientific American*. C'est actuellement le cryptosystème à clé publique le plus utilisé au monde. On le trouve dans un nombre toujours croissant de produits commerciaux liés à la sécurisation des échanges de données sur Internet, à la protection de la confidentialité et de l'authenticité du courrier électronique, au paiement électronique au moyen de cartes à puce, etc.

Cela fait maintenant 25 ans que la communauté cryptographique mondiale étudie la solidité du RSA face à toute une panoplie d'attaques. Et il a pour l'instant résisté.

Mathématiquement, on peut décrire l'algorithme RSA de la manière suivante. On commence par choisir l'exposant public e (des exemples courants sont $e = 3$, $e = 17$, $e = 257$ ou $e = 65537$). On utilise ensuite un générateur de nombres aléatoires pour obtenir deux nombres premiers p et q tels que e soit premier avec $p - 1$ et avec $q - 1$. Si on pose $n = p \times q$, la clé publique est alors constituée de e et de n , alors que la clé secrète (ou privée) est constituée de p et q . Typiquement on prend souvent actuellement p et q de taille 512 bits, et donc n de taille 1024 bits. La fonction de chiffrement est alors définie par $f : x \mapsto y = x^e \bmod n$ et la fonction de déchiffrement par $f^{-1} : y \mapsto x = y^d \bmod n$, où d est l'inverse de e modulo $\text{ppcm}(p - 1, q - 1)$. Ici d est également une valeur qui doit rester secrète.

Travail à fournir

1. Implémentation de fonctions mathématiques agissant sur des entiers de longueur arbitraire (écrits sur plusieurs dizaines d'octets).
2. Étude et implémentation de méthodes de recherche de grands nombres premiers
3. Établissement d'un système d'édition de clefs publiques.
4. Implémentation de l'algorithme RSA.
5. Implémentation d'un système de signature permettant la validation des documents.

References

- [1] A. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996. Version électronique disponible sur <http://www.cacr.math.uwaterloo.ca/hac>

Calendrier

Remise du cahier des charges	Mercredi 11 mars 2020
Présentation orale	Mardi 17 mars 2020
Remise des spécifications	Mardi 21 avril 2020
Remise du compte rendu	Lundi 25 mai 2020
Soutenance et démonstration	Jeudi 28 mai 2020

Le cahier des charges ainsi que le compte rendu devront obligatoirement être tapés en \LaTeX .

Simulation de réseaux de files d'attente

Résumé du sujet

Un réseau de files d'attente est caractérisé par les files qui le composent. Une file est caractérisée par :

- la description du service reçu ;
- l'algorithme d'ordonnancement des clients avant le service ;
- la fonction de routage vers une autre file.

Toutes ces informations sont des fonctions qui peuvent utiliser en paramètre l'état des files composant le réseau. L'état d'une file est le nombre (et éventuellement l'ordre) des clients qui sont présents dans la file.

La simulation à événements discrets consiste à gérer un échéancier des événements se produisant dans le futur. On exécute le prochain événement, on génère les événements résultants et on les insère dans l'échéancier.

Travail à fournir

Vous devez fournir un package permettant de faire une simulation d'un réseau de files d'attente. Ce package doit permettre la description d'un réseau de files élémentaires ainsi que la simulation à événements discrets de ce réseau.

1. un mécanisme de saisie d'un réseau de files d'attente à partir d'un fichier texte.
2. un mécanisme de simulation à événements discrets d'un réseau de files d'attente.

Certains algorithmes d'ordonnancement, de service et de routage seront programmés dans le package afin de faciliter les descriptions de files simples. Mais il doit être possible à l'utilisateur de créer une file avec ses propres fonctions de description. Les lois de service exponentielles, uniformes, constantes devront également être présentes.

Calendrier

Remise du cahier des charges	Mercredi 11 mars 2020
Présentation orale	Mardi 17 mars 2020
Remise des spécifications	Mardi 21 avril 2020
Remise du compte rendu	Lundi 25 mai 2020
Soutenance et démonstration	Jeudi 28 mai 2020

Le cahier des charges ainsi que le compte rendu devront obligatoirement être tapés en L^AT_EX.