Département d'Informatique
Faculté des Sciences
Université de Mons

Propositions de Projets Master 1
Année 2018-2019

Contents

1 Illustration du développement d'une nouvelle province pétrolière

Service: Systèmes d'Information
Directeur: P. Brocorens et J. Wijsen
Rapporteurs: *à définir*

Description

Dans les années 1950s, le géophysicien M.K. Hubbert développa un modèle permettant d'estimer l'évolution future de la production pétrolière qui, sans contraintes, suit une courbe en cloche. Il présenta son modèle en 1956 en prévoyant que le sommet de la courbe serait atteint entre 1965 et 1970 pour les Etats-Unis. Ce pays connut effectivement un pic en 1970, assurant la notoriété à Hubbert et à son modèle, qui est encore aujourd'hui utilisé pour prévoir le futur de la production pétrolière mondiale.

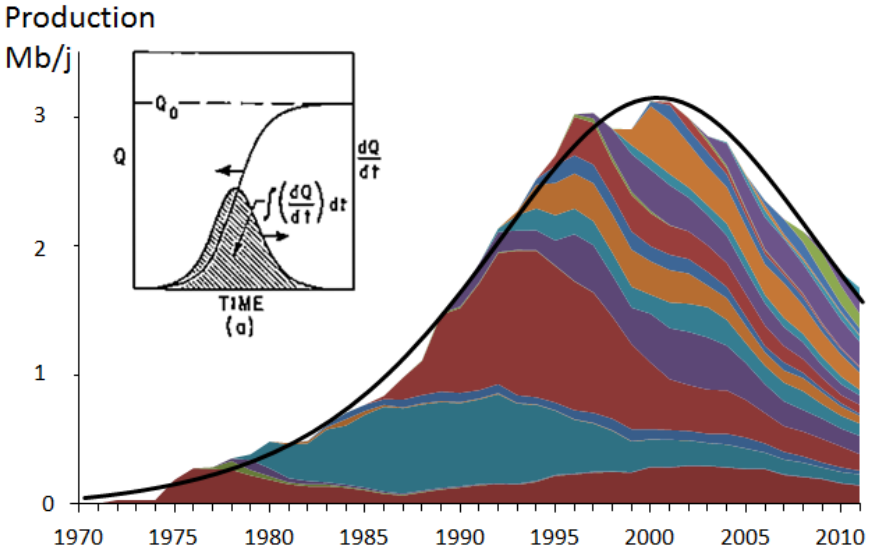


Figure 1: Modèle d'évolution de la production pétrolière développé par M.K. Hubbert dans les années 1950s (encart), et évolution de la production pétrolière de Norvège.

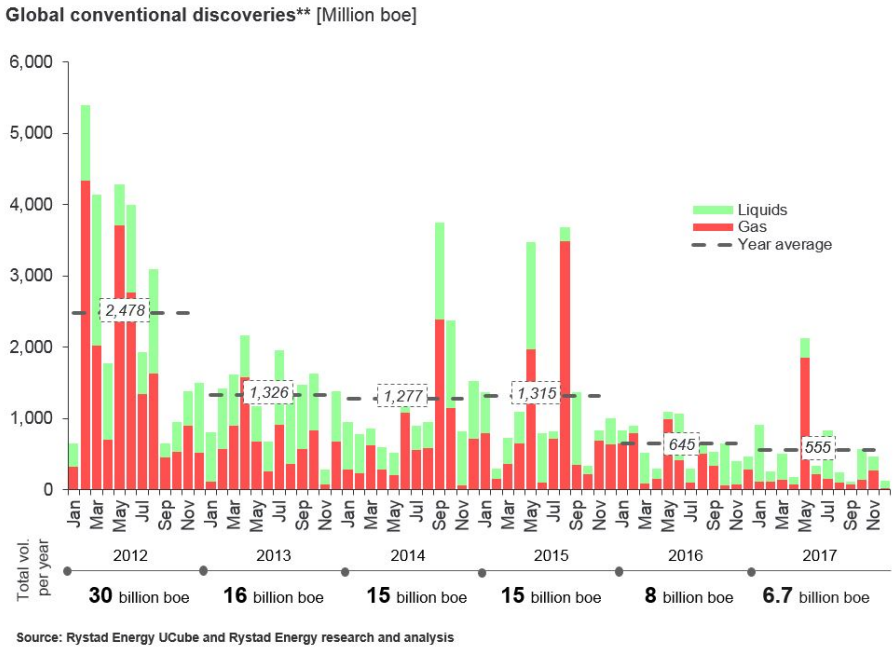
A priori, ce n'est une évidence pour personne que, sans contraintes, la production doit suivre une courbe en cloche. L'utilisation d'un modèle de mise en valeur d'une nouvelle province pétrolière, et la visualisation des résultats grâce à un programme, devrait permettre de supporter (ou non) ce modèle. Ce modèle est en effet similaire à certains modèles utilisés par les biologistes pour étudier les croissances de population en présence de limites environnementales. Et ici, on a croissance de l'extraction d'une ressource limitée. Methodologie : Il s'agit d'une première approche, simplifiée. Un ≪ territoire ≫ quadrillé contient un certain nombre de gisements dissimulés dans des zones favorables. Ces zones favorables sont sondées aléatoirement et les gisements découverts mis en production. L'évolution des découvertes et des productions est suivie au cours du temps en même temps que l'exploration du territoire. Un certain nombre de facteurs issus du monde réel sont inclus dans le modèle tels que des ressources finies, une distribution réaliste des tailles de gisements (loi fractale parabolique), une production qui peut se maintenir jusqu'à un taux maximal de déplétion, et un déclin qui suit une loi exponentielle. Si possible, programmation en R, pour visualiser les résultats sur une page web avec Shiny R.

2 Extraction de données pétrolières sur internet

Service: Systèmes d'Information
Directeur: P. Brocorens et J. Wijsen
Rapporteurs: *à définir*

Description

Un grand nombre de données pétrolières sont confidentielles, et très coûteuses à obtenir auprès de compagnies spécialisées. Cependant, certaines de ces données apparaissent parfois ici ou là, présentées sous forme de graphiques accompagnés ou non de valeurs numériques. Par exemple ci-dessous, certaines valeurs numériques sont disponibles, mais pas le détail par mois, ni le détail entre pétrole et gaz.



- Quels sont les outils informatiques permettant d'extraire des données numériques de tels graphiques, et quelle précision peut-on en attendre ?
- Quels sont les outils informatiques permettant de rechercher de façon optimale sur internet ces informations sous forme de données numériques ou graphiques ?

3 Création d'une base de données pétrolières extraites de sites web

Service: Systèmes d'Information
Directeur: P. Brocorens et J. Wijsen
Rapporteurs: *à définir*

Description

Un grand nombre de données pétrolières sont confidentielles, et très coûteuses à obtenir auprès de compagnies spécialisées. Un petit nombre de ces données sont publiées gratuitement par des agences gouvernementales, mais elles sont réparties dans plusieurs fichiers (xlsx, txt ou csv), et ne sont pas toujours utilisables facilement. Elles sont aussi parfois mises à jour régulièrement.

Le but du travail est de rassembler un certain nombre de ces données dans une base de données optimisée (en Mysql si possible), et qui suit les mises à jour gouvernementales (si possible).

Voir par exemple

- Bureau of Safety and Environmental Enforcement, lequel reporte les données pétrolières du Golfe du Mexique.
  `https://www.data.bsee.gov/Main/RawData.aspx`
- Norwegian Petroleum Directorate, pour les productions de Norvège
  `http://www.npd.no/en/Topics/Resource-accounts-and--analysis/Temaartikler/Resource-accounts/`

4 Alpha-acyclicité

Service: Systèmes d'Information
Directeur: Jef Wijsen
Rapporteurs:

Description

Le cours de *Bases de données II* à Mons établit un lien direct entre la notion d'α-acyclicité des hypergraphes et le calcul efficace d'une jointure de plusieurs relations. Dans ce cours, deux définitions équivalentes d'α-acyclicité émergent, s'appuyant sur la réduction GYO et l'arbre de jointure. Cependant, le cours ne détaille pas la complexité de tester si un hypergraphe est α-acyclique.

Tarjan et Yannakakis [1] ont présenté un algorithme linéaire pour tester si un hypergraphe est α-acyclique. L'objectif de ce projet est d'étudier cet algorithme de Tarjan et Yannakakis, de l'implémenter et de vérifier expérimentalement sa linéarité.

Exigences ou prérequis

Avoir "bien" réussi le cours de *Bases de données II*.

References

[1] Robert Endre Tarjan and Mihalis Yannakakis. "Simple Linear-Time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Acyclic Hypergraphs". In: *SIAM J. Comput.* 13.3 (1984), pp. 566–579. DOI: 10.1137/0213035. URL: https://doi.org/10.1137/0213035.

5 Algorithme de De Fraysseix

Service: Systèmes d'Information
Directeur: Jef Wijsen
Rapporteurs:

Description

De Fraysseix et al. [1] ont démontré que chaque graphe planaire de n sommets peut être dessiné sur une grille de $(2n-4) \times (n-4)$ tel que les arcs sont des lignes droites qui ne se croisent pas. Ce dessin peut être obtenu en temps $\mathcal{O}(n \log n)$. La figure 5 montre un exemple. L'objectif du projet est d'étudier et d'implémenter cet algorithme, ainsi que ses généralisations [2].

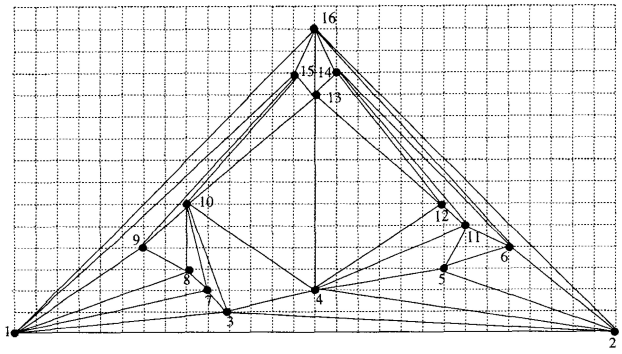


Figure 2: Dessin d'un graphe planaire

References

[1] Hubert de Fraysseix, János Pach, and Richard Pollack. "How to draw a planar graph on a grid". In: *Combinatorica* 10.1 (1990), pp. 41–51. DOI: 10.1007/BF02122694. URL: http://dx.doi.org/10.1007/BF02122694.

[2] Takao Nishizeki and Md. Saidur Rahman. *Planar Graph Drawing*. Vol. 12. Lecture Notes Series on Computing. World Scientific, 2004. ISBN: 981-256-033-5. DOI: 10.1142/5648. URL: http://dx.doi.org/10.1142/5648.

6 Génération automatique d'interfaces utilisateurs pour outils en ligne de commande

Service: Génie Logiciel
Directeur: Alexandre Decan
Rapporteurs: *à définir*

Description

Malgré leur nombre et leur diversité, les outils en ligne de commande (tels que "ls", "git", etc.) présentent souvent de nombreuses similarités dans l'usage de leur interface en ligne de commande (CLI) : paramètres positionnels (ex. "ls [FILE]"), paramètres courts (ex. "ls -l"), paramètres longs (ex. "ls –all"), etc. Le rôle et l'usage de ces différents paramètres sont typiquement détaillés dans la documentation de l'outil, généralement accessible via le paramètre "-h " ou "–help" (ex. "git –help"). La manière d'écrire cette documentation est libre, mais des conventions ont émergé au fil des années.

docopt est un langage de description d'interfaces en ligne de commande basé sur ces conventions. docopt, en plus de formaliser ce langage de description d'interfaces en ligne de commande, propose des parseurs de telles documentations pour de nombreux langages.

L'objectif du projet est d'écrire un outil de génération automatique d'interfaces utilisateurs pour des outils en ligne de commande.

reposant sur le parseur docopt. Cet outil analyse la documentation d'une CLI et génère automatiquement dans une interface graphique des champs (texte libre, case à cocher, liste déroulante, ...) que l'utilisateur peut compléter, et qui seront ensuite utilisés pour exécuter l'outil en ligne de commande.

Cet outil doit être capable d'extraire de la documentation les différents usages (paramètres, types, usage) à l'aide de docopt et de les présenter dans une interface graphique à l'aide de différents widgets (texte libre, case à cocher, fichier, liste déroulante, ... en fonction du type). L'utilisateur peut ensuite compléter ces champs et exécuter l'outil en ligne de commande sous-jacent à l'aide de ces paramètres. L'outil doit également permettre de personnaliser l'UI ainsi présentée à l'utilisateur, en permettant de définir quels sont les champs qui doivent ou peuvent être documentés, ceux qui doivent être visibles ou masqués (voire éventuellement inaccessibles tant qu'un autre champ n'est pas complété), grouper les paramètres liés aux sous-commandes, etc. Cette personnalisation est ensuite stockée dans un fichier de configuration réutilisable lors des appels ultérieurs.

- docopt - `http://docopt.org/`

7 Environnement de simulation pour robots tondeuses

Service: Génie Logiciel
Directeur: Alexandre Decan
Rapporteurs: *à définir*

Description

Les tondeuses autonomes pour jardin sont des robots équipés de plusieurs capteurs et évoluant en toute autonomie dans le jardin en vue de procéder à sa tonte. L'usage de robots tondeuses dans un jardin nécessite l'installation d'un câble périmétrique délimitant les zones dans lesquelles le robot est autorisé à évoluer.

Ces robots sont bien souvent bardés de nombreux capteurs, typiquement deux capteurs magnétiques pour détecter le câble périmétrique, un ou plusieurs capteurs de collision, un capteur de température, un capteur d'humidité, ... Ces nombreux capteurs permettent au robot de "faire le travail" en suivant un comportement aléatoire dont les règles sont pré-enregistrées et définies par le constructeur. Si, dans la majorité des cas, le robot réalise sa tâche avec succès, il y a de nombreuses situations (dépendant de la géométrie des zones, des obstacles ou encore du modèle de robot) où des lacunes importantes apparaissent à l'usage (contournement des obstacles, ilôts isolés, ...). Malheureusement, le comportement de ces robots n'est que rarement modifiable par l'utilisateur, et même dans les cas où cela s'avère possible, l'effort de programmation ne fut-ce que pour tester une solution alternative est important.

Les statecharts sont un formalisme visuel bien connu pour la modélisation du comportement exécutable de systèmes complexes basés sur des événements. Ils ont été inventés dans les années 80 par David Harel, et sont largement adoptés depuis leur intégration au standard UML. Les statecharts proposent des mécanismes plus avancés que les classiques machines à états. Par exemple, les statecharts supportent la composition hiérarchique d'états, les états parallèles, les gardes (conditions) sur les transitions, etc. Il existe de nombreuses sémantiques et de nombreux outils pour exécuter les statecharts. Sismic est l'un d'eux. Il s'agit d'une librairie Python permettant notamment l'exécution complète d'un statechart définit au format YAML (un format textuel aisément lisible), suivant la sémantique UML/SCXML.

L'objectif de ce projet est de concevoir un environnement de simulation de robots tondeuses dont le comportement est décrit par des statecharts exécutés par Sismic. Ce simulateur doit permettre *le chargement de "jardins"* correspondant à l'environnement dans lequel le robot va évoluer, doit offrir une *physique 2D basique mais réaliste* assurant que le robot évolue dans son environnement en respectant (notamment) la géométrie et les obstacles de ce dernier, les contraintes physiques simulées des moteurs, les capteurs, etc. Le simulateur doit pouvoir fournir des *statistiques de tonte* afin de pouvoir évaluer l'efficacité de l'IA du robot. Enfin, le simulateur doit *générer les événements* (issus des capteurs simulés) à destination du statechart, et *réagir aux instructions* envoyées par le robot (par exemple, le contrôle des moteurs).

- Sismic - `https://pypi.org/project/sismic/`

8 Empirical Analysis of Library Usage in Open Source Software Projects

Service: Génie Logiciel
Directeur: Tom Mens
Rapporteurs: *à définir*

Description

When developing software, programmers heavily rely on external software libraries to realise part of the functionality of their application. These libraries can be very diverse and are accessed via their API (Application Programmer Interface), defining the set of features (e.g. classes, interfaces, operation signatures, annotations, etc...) through which the functionalities of the library can be used. Gaining a better understanding of how libraries are effectively used in practice by existing software projects can be very useful to library developers, as it allows them to understand which of the APIs features are used very frequently, and which of them are rarely or never used. Based on this, the API developers can make informed decisions about which changes to make in future versions of their API. For example, API features that are very frequently used should only be changed in backward compatible ways, while rarely used features should be promoted or could be removed or deprecated in newer versions.

The goal of this project is to develop a generic application that takes as input a given (set of) APIs, and open source projects, and that statically analyses the source code of the projects to determine how frequently each of the APIs functionalities are accessed, and how this is distributed over the considered projects. As a proof-of-concept, the student should validate the tool he has developed by carrying out an empirical analysis of the usage of a small number of different libraries/APIs by a large number of different open source software projects. The results should be reported to the user in different forms:

- comma-separated value (CVS) files that can easily be processed by statistical analysis tools or spreadsheets
- a web-based dashboard that visually displays the API usage statistics
- textual reports in PDF or HTML format

The technical details of how to realise the project are left to the discretion of the student, after agreement of the supervisor. Based on an objective and convincing motivation, the student is free to select :

- the programming language of choice in which to implement his tool.
- the programming language of choice in which the libraries and software projects to be analysed are developed. Popular languages like Java are likely to have a much wider range of libraries available, as well as a wider range of open source projects using them.
- the set of software projects to consider for analysis, and the open source repository from which to extract these projects (and their histories). Existing open source repositories of interest could be, for exemple, SourceForge or GitHub.
- the set of libraries/APIs to be considered. The more popular APIs are probably more interesting to study, as it will be easier to identify and analyse projects using these APIs.

[Optional (bonus)] If time permits, it could be very useful to provide tool support to analyse the evolution of the API usage over time. This involves two dimensions:

1. Study how the use of a given API version changes across successive versions of (a given set of) software projects
2. Study how the API usage evolves over different API versions for a given set of software projects
3. Study how the functionalities of an API evolve over time

Exigences ou prérequis

Un intérêt pour le génie logiciel et la recherche empirique. Un intérêt fort en open source. Une bonne expérience en programmation.

9 An evolutionary comparison of the health of open source e-learning management systems

Service: Software Engineering Lab
Directeur: T. Mens and R. Viseur
Rapporteurs: *à définir*

Description

The general goal of this project is to quantitatively study the real impact of important events during the development history of open source software ecosystems and their supporting communities. In particular, some events may result in a disengagement of the community, or even lead to forking the project. Several such cases of evolution problems have been identified for different open source projects.

In this project, we will study and compare different open source software systems have been proposed for collaborative e-learning content management. Examples are Chamilo `https://www.chamilo.org`, Claroline `https://claroline.net` and its successor Claroline Connect. Claroline has witnessed a decreasing interest of its community over time, leading to an abandonment of this project, and necessitating its replacement by its successor Claroline Connect. In parallel to this, another similar project called Chamilo has been evolving over time.

The goal is to use a variety of historical data sources (e.g. source code repositories, bug and issue trackers, mailing lists, etc...) for these systems, in order to analyse and compare their evolution over time according to different factors of interest: source code quality (using e.g. tools like Sonarqube), contributor involvement, productivity. Data extraction tools like the Grimoire Lab tool suite could be used to extract the required information. Data analysis tools (e.g. Python libraries) could be used for analysing and interpreting the extracted data.

Some of the research questions we would like to explore based on the extract data are:

- Is it possible to detect the abandonment of the Claroline community based on the quantitative data? Which types of metrics allow to detect this?
- Can one measure and compare the effort of the different partners of the Claroline Connect partners within the Claroline consortium? How and to which parts of the project does each partner contribute? How did these efforts evolve over time?
- Would it have been possible to anticipate or predict the abandonment of key partners in the analysed projects?
- Can one observe migration of developers from the Claroline community to its competitor Chamilo or vice versa? Chamilo was initially created as a fork of Claroline. Was the creation of this fork beneficial for the Chamilo contributors?
- Are there observable qualitative differences and similarities between Chamilo, Claroline Connect, and perhaps other competing projects?

10 Using Constraint Logic Programming and Model Checking for University Scheduling Problems

Service: Software Engineering Lab
Directeur: T. Mens
Rapporteurs: *à définir*

Description

Many processes and activities in a university correspond to non-trivial scheduling and planning problems. The goal of this project is to select and use a language and/or tool based on constraint logic programming for this purpose, evaluate its usefulness, and develop different proof-of-concept prototype implementations (including an interactive and "convivial" visual user interface for different scheduling problems.

No specific programming language is imposed for this project. The motivation and choice of the most appropriate CLP or model checking tool and language will be part of the project's analysis. Possible solutions include:

- Choco, a free Open-Source Java library dedicated to Constraint Programming:
  `http://www.choco-solver.org`
- Constraint Logic Programming in Prolog, using one of its supported constraint solvers (e.g. CLP(FD) for integers, or CLP(B) for Boolean variables):
  `http://www.swi-prolog.org/pldoc/man?section=clp`
- Alloy, a tool and language for solving models based on logic constraints:
  `http://alloy.lcs.mit.edu/alloy/`
- OptaPlanner is an efficient open source constraint solver and planning engine implemented in Java:
  `https://www.optaplanner.org`
- ECliPSe CLiPSe is an open-source solution for developing constraint programming applications in the areas of planning, scheduling, resource allocation, timetabling, transport etc. It contains several constraint solver libraries, a high-level modelling and control language, interfaces to third-party solvers, an integrated development environment and interfaces for embedding into host environments.
  `http://eclipseclp.org`

At least two case studies need to be carried out as part of this project, to evaluate and/or compare the selected approach. The first case study involves the process for a student's "PAE": which courses should a student select in which year in order to satisfy all credits, all course prerequisites, as well as any possible scheduling conflicts. (This should be supported both from the student's point of view, as well as from the department's point of view.) The second case study is more traditional, as it involves defining a course schedule, taking into account the above PAE, the availability of teachers and rooms, and many other factors.

Both case studies should involve not only a CLP-based solution, but also a nice and easy-to-use visual interface to interact with the application; to define, modify and verify the constraints; and to explore the proposed solutions.

11 An empirical study of the health of the Drupal software ecosystem and community

Service: Software Engineering Lab
Directeur: T. Mens and A. Decan
Rapporteurs: *à définir*

Description

Drupal (see `https://www.drupal.org`) is an open source content management system, developed in PHP. It has an active developer and user community (see `https://www.drupal.org/community`) with regular local meetings (e.g. `https://drupalcamp.be`). It also has a surrounding ecosystem of companies developing external software and consultancy services on top of Drupal (see `https://www.drupal.org/try-drupal`) The Drupal source code is available through a git distributed version control repository: `https://cgit.drupalcode.org/project/` The Drupal Core (see `https://www.drupal.org/project/drupal`) can be extended with a high number of projects and distributions (see `https://www.drupal.org/project/project_distribution`).

This project fits within the ongoing interuniversity research projects SECOHealth (`https://secohealth.github.io/`) and SECO-ASSIST (`https://secoassist.github.io`) carried out by the Software Engineering Lab. In this context we have empirically studied the evolution and health factors of many software ecosystems in the past (Gnome, Debian, and package dependency managers for a wide variety of programming languages: npm, RubyGems, CRAN and many more). Some example publications can be found on `https://secoassist.github.io/results.html` and `https://secohealth.github.io/results.html`.

In line with this research, the goal of this masters project is to analyse the evolution of the Drupal ecosystem over time, in order to assess its health, longevity, maintainability and related factors from different points of view (social, technical, commercial, ...). The empirical and historical analysis of Drupal will involve extracting evolutionary data from different data sources (the Drupal website, its version control repository, bug and issue tracker, mailing lists, ...), define and use metrics related to the health of the ecosystem, statistically analyse these metrics over time and visually report the results, and compare these results with other software ecosystems.

The preferred language of choice for carrying out most of the research will be Python, because of its many existing tools and libraries for data analysis and data mining.

12 A tool for component-based software development based on executable statecharts

Service: Software Engineering Lab
Directeur: T. Mens and A. Decan
Rapporteurs: *à définir*

Description

Sismic (`https://pypi.org/project/sismic/`) is a well-documented (`https://sismic.readthedocs.io`) open source Python 3 library that has been developed for executing, simulating and testing executable statecharts. Statecharts are a visual modeling language based on hierarchical state machines. The Sismic library provides support for a variety of well-known software engineering techniques, such as design by contract, runtime verification of properties, and behaviour-driven development. More details can be found in the scientific journal publication "A method for testing and validating executable statechart models" (`https://doi.org/10.1007/s10270-018-0676-3`).

The research domain of component-based software development (CBSD) is concerned with developing software in terms of independent software components that can interact and communicate in predefined ways using specific interaction protocols. The goal of this project is to implement an event-based CBSD framework, in which different components can communicate with one another by sending and receiving events through the input ports and output ports on these components. The components themselves can be seen as black boxes, whose internal behaviour can be implemented in any way (e.g. using Python code, or using Sismic statecharts). Assuming that the components are implemented as Sismic statecharts, the goal is to implement an extension of Sismic that facilitates the dynamic creation and execution of component-based architectures, linking each component's behaviour to its internal statechart implementation, and defining and controlling inter-component interaction.

It would be desirable to have at the intercomponent level the same kind of testing and validation mechanisms as those provided by Sismic, i.e., design by contract, runtime property verification, unit testing, and behaviour-driven development.

Part of the project will be devoted to exploring existing CBSD solutions (and in particularly those based on statecharts) in order to decide on the most appropriate ways to implement such an approach for Sismic.

Exigences ou prérequis

Excellent programming skills in Python are required.

13 Generating Python Sismic statechart code from Yakindu statecharts

Service: Génie Logiciel
Directeur: Tom Mens
Rapporteurs: Alexandre Decan et *autre personne à définir*

Description

[Etant donné que la majorité de la littérature scientifique est en anglais, cette proposition de sujet de projet est rédigé en anglais également.]

Statecharts (a.k.a. behavioural state machines) are of the UML diagram types that allow to specify the executable behaviour of real-time event-driven systems.

Yakindu Statechart Tools `https://eclipse.org/papyrus/` is a powerful visual modelling environment for visualising and simulating statecharts, as well as the ability to generate (Java and C++) code from these statecharts. *Sismic* (`https://github.com/AlexandreDecan/sismic`) is a Python library for interpreting and executing statecharts.

The goal of this project is to combine both tools, by exploiting the visual capabilities of Yakindu to specify statecharts, and generate Sismic statecharts that can be executed using the Sismic library. The novel features of Sismic (such as the possibility to specify contracts over statecharts) should also be integrated in Yakindu.

Given a Sismic statechart as input, it should also be possible to generate Yakindu statecharts.

Exigences ou prérequis

Excellent programming experience in Java and Python. An interest in UML and software modeling.

14 Développer une SmartWatch avec des Statecharts et Arduino

Service: Génie Logiciel - Réseaux et télécommunications
Directeur: Tom Mens et Bruno Quoitin
Rapporteurs: à définir

Description

L'objectif du projet est la conception d'une horloge de bureau multifonction sur Arduino. Cette horloge doit intégrer un minuteur et un chronomètre et est contrôlable à l'aide de 3 boutons.

L'horloge doit être modélisée à l'aide du formalisme des statecharts et le code de l'implémentation réelle sera généré automatiquement à partir du modèle. Il est recommandé d'utiliser à cette fin l'outil Yakindu car il permet déjà de génerer du code pour Arduino à partir d'un statechart donné:
`blogs.itemis.com/en/developing-software-for-arduino-with-yakindu-statechart-tools`

Concernant la partie technique, l'horloge de bureau devrait être capable de se synchroniser automatiquement, par le réseau WiFi, avec une horloge de référence. Le protocole à utiliser sera Network Time Protocol (NTP) [1]. L'horloge sera développée sur la plateforme Espressif ESP8266 [2], un System-on-Chip intégrant un transceiver Wifi. Le SoC contrôlera un afficheur LCD ou LED afin d'afficher l'heure actuelle.

Finalement, une attention particulière sera prêtée à la consommation énergétique de l'implémentation afin de permettre son alimentation par batterie. La consommation électrique du SoC ESP8266 étant dominée par la celle du transceiver WiFi, le projet considèrera sérieusement l'usage des modes basse consommation du SoC et la mise en oeuvre d'un cycle éveil/veille dans lequel le SoC est endormi la plupart du temps.

Exigences ou prérequis

Intérêt pour la modélisation logicielle (notamment les statecharts), la génération automatique du code, les réseaux informatiques et les systèmes embarqués. Envie de "bidouiller".

References

[1] Jack Burbank, William Kasch, Professor David L. Mills, and Jim Martin. *Network Time Protocol Version 4: Protocol and Algorithms Specification*. RFC 5905. June 2010. DOI: 10.17487/rfc5905.

[2] Espressif Systems. *ESP8266EX Datasheet, version 4.3*. `https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf`. June 2015.

15 Réseau Wi-Fi multi-sauts sur plateforme ESP

Service: Réseaux et Télécommunications
Directeur: B. Quoitin
Rapporteurs: *à définir*

Description

La portée d'une communication Wi-Fi point-à-point est typiquement limitée à quelques dizaines ou centaines de mètres. Elle dépend entre autres de la fréquence utilisée, de la puissance de l'émetteur, de la sensibilité du récepteur et de l'environnement. Afin d'étendre cette portée, des communications multi-sauts peuvent être utilisées. Dans celles-ci, les noeuds Wi-Fi agissent comme relais pour les communications d'autres noeuds. Etant données les conditions de transmission changeantes, un protocole de routage dynamique détermine les relais par lesquels passer pour joindre chaque destination.

L'objectif de ce projet est de mettre en oeuvre des communicatons Wi-Fi multi-sauts dans un réseau composé de systèmes embarqués. Les noeuds de ce réseau sont basés sur un System-on-Chip intégrant un micro-contrôleur et un transceiver Wi-Fi. Les solutions ESP8266 ou ESP32 d'Espressif sont envisagées en raison de leur très faible coût.

Le projet doit également considérer la consommation énergétique des noeuds. Chacun est alimenté par batterie, ce qui implique qu'une attention particulière devra être pretée au coût des communications. Un cycle éveil-veille devra très probablement être établi.

Un point de départ possible pour le projet pourrait être le SDK ESP-Mesh [1] qui permet de créer un réseau multi-sauts sous forme d'arbre. Cependant, d'autres organisations de réseaux sont possibles et souhaitables, étant donné qu'un arbre est peu robuste en cas d'interférences ou de panne. Un protocole standard tel qu'AODV [2] pourrait être envisagé.

Le projet visera à expérimenter avec le réseau ainsi créé de façon à en valider les fonctionnalités. Par ailleurs, sur base du prototype créé, il serait possible d'analyser la longueur des chemins établis, d'étudier la durée de vie du réseau sur batteries ainsi que d'estimer les délais des communication et de l'établissement de routes.

Exigences ou prérequis

Intérêt pour les systèmes embarqués et les réseaux informatiques.

References

[1] *SDK ESP-Mesh*. `https://github.com/espressif/esp-mdf`.

[2] Samir R. Das, Charles E. Perkins, and Elizabeth M. Belding-Royer. *Ad hoc On-Demand Distance Vector (AODV) Routing*. RFC 3561. July 2003. URL: `https://rfc-editor.org/rfc/rfc3561.txt`.

16 Dompter son Arduino avec la programmation fonctionnelle

Service: Réseaux et Télécommunications
Directeur: B. Quoitin
Rapporteurs: D. Hauweele et *personne à définir*

Description

Le développement d'applications pour systèmes embarqués communicants est réputé difficile et sujet à de nombreuses erreurs de programmation. Plusieurs facteurs sont à l'origine de ces erreurs tels que l'absence de mécanisme de protection de la mémoire, l'absence de système d'exploitation complet, le contrôle de périphériques matériels et l'utilisation de langages de programmation de bas niveau (typiquement le langage C voire un langage d'assemblage).

L'objectif du projet est d'investiguer les possibilités de développer des programmes pour une plateforme embarquée (Arduino ou similaire) en utilisant un *Domain Specific Language* (DSL) embarqué dans un langage fonctionnel [1]. Une telle approche permet de respecter certaines contraintes par construction. Le projet s'intéressera en particulier au langage Haskell [2] et aux frameworks Atom [3] et CoPilot[4, 5].

Afin d'illustrer l'usage d'Haskell/CoPilot, le projet visera à développer un DSL permettant d'exprimer la machine à états de protocoles de communication simples (p.ex. bit/byte stuffing ou Manchester encoding). Les programmes décrits avec ce DSL seront ensuite testés sur une plateforme embarquée réelle.

On pourra s'inspirer d'un projet similaire `https://leepike.wordpress.com/2010/12/18/haskell-and-hardware-for-the-holidays`.

Exigences ou prérequis

Intérêt pour les systèmes embarqués. Bonne connaissance du langage de programmation C. Notions de programmation fonctionnelle.

References

[1] A. Gill. "Domain-specific Languages and Code Synthesis Using Haskell – Looking at embedded DSLs". In: *ACM Queue* 12.4 (May 2014).

[2] *Haskell*. `https://www.haskell.org`.

[3] *Atom*. `https://hackage.haskell.org/package/atom`.

[4] L. Pike, N. Wegmann, S. Niller, and AL Goodloe. *CoPilot: Monitoring Embedded Systems*. Tech. rep. NASA/CR-2012-217329. Jan. 2012.

[5] *CoPilot*. `http://hackage.haskell.org/package/copilot`.

17 Code détecteurs et correcteurs d'erreurs : découverte et expérimentation

Service: Réseaux et Télécommunications
Directeur: B. Quoitin
Rapporteurs: *à définir*

Description

Les codes détecteurs et correcteurs d'erreur [1, 2, 3, 4] font intimement partie des protocoles de communication utilisés aujourd'hui. La compréhension de leurs principes de fonctionnement n'est cependant pas aisée.

Le premier objectif de ce projet, si vous l'acceptez, est d'établir un **état de l'art** des codes les plus connus/répandus. Parmi ceux-ci devraient notamment figurer le bit de parité, les codes de Hamming, la somme de contrôle Internet, les codes cycliques (CRC), les codes de Reed-Solomon. Pour chacun des codes abordés, le rapport du projet fournira une explication détaillée de son principe de fonctionnement, enrichi d'exemples.

Le second objectif est de fournir un **outil de simulation** de ces codes. Cet outil devra fournir d'une part l'implémentation de certains codes et d'autre part, la possibilité de générer des messages synthétiques, de les corrompre de façon aléatoire (modèle d'erreurs du canal de transmission) et de mesurer le taux de succès (détection et/ou correction) des codes. L'outil devra être suffisamment générique que pour permettre l'adjonction ultérieure de nouveaux codes et modèles d'erreurs.

Exigences ou prérequis

Intérêt pour les réseaux informatiques.

References

[1] William W. Peterson. *Error Correcting Codes*. MIT Press, 1972.

[2] Gerard J. Holzmann. *Design and Validation of Computer Protocols*. Prentice Hall, 1991.

[3] Madhu Sudhan. *Coding Theory : Tutorial & Survey*. Tutorial at the 42nd Annual Symposium on Foundations of Computer Science (FOCS), University of Nevada at Las Vegas, available from `http://madhu.seas.harvard.edu/papers/2001/focs01-tut.pdf`. 2001.

[4] David J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. `http://www.inference.org.uk/itprnn/book.pdf`. Cambridge University Press, 2003.

18 Gestion d'un banc de test pour réseaux de capteurs sans fil

Service: Réseaux et Télécommunications
Directeur: B. Quoitin
Rapporteurs: M. Charlier et *à définir*

Description

L'évaluation des protocoles et applications des réseaux de capteurs sans fil est réalisée la plupart du temps par simulation puis en banc de test avec des noeuds réels. Plusieurs bancs de tests tels que le FIT-IoT-Lab [1] sont disponibles à cet effet pour la communauté scientifique.

Malheureusement, il n'existe pas à ce jour de banc de test mettant en oeuvre des communications radio *Ultra Wideband* (UWB) [2] qui soit disponible publiquement. Pour cette raison, le service Réseaux et Télécommunications déploie progressivement un tel banc de test au 2ème étage du bâtiment De Vinci. Dans sa première phase, ce banc de test devrait comporter 40 noeuds, chacun muni d'un transceiver IEEE 802.15.4 UWB et d'un processeur ARM Cortex-M3 (Texas Instruments CC2538).

L'objectif de ce projet est de concevoir un système de gestion du banc de test. Ce système s'interconnecterait avec chacun des noeuds via une ligne série. Il pourrait ainsi en surveiller l'activité, mais également en effectuer la re-programmation à distance. Le système développé devrait mettre en oeuvre des mécanismes de compression des messages ainsi qu'un horodatage précis de ceux-ci. Le projet devrait également mettre à disposition une interface utilisateur (p.ex. via un ensemble de pages web) permettant l'interaction avec les nœuds sur une carte du réseau.

Le projet devrait respecter les contraintes suivantes

- La gestion du réseau se fait via un Raspberry Pi connecté au réseau de l'UMONS.
- Les noeuds sont connectés au Raspberry Pi via une interface USB-série.
- Le système permet l'écriture de log dans des fichiers.
- Le système permet de limiter la simulation en durée ou en nombre de données récoltées (taille maximale du fichier de log).
- Le système permet l'arrêt et le redémarrage programmés de nœuds afin d'évaluer la réaction de certains protocoles ou applicatifs.
- L'accès à l'interface du système doit-être sécurisée (login/mot de passe).

Ce projet commencera avec une version réduite du banc de test (p.ex. un Raspberry Pi et 5 noeuds). En fonction de la progression du travail un accès à l'ensemble du banc de test sera envisagé.

Exigences ou prérequis

Intérêt pour les systèmes embarqués et les réseaux informatiques.

References

[1] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne. "FIT IoT-LAB: A large scale open experimental IoT testbed". In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. 2015, pp. 459–464. DOI: 10.1109/WF-IoT.2015.7389098.

[2] "IEEE Standard for Low-Rate Wireless Networks". In: *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)* (2016), pp. 1–709. DOI: 10.1109/IEEESTD.2016.7460875.

19 Algorithme de localisation adapté aux réseaux de capteurs sans fil

Service: Réseaux et Télécommunications
Directeur: B. Quoitin
Rapporteurs: M. Charlier et P. Hauweele

Description

La localisation dans les réseaux de capteurs est un domaine étudié depuis des années [1]. Cependant, l'utilisation de matériel limité en ressources (calcul, mémoire, énergie) nécessite l'adaptation des algorithmes qui effectuent la localisation. De plus, la présence de bruit dans les mesures de distance entraîne la nécessité de filtrer les données afin d'obtenir une localisation précise.

L'objectif du projet consiste à d'abord effectuer un état de l'art des algorithmes de localisation pouvant être utilisés dans un réseau de capteurs et d'ensuite fournir un délivrable permettant la localisation d'un nœud basée sur des distances le séparant de points connus dans l'espace (ancres). La localisation s'effectuerait en 2D et si possible en 3D. Afin d'améliorer la robustesse de l'algorithme par rapport au bruit dans les mesures de distance [2], le recours à des filtres de Kalman pourrait être envisagé.

Le projet devrait respecter les contraintes suivantes:

- Être exécutable par un système embarqué de type Zolertia Firefly (fourni).
- Utiliser le système d'exploitation Contiki OS.
- Prendre en entrée (via ligne série) des mesures de distance et retourner une localisation 2D ou 3D.
- Fournir une interface permettant de mesurer l'erreur de localisation à partir d'un grand nombre d'échantillons.
- Veillez à garder le temps de calcul aussi faible que possible.

Exigences ou prérequis

Intérêt pour les systèmes embarqués et les réseaux informatiques.

References

[1] Guangjie Han, Huihui Xu, Trung Q Duong, Jinfang Jiang, and Takahiro Hara. "Localization algorithms of wireless sensor networks: a survey". In: *Telecommunication Systems* 52.4 (2013), pp. 2419–2436.

[2] Maximilien Charlier. *Application d'UWB à la mesure du temps de propagation et à la localisation*. Mémoire de Master en Sciences Informatiques, Université de Mons. 2017.

20 Environnement de gestion automatique des certificats (ACME).

Service: Réseaux et Télécommunications.
Directeur: Alain BUYS
Rapporteurs: à définir.

Description

L'obtention d'un certificat non self-signé nécessite traditionnellement de prouver auprès d'une autorité de certification que l'on est bien détenteur des droits sur le site ou domaine où ce certificat va être utilisé. Cette procédure demande à ce stade un traitement manuel et engendre un coût important.

Des initiatives existent, pour permettre le déploiement automatisé d'une infrastructure à clé publique à très faible coût. L'Internet Security Research Group (ISRG) a créé il y a peu un nouveau protocole ("Automatic Certificate Management Environment") pour son service de création de certificats automatisé "Let's Encrypt". Le but premier de "Let's Encrypt" est d'encourager la conversion de sites web existants de HTTP vers HTTPS. Afin de prouver la propriété du domaine, la technique utilisée actuellement demande que le site concerné soit déjà opérationnel (résolution DNS publique, réponse à des défis sous forme de requêtes HTTP, ...).

Cette procédure de vérification, le rapatriement et l'installation des certificats peuvent être automatisés à l'aide de clients tel que Certbot.

Le projet consistera à faire le point sur le sujet de la certification automatisée en général, notamment en examinant les failles possibles de ces nouveaux systèmes et leurs limitations éventuelles.

Il serait intéressant d'étudier dans quelle mesure les systèmes existants pourraient être adaptés pour une gestion "off-line" de serveurs, de serveurs internes d'une entreprise et notamment l'obtention de certificats utilisés pour d'autres protocoles que HTTP (POP, IMAP, ...).

Références :

`https://datatracker.ietf.org/doc/draft-ietf-acme-acme/`

`https://letsencrypt.org/`

`https://letsencrypt.org/howitworks/technology/`

`https://certbot.eff.org/docs/intro.html`

Exigences ou prérequis

Un intérêt pour la cryptographie et les technologies web.

21 Sécurité du verrouillage central automobile.

Service: Réseaux et Télécommunications.
Directeur: Alain BUYS
Rapporteurs: Alexandre Decan + un autre rapporteur à définir.

Description

Les automobiles récentes sont généralement équipées d'un système de verrouillage centralisé : l'utilisateur peut verrouiller ou déverrouiller les portières et/ou le coffre (diverses variantes existent) à l'aide d'un signal radio-fréquence émis à l'aide de boutons sur la clé. Très tôt, il s'est avéré nécessaire d'empêcher les possibles tentatives de rejeu (le signal est intercepté par un attaquant et reproduit à un moment ultérieur). Le système de protection semble assez basique et évite qu'un même code soit réutilisé, via un "roulement" entre une série de codes générés au préalable. Il apparaît cependant que ce système puisse être contourné en associant l'interception à des techniques de brouillage du signal.

Un des buts du projet sera de faire le point sur l'étendue de cette faille et de voir comment ce système pourrait être amélioré.

Il existe une variante "mains libres" au verrouillage classique décrit ci-dessus : il suffit que la clé soit dans le voisinage immédiat du véhicule pour permettre le verrouillage et le déverrouillage via un bouton ou un senseur sur la poignée de porte. Il s'agira de déterminer dans quelle mesure celle-ci est également vulnérable aux attaques évoquées plus haut.

Références :

https://assets.documentcloud.org/documents/3010178/Volkswagen-amp-HiTag2-Keyless-Entry-System.pdf

https://andrewmohawk.com/2016/02/05/bypassing-rolling-code-systems/ (consulté le 29/5/2017)

https://conference.hitb.org/hitbsecconf2017ams/sessions/chasing-cars-keyless-entry-system-attacks/ (consulté le 29/5/2017)

Exigences ou prérequis

Un intérêt pour la cryptographie et les technologies de type radio-fréquence.

22 Pare-feu domestique.

Service: Réseaux et Télécommunications.
Directeur: Alain BUYS, Bruno QUOITIN
Rapporteurs: à définir.

Description

Au contraire d'ordinateurs plus ou moins "classiques" (laptop, desktop, smartphones) où en principe un anti-virus ou un pare-feu logiciel peuvent aisément être installés, de plus en plus d'objets connectables à internet apparaissent, qui ne présentent pas cette caractéristique (smart TV, systèmes d'alarme, domotique, ...).

Le projet consistera à étudier les techniques de surveillance du traffic et de protection des réseaux domestiques [1, 2, 3, 4, 5]. Il s'intéressera ensuite à la conception en version open-source d'un pare-feu domestique ou tout au moins un système de contrôle/monitoring du trafic (sortant) pour les réseaux filaire et wi-fi.

Exigences ou prérequis

Un intérêt pour la sécurité des réseaux.

References

[1] R. Mortier, T. Rodden, T. Lodge, D. McAuley, C. Rotsos, A.W. Moore, A. Koliousis, and J. Sventek. "Control and Understanding: Owning Your Home Network". In: *Proceedings of the Fourth International Conference on Communication Systems and Networks (COMSNETS)*. 2012.

[2] Abdesselem Kortebi, Zied Aouini, Martin Juren, and Jan Pazdera. "Home Networks Traffic Monitoring Case Study: Anomaly Detection". In: *2016 Global Information Infrastructure and Networking Symposium (GIIS)* (2016), pp. 1–6.

[3] Zied Aouini, Abdesselem Kortebi, and Yacine Ghamri-Doudane. "Traffic monitoring in home networks: Enhancing diagnosis and performance tracking." In: *IWCMC*. IEEE, 2015, pp. 545–550. ISBN: 978-1-4799-5344-8. URL: http://dblp.uni-trier.de/db/conf/iwcmc/iwcmc2015.html#AouiniKG15.

[4] A. B. M. Musa and Jakob Eriksson. "Tracking Unmodified Smartphones Using Wi-fi Monitors". In: *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. SenSys '12. Toronto, Ontario, Canada: ACM, 2012, pp. 281–294. ISBN: 978-1-4503-1169-4. DOI: 10.1145/2426656.2426685. URL: http://doi.acm.org/10.1145/2426656.2426685.

[5] Garen Kutukian and Mohammad Husain. *Raspberry Pi 3 Home Network Monitoring Tool*. Tech. rep. http://www.cpp.edu/~polysec/network/icncc.pdf.

23 Recherche de cycles d'une forme particulière dans les graphes

Service: Informatique Théorique
Directeur: Véronique Bruyère
Rapporteurs: A définir

Description

Un *système réactif* est un système plongé dans un environnement, qui doit réagir continuellement aux événements produits par cet environnement. Un exemple est le système ABS qui assiste au freinage chaque fois que celui-ci doit être intense. Une modélisation classique d'un système réactif est celle d'un graphe orienté fini dont les sommets sont répartis en les sommets appartement au système et les sommets appartenant à l'environnement, et dont les arcs décrivent les interactions entre le système et l'environnement.

Un problème beaucoup étudié est celui de la *synthèse de contrôleur* : on souhaite concevoir un contrôleur qui assure au système réactif de satisfaire une spécification donnée quels que soient les événements produits par l'environnement. Au niveau de la modélisation, la spécification est traduite par une propriété sur le graphe (par exemple atteindre ou éviter un sommet particulier du graphe, passer infiniment souvent par un sommet du graphe, etc.), et concevoir un contrôleur revient à construire une stratégie gagnante du système contre l'environnement pour satisfaire cette propriété.

Dans plusieurs cas de propriétés, l'existence d'une stratégie gagnante dépend de l'existence de certains *cycles* dans le graphe : cycle passant par un sommet d'une certaine forme, cycle de plus petit poids moyen, cycle étiqueté par des tuples de poids tous positifs, etc.

Le but de ce projet est d'étudier en détails plusieurs algorithmes [1, 2, 3] permettant de détecter des cycles d'une forme particulière, et d'implémenter ceux-ci. Une attention particulière sera portée à l'efficacité de ces algorithmes et à la construction de tels cycles.

Exigences ou prérequis

Il faut avoir un goût prononcé pour l'algorithmique et les structures de données avancées.

References

[1] Richard M Karp. "A characterization of the minimum cycle mean in a digraph". In: *Discrete Mathematics* (23 1978), pp. 309–311.

[2] S. R. Kosaraju and G. F. Sullivan. "Detecting cycles in dynamic graphs in polynomial time (preliminary version)". In: *Proceedings of STOC: Symposium on Theory of Computing*. ACM, 1988, 398–406.

[3] Damien Busatto-Gaston, Benjamin Monmege, and Pierre-Alain Reynier. "Optimal Reachability in Divergent Weighted Timed Games". In: *Proceedings of FoSSaCS*. Lecture Notes in Computer Science 10203. Springer, 2017, pp. 162–178.

24 Apprentissage d'automates

Service: Informatique Théorique
Directeur: Véronique Bruyère
Rapporteurs: A définir

Description

L'*apprentissage d'automates* trouve de nombreuses applications en traitement de la parole, en traduction automatique, en vérification et synthèse de systèmes informatiques, en biologie computationelle, en data mining, et même en musique (voir le survey [1]).

Dans le cadre de ce projet, dans un premier temps, l'étudiant étudiera en profondeur certaines techniques *passives* d'apprentissage d'automates où un automate déterministe de taille minimale est appris automatiquement à partir d'un ensemble donné de mots classés comme appartenant ou n'appartement pas au langage accepté par l'automate (voir par exemple [2]). Dans un second temps, l'étudiant appliquera ces techniques d'apprentissage au *regular model-checking* (voir [2, 3]). Le problème du model-checking consiste à vérifier qu'un système informatique satisfait une spécification quand ceux-ci sont donnés sous la forme de modèles. Des spécifications typiques sont : est-ce que le système n'atteint jamais d'état de deadlock ? Est-ce qu'une requête reçoit toujours une réponse ? De nombreux algorithmes de model-checking ont été élaborés [4]. On parle de "regular" model checking quand le système informatique à vérifier est modélisé en termes d'automates finis.
Remarque :

- Ce sujet convient aussi bien pour un projet de Master 1 que pour un mémoire de Master 2. Son niveau de difficulté et son ampleur seront adaptés en fonction.
- Les étudiants intéressés sont invités à me rencontrer pour discuter plus en détails du sujet.

Exigences ou prérequis

Algorithmique et Structures de données. Calculabiilité et Complexité.

References

[1] Colin de la Higuera. "A bibliographical study of grammatical inference". In: *Pattern Recognition* 9 (38 2005), 1332–1348.

[2] Daniel Neider. "Applications of automata learning in verification and synthesis". PhD thesis. RWTH Aachen University, 2014.

[3] Parosh Aziz Abdulla, Bengt Jonsson, Marcus Nilsson, and Mayank Saksena. "A Survey of Regular Model Checking". In: *Proceedings of CONCUR: International Conference on Concurrency Theory*. Lecture Notes in Computer Science 3170. Springer, 2004, pp. 35–48.

[4] Christel Baier and Joost-Pieter Katoen. *Principles of model-checking*. The MIT Press, 2008.

25 Parallélisation pour le calcul d'un grand nombre d'invariants de graphes

Service: Algorithmique
Directeur: H. Mélot
Rapporteurs: P. Hauweele et *personne à définir*

Description

La théorie extrémale des graphes est une branche de la théorie des graphes s'attachant à la recherche de bornes, inégalités optimales, entre invariants (numériques) sur les graphes. Le système PHOEG (PHOEG Helps Obtaining Extremal Graphs), développé dans le Service d'Algorithmique, est utilisé comme un outil d'assistance à la découverte/preuve dans le domaine de la théorie extrémale des graphes. Dans son traitement, ce système nécessite de connaître un grand nombre de données concernant des graphes. Il doit typiquement calculer une valeur pour chaque élément d'un grand ensemble de petits graphes (plusieurs millions de graphes d'ordre au plus 30), filtré, avec une éventuelle agrégation des résultats. Étant donné le type de traitement et la quantité de données, il est indiqué d'exploiter la parallélisation, éventuellement sur grille de calcul. Pour l'instant, cette parallélisation est implémentée de manière ad-hoc pour les problèmes à résoudre. Le but de ce projet est d'explorer un ou plusieurs frameworks de parallélisation génériques et des bases de données relationnelles parallèles et leur mise en œuvre dans le fonctionnement du système PHOEG.

Exigences ou prérequis

Intérêt pour l'Algorithmique. Les langages utilisés par le système PHOEG sont le C++ et PostgreSQL.

26 Parallélisation de la recherche par voisinage variable

Service: Algorithmique
Directeur: H. Mélot
Rapporteurs: P. Hauweele et *personne à définir*

Description

La recherche par voisinage variable (Variable Neighborhood Search, VNS) est une méta-heuristique qui a été utilisée avec succès pour de nombreux problèmes d'optimisation. Après avoir étudié en profondeur le fonctionnement de cette métaheuristique, le but de ce projet est de proposer une adaptation de cette méta-heuristique pour permettre facilement sa parallélisation. En effet, le fait que le VNS exploite une série de voisinages permet de séparer facilement les recherches locales spécifiques à chaque voisinage.

Une implémentation de l'algorithme proposé (si possible en C++) sera réalisée par l'étudiant pour quelques problèmes d'optimisation en vue d'étudier et de valider le gain d'efficacité obtenu grâce à la parallélisation.

Exigences ou prérequis

Intérêt pour l'Algorithmique et les méta-heuristiques.

27 Création d'une application pour la gestion des gardes parentales alternées en cascade

Service: Algorithmique
Directeur: H. Mélot
Rapporteurs: *Personnes à définir*

Description

Le projet consiste à développer une application pour la gestion des gardes d'enfants alternées. En effet, beaucoup de familles recomposées sont confrontées à l'établissement d'un planning des gardes d'enfants devant être établi en cascade. Concrètement, supposons que des enfants alternent entre un foyer A et un foyer B. Les contraintes de garde du foyer A dépendent de celle des contraintes du foyer B (et inversément). Mais les contraintes du foyer B peuvent également dépendre de contraintes d'un troisième foyer C, et ainsi de suite. Ce problème est d'autant plus complexe à gérer qu'il peut exister des tensions entre les parents et qu'une succession de foyers peut dépendre de la volonté ou des demandes d'un foyer éloigné dans cette chaîne et avec qui ils ne sont pas forcément directement liés. C'est pourquoi une application simple (pour smartphone) pourrait être utilisée par les différents parents impliqués pour encoder leur contraintes (ou celles d'autres parents impliqués mais ne disposant pas de l'application) et pour calculer un planning de garde qui serait un bon compromis pour toutes les familles.

L'objectif du projet est triple. Dans un premier temps, il s'agit d'identifier les contraintes possibles et de modéliser le problème du planning sous la forme d'un problème d'optimisation où une contrainte d'un foyer sera modélisée numériquement (avec une valeur égale à 0 si la contrainte est respectée et égale à une valeur > 0 sinon) et où la fonction objectif consiste à minimiser la somme des contraintes de chaque foyer.

Dans un deuxième temps, il s'agira d'implémenter une heuristique (basée par exemple sur la Recherche par Voisinage Variable) qui a pour but de trouver rapidement une (bonne) solution au problème défini lors de la première étape.

Dans un troisième temps, l'étudiant développera un prototype de l'application pour smartphone brièvement décrite dans le premier paragraphe.

Exigences ou prérequis

Intérêt pour l'algorithmique et l'optimisation (méta-heuristiques).

28 Partial solvers for parity games: the window approach

Service: Mathématiques Effectives.
Directeur: Mickael Randour (co-directeur possible).
Rapporteurs: à définir.

Description

Parity games are a core mathematical model underlying many techniques in the formal verification and automated synthesis of provably-correct computer systems. They are two-player zero-sum turn-based games played on graphs [1]. Given a starting vertex, one of the player necessarily has a winning strategy, as such games are determined. Deciding who has a winning strategy is a canonical problem for the complexity class NP ∩ coNP and despite continuous effort, no polynomial algorithm has been found yet [2].

To provide efficient software tools despite this barrier, researchers have developed *partial solvers* for parity games: algorithms that provide correct answers but may not be complete (with regard to vertices of the graph). The goal of this project is to study such approaches (e.g., [3, 4]) and assess the applicability of *window parity games* [5] as a new mechanism for partial solvers.

Exigences ou prérequis

Basic notions of algorithmics and programming.

References

[1] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, eds. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*. Vol. 2500. Lecture Notes in Computer Science. Springer, 2002. ISBN: 3-540-00388-6. DOI: 10.1007/3-540-36387-4. URL: https://doi.org/10.1007/3-540-36387-4.

[2] Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. "Deciding parity games in quasipolynomial time". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*. Ed. by Hamed Hatami, Pierre McKenzie, and Valerie King. ACM, 2017, pp. 252–263. ISBN: 978-1-4503-4528-6. DOI: 10.1145/3055399.3055409. URL: http://doi.acm.org/10.1145/3055399.3055409.

[3] Steen Vester. "Winning Cores in Parity Games". In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*. Ed. by Martin Grohe, Eric Koskinen, and Natarajan Shankar. ACM, 2016, pp. 662–671. ISBN: 978-1-4503-4391-6. DOI: 10.1145/2933575.2933589. URL: http://doi.acm.org/10.1145/2933575.2933589.

[4] Michael Huth, Jim Huan-Pu Kuo, and Nir Piterman. "Static Analysis of Parity Games: Alternating Reachability Under Parity". In: *Semantics, Logics, and Calculi - Essays Dedicated to Hanne Riis Nielson and Flemming Nielson on the Occasion of Their 60th Birthdays*. Ed. by Christian W. Probst, Chris Hankin, and René Rydhof Hansen. Vol. 9560. Lecture Notes in Computer Science. Springer, 2016, pp. 159–177. ISBN: 978-3-319-27809-4. DOI: 10.1007/978-3-319-27810-0_8. URL: https://doi.org/10.1007/978-3-319-27810-0_8.

[5] Véronique Bruyère, Quentin Hautem, and Mickael Randour. "Window parity games: an alternative approach toward parity games with time bounds". In: *Proceedings of the Seventh International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2016, Catania, Italy, 14-16 September 2016*. Ed. by Domenico Cantone and Giorgio Delzanno. Vol. 226. EPTCS. 2016, pp. 135–148. DOI: 10.4204/EPTCS.226.10. URL: https://doi.org/10.4204/EPTCS.226.10.

29 Machine learning for strategy synthesis in Markov decision processes

Service: Mathématiques Effectives.
Directeur: Mickael Randour (co-directeur possible).
Rapporteurs: à définir.

Description

Markov decision processes (MDPs) are widely used to reason about decision-making in stochastic environments (e.g., journey planning [1]). Efficient algorithms exist to compute optimal control strategies, and have been implemented in successful tools such as Storm (http://www.stormchecker.org/) or PRISM (https://www.prismmodelchecker.org/). Still, they fail when considering very large MDPs that arise naturally in practice.

Recently, approaches have been developed to combine machine learning and formal methods in a way that permits to synthesize strategies with provable guarantees with largely increased efficiency (e.g., [2]). The goal of this project is to study such methods and to try to extend them to *multi-objective* models.

Exigences ou prérequis

Basic notions of probabilities, algorithmics, and programming.

References

[1] Mickael Randour, Jean-François Raskin, and Ocan Sankur. "Variations on the Stochastic Shortest Path Problem". In: *Verification, Model Checking, and Abstract Interpretation - 16th International Conference, VMCAI 2015, Mumbai, India, January 12-14, 2015. Proceedings*. Ed. by Deepak D'Souza, Akash Lal, and Kim Guldstrand Larsen. Vol. 8931. Lecture Notes in Computer Science. Springer, 2015, pp. 1–18. ISBN: 978-3-662-46080-1. DOI: 10.1007/978-3-662-46081-8_1. URL: https://doi.org/10.1007/978-3-662-46081-8_1.

[2] Tomas Brazdil, Krishnendu Chatterjee, Martin Chmelik, Vojtech Forejt, Jan Kretinsky, Marta Z. Kwiatkowska, David Parker, and Mateusz Ujma. "Verification of Markov Decision Processes Using Learning Algorithms". In: *Automated Technology for Verification and Analysis - 12th International Symposium, ATVA 2014, Sydney, NSW, Australia, November 3-7, 2014, Proceedings*. Ed. by Franck Cassez and Jean-François Raskin. Vol. 8837. Lecture Notes in Computer Science. Springer, 2014, pp. 98–114. ISBN: 978-3-319-11935-9. DOI: 10.1007/978-3-319-11936-6_8. URL: https://doi.org/10.1007/978-3-319-11936-6_8.

30 Correctness in AI through formal methods

Service: Mathématiques Effectives.
Directeur: Mickael Randour (co-directeur possible).
Rapporteurs: à définir.

Description

The ever-increasing resort to learning and AI in our lives has led to concerns regarding their safety. Obviously, researchers in the field did not wait for formal methods to reason about the correctness of learners. Core concepts such as **probably approximately correct (PAC) learning** have been around for decades [1]. Their overall objective is to guarantee that with high probability, the approximation granted by the learner will converge to a solution close to the optimum.

As formal methods get closer to learning, we hope to go further and provide strong guarantees (e.g., worst-case) on learning models and AI frameworks, notably based on concepts of probabilistic model checking. The goal of this project is to establish a comparative overview of correctness measures in AI and formal methods and to draw the first lines of an alliance between both fields.

Exigences ou prérequis

Basic notions of probabilities, algorithmics, and programming.

References

[1] Leslie G. Valiant. "A Theory of the Learnable". In: *Commun. ACM* 27.11 (1984), pp. 1134–1142. DOI: 10.1145/1968.1972. URL: http://doi.acm.org/10.1145/1968.1972.

31 Acquisition et traitement sonore d'une guitare électrique sous Linux

Service: Informatique Théorique
Directeur: O. Delgrange
Rapporteurs: *à définir*
Mots clefs: Linux, guitare électrique, temps réel, PureData, traitement de signal

Description

Le projet consiste à étudier les environnements logiciels Linux capables d'acquérir et de traiter, en temps réel, le son provenant d'une guitare électrique. Dans un premier temps, l'étudiant s'intéressera aux différents protocoles sonores de Linux, aux logiciels open source d'acquisition sonore et en fera une étude comparative.

Ensuite, il conviendra de s'intéresser au traitement en temps réel des séquences provenant de la guitare. Pour cela, le logiciel *PureData* (langage de programmation visuel) sera étudié et utilisé. Ce dernier permet de connecter différents composants sonores logiciels (filtres, ...) en fonction des observations ou transformations que l'on veut opérer sur le signal.

À titre d'exemple, des applications temps réel seront développées. Selon les possibilités, il pourra s'agir d'un accordeur (éventuellement polyphonique), d'un programme de reconnaissance d'accords, d'effets guitare, ...

Exigences ou prérequis

Avoir un intérêt pour le traitement du signal. Maîtriser Linux. Disposer d'une guitare électrique et posséder des connaissances rudimentaires d'utilisation de la guitare.

32 Étude des systèmes de fichiers pour mémoires flash

Service: Informatique Théorique
Directeur: O. Delgrange
Rapporteurs: *à définir*
Mots clefs: Mémoires flash, SSD, TRIM, JFFS2, YAFFS, MTD

Description

Les mémoires flash (disques SSD, ...) sont des périphériques électroniques permettant de stocker des données de manière permanente. Elles sont donc largement utilisées en informatique en lieu et place des disques durs magnétiques et des disquettes. Elles sont divisées en *blocs* de tailles fixes qui peuvent être accédés en une fois, tout comme les disques durs, mais sans aucune pénalisation de "déplacement de têtes de lecture/écriture". Ces mémoires sont moins encombrantes, et consomment moins d'énergie, elles sont donc souvent préférées pour les ordinateurs portables et les systèmes embarqués. Par contre, de nos jours, elles sont beaucoup plus onéreuses que les disques magnétiques.

Le projet consiste en l'étude et l'expérimentation des systèmes de fichiers utilisant les mémoires flash.

Une contrainte technique majeure régit le fonctionnement des mémoires flash : avant d'écrire une *page* (chaque bloc est composé de pages elles mêmes de tailles fixes), il faut qu'elle ait été "effacée". Toutefois, une page ne peut pas effacée seule, l'effacement doit porter sur tout le bloc. Le processus d'écriture est donc assez complexe et peut diminuer les performance du périphérique. De plus, les blocs possèdent un nombre de cycles d'effacement/écriture limité, ils ont donc une "durée de vie" limitée. Les systèmes de fichiers prévus pour les disques magnétiques ne sont donc pas adaptés aux mémoires flash car il convient de répartir uniformément les effacements/écritures pour éviter une sur utilisation de certains blocs et une sous utilisation d'autres.

Il conviendra, pour démarrer le projet, de faire un état de l'art concernant les technologies et les algorithmes, les limitations et les systèmes de fichiers pour mémoires flash. Différents systèmes seront présentés, concernant différents systèmes d'exploitation.

Des expérimentations seront ensuite menées en Linux.

Exigences ou prérequis

Un intérêt pour les systèmes d'exploitation.

Posséder un ordinateur installé (ou installable) sous Linux avec un disque SSD dont le contrôleur est compatible avec la spécification "TRIM" et sur lequel pourront être menées des expérimentations. Pour le vérifier, utiliser la commande linux
`sudo hdparm -I /dev/sdX | grep TRIM`, où `/dev/sdX` désigne votre disque SSD.