# 2022 Center for Undergraduate Research and Scholarship (CURS) Summer Scholar Program

### CURS 4990 Syllabus - Summer 2022

## Presentation

This class was funded and organized by the Center for Undergraduate Research & Scholarship at Augusta University, and allowed Peter Browning and myself to work on the following project for 5 weeks.

Find below the syllabus and a brief presentation of the outcomes of this class.

## Outcomes

The three most important outcomes of this class were:

- The actual code, accessible at https://github.com/peterbro1/IRDC-CCSK,
- The poster, presented below, that can be downloaded as a pdf or as a pptx,
- The presentation, made by Peter during the SSP Symposium.

## Syllabus

### Research Project Presentation

**Overview.** Many of today's computers' benefits are enabled by connecting them to networks, where they exchange data (messages, emails, requests, …) and programs (through updates). But allowing distributed computation requires to handle concurrent phenomenon: how to make sure that no error arise when multiple clients try to book the same ticket, or when conflicting updates are pushed at the same time? By mathematically abstracting concurrent behaviors, Computer Science made considerable progress in preventing conflicts or inconsistencies that could arise in such situations.

But new techniques and paradigms of computation emerge constantly and require to revisit established results continuously. Reversibility is one of those new paradigms: the possibility of undoing any action performed by a computer makes the promise of more secure and energy-efficient computers where bugs become easier to find and address. The field of reversible computing—that includes quantum computing—will shape an exciting future, and as such is the target of consequent effort and among NSF's 2026 Idea Machine [1].

However, no definitive mathematical model of reversible *and* concurrent computation exists, and the existing attempts have not been implemented in computers: we offer to design a novel model surpassing the shortcomings of the first models, and to implement it so that it can become an experimental tools for researcher in our community.

**Context.** Process algebras ($\pi$-calculus, CCS, Ambiant calculus, etc.) are an abstraction of concurrent systems useful to study, specify and verify distributed behaviors. Implementing process calculi serves three overlapping goals:

- It allows to machine-check theorems and definitions [2,3] using proof assistants such as Coq [4], resulting sometimes in simplications [3] or the finding of regrettable imprecisions and errors [5].
- Using it as an actual programming language, it enables the implementation of toy programs [6] that exemplifies the purpose and expressivity of the calculus.
- It can also be used as a specification language: typically, the Proverif tool [7], which implements the applied $\pi$-calculus [8], has been used to certify and model security protocols in a variety of areas [9].

The CCS language undergoes two different efforts making it amenable to represent computation that can move backward and forward: Reversible CCS (RCCS) [10] and CCS with keys (CCSK) [11] were both developed with the goal of becoming *the* extension to CCS providing a better understanding of the mechanisms underlying reversible concurrent computation—they actually turned out to be the two faces of the same coin [12]. Reversible computation in general has received a lot of attention from different communities [13], and the study of reversible process calculi has made important progresses in the recent years [13, Sect. 6]. However, aside from SimCCSK [14]—which is not publicly available and not maintained since 2008—no implementation of concurrent, reversible CCS exists.

## Goals

### Educational

The main educational goal of the project is to **give to Peter the taste of a scholarly culture, of high-quality code production, and to stress the relevance and importance of undergraduate research**. More precisely, our goals are to:

- Foster Peter's independence, to sharpen his capacities to solve problems by himself.
- Improve his self-efficacy, to believe in his capacities but also to identify when and how to ask for help.
- Encourage his curiosity, to drive his own intellectual journey.
- Leverage the CURS workshops to have a better understanding of the multiple facets of high-quality undergraduate research, and to benefit from a good professional development.
- Introduce to the administrative aspects of research[1] by e.g. searching for good-quality journals and venues to submit our work, or funds to sparkle other projects.
- Become confident with technologies (git, LaTeX, references managers, markdown, …) common in research.

### Research

Peter will work toward an implementation of either CCSK, RCCS, or a declension of them [15], possibly taking inspiration of existing implementation of forward-only CCS (among wich this project or this web-interface [16]) or of intermediate languages such as HOcore [5]. A great care will be required toward good software engineering practices, the development of good examples, the possible certification of some results using machine-checked proof, and / or the implementation of an efficient mechanism to distribute the generation of keys and identifiers.

Our main goals, aside from the educational goals detailed above, are to produce a research document and a computer program, documented and with examples, that will be shared with experts in the field through self-archiving on the arXiv.org and github repositories, and a poster for the Symposium that will get re-used

[1] Something that already started, since Peter had a chance to review the funded proposal and this syllabus.

in other venues and will be self-archived as well. All the material produced will be released under Creative Commons licenses (or similar open-source licence), to ease distribution and re-use by the community.

A regularly updated document and journal will help in completing those milestones, and will constitute a way of assessing of the progresses of the project as well as of the clarity of our understanding of the project's developments.

### Outcomes

Our goal is to have three tangible traces of our investigation by the end of the program:

- A poster, to be shared at the CURS final symposium, and hopefully re-used in other venues,
- A program along with multiple examples and its documentation,
- A self-archived manuscript, that hopefully will be submitted to an open-access (undergraduate) journal.

## Team & Participant Roles

Will be involved in this project (in alphabetical order):

- Dr. Clément Aubert, Augusta University, caubert@augusta.edu (mentor),
- Peter Browning, Augusta University, pebrowning@augusta.edu (mentee).

### Peter Browning's Roles

Peter Browning already made his debuts in research in Computer Science, and starts to have some familiarity with how to conduct research in this field. For this project, his roles will include:

- Being the lead on the development and testing of our program(s), documentation and examples.
- Being the lead on the writing and editing of a document that summarizes our findings, as well as on the Symposium poster.
- Reading and understanding important (excerpts of) papers in the literature.
- Being able to re-formulate in his own term the research project we are following, to write a clear introduction to our document.
- Write a daily entry in his research journal (possibly as a detailled git commit message log).
- Reporting regularly to Dr. Aubert, and articulate clearly what has been achieved, what needs to be done, and where help is needed.

### Dr. Aubert's Roles

For this project, his roles will include:

- Providinig guidance and impromptu clarifications whenever needed, to develop Peter's skills and understanding, and to encourage interactions by letting him drive our discussions.
- Valuing Peter's skills and benefiting from his rigor and intuition to strengthen the project.
- Proposing short-terms goals that support and realize long-term plan and achievements.
- Giving to Peter as much independence and liberty as wished, in conjunction with a constantly available support, while respecting his individuality.
- Easing Peter into formulating his own hypothesis, testing them, and guiding the future progresses of the project.

- Helping Peter formulating the research problem in his own term, and encouraging him to present his understanding of the project in public venues or through publication(s).

More generally, Dr. Aubert will always make sure that his expectations are clearly communicated and understood, and convey through regular meetings and email exchanges his intuitions, solutions, and suggestions to support the project's progress and our educational goals.

## Timeline

The program will start on May 23 (week 1) and ends on June 24 (week 5).

### Research Timeline

| Week | Main Activities |
|---|---|
| 1 | Review of existing literature and implementations of forward-only CCS |
| 2 | Write the specification of the implementation, in terms of functionalities and features |
| 3 | Set-up the overall program following the architecture agreed upon |
| 4 | Implement, implement, implement |
| 5 | Document, beta-test and write examples |

A regularly updated to-do list will support and clarify the steps needed to achieve those overall goals. The poster is due on July 18th.

### Training Timeline

Peter will attend the orientation, training, workshop luncheons, and symposium as follows:

| Day | Time | Event |
|---|---|---|
| Wed. May 18 | 15:00 – 16:30 | Student orientation |
| Wed. May 25 | 12:00 – 12:45 | Lunch-n-learn: Collaborations |
| Wed. June 1 | 12:00 – 13:30 | Lunch-n-learn: Ethics |
| Wed. June 8 | 12:00 – 12:50 | Lunch-n-learn: Presenting 101 |
| Wed. June 15 | 12:00 – 13:30 | Lunch-n-learn: Professional Communications |
| Wed. June 22 | 12:00 - 13:30 | Lunch-n-learn: Data visualization |
| Wed. June 29 | 12:00 - 13:30 | Lunch-n-learn: Build your research pitch |
| Wed. July 13 | 12:00 - 13:30 | Lunch-n-learn: Research in your Resume, Publications Opportunities |
| Thur. July 21 | 16:00 – 18:00 | SSP Symposium |

He will be excused from the workshop after June 24 if he wishes.

### Weekly Organization

- Every week, Dr. Aubert and Peter Browning will meet on Monday and Thursday at 10:00, and by need, on teams.
- Every day, Peter will complete his journal, by indicating what he accomplished, where he stands, and what he plans on doing for the next day. This will help
  - Him, to clarify his thoughts,
  - His mentor, to grasp where he is,
  - Him (again!), in case he needs to come back to them,
  - The CURS, to ease final check-in.

It is recommended that Peter works at least 2 hours / working day for the duration of the program, and spend at least 5 minutes after each work session documenting what he achieved.

## Tools

Will be used during this program, among other resources:

- References and our document will be shared on https://github.com/aubertc/Implementing_Reversible_Process_Algebras,
- Our program will be developed on https://github.com/CinRC/IRDC-CCSK,
- teams will be our main method of "live" communication,
- Email will be our main method of asynchronous communication.

# Miscellaneous

- Reservation of rights: I reserve the right to change this syllabus without limitation and without prior notice. If I do substantially modify any item or policy, I will notify you during a lecture, or send an e-mail to your `augusta.edu` e-mail account.
- Download a `pdf` version of this page.
- Contact: caubert@augusta.edu
- Created with debian, pandoc and latex.
-  Sources are available upon motivated request.
- You will need a `pdf` reader to consult some of the documents: I recommend choosing an open-source `pdf` reader.

# References

[1] Reversibility: Future of Life on Earth | The NSF 2026 Idea Machine, (2018). https://nsf2026imgallery.skild.com/entries/reversibility-future-of-life-on-earth (accessed October 9, 2021).

[2] D. Hirschkoff, A full formalisation of pi-calculus theory in the calculus of constructions, in: E.L. Gunter, A.P. Felty (Eds.), Springer, 1997: pp. 153–169. https://doi.org/10.1007/BFb0028392.

[3] J. Despeyroux, A higher-order specification of the pi-calculus, in: J. van Leeuwen, O. Watanabe, M. Hagiya, P.D. Mosses, T. Ito (Eds.), Theoretical Computer Science, Exploring New Frontiers of Theoretical Informatics, International Conference IFIP TCS 2000, Sendai, Japan, August 17-19, 2000, Proceedings, Springer, 2000: pp. 425–439. https://doi.org/10.1007/3-540-44929-9_30.

[4] T.C.D. Team, The coq proof assistant, version 8.7.0, (2017). https://doi.org/10.5281/zenodo.1028037.

[5] P. Maksimovic, A. Schmitt, HOCore in coq, in: C. Urban, X. Zhang (Eds.), Interactive Theorem Proving - 6th International Conference, ITP 2015, Nanjing, China, August 24-27, 2015, Proceedings, Springer, 2015: pp. 278–293. https://doi.org/10.1007/978-3-319-22102-1_19.

[6] R. Affeldt, N. Kobayashi, A coq library for verification of concurrent programs, Electronic Notes in Theoretical Computer Science. 199 (2008) 17–32. https://doi.org/10.1016/j.entcs.2007.11.010.

[7] B. Blanchet, Modeling and verifying security protocols with the applied pi calculus and ProVerif, Foundations and Trends in Privacy and Security. 1 (2016) 1–135. https://doi.org/10.1561/3300000004.

[8] M. Abadi, B. Blanchet, C. Fournet, The applied pi calculus: Mobile values, new names, and secure communication, Journal of the ACM. 65 (2018) 1:1–1:41. https://doi.org/10.1145/3127586.

[9] M.D. Ryan, B. Smyth, Applied pi calculus, in: V. Cortier, S. Kremer (Eds.), Formal Models and Techniques for Analyzing Security Protocols, IOS Press, 2011: pp. 112–142. https://doi.org/10.3233/978-1-60750-714-7-112.

[10] V. Danos, J. Krivine, Reversible communicating systems, in: P. Gardner, N. Yoshida (Eds.), CONCUR 2004 - Concurrency Theory, 15th International Conference, London, UK, August 31 - September 3, 2004, Proceedings, Springer, 2004: pp. 292–307. https://doi.org/10.1007/978-3-540-28644-8_19.

[11] I. Phillips, I. Ulidowski, Reversing algebraic process calculi, in: L. Aceto, A. Ingólfsdóttir (Eds.), Foundations of Software Science and Computation Structures, 9th International Conference, FOSSACS 2006, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2006, Vienna, Austria, March 25-31, 2006, Proceedings, Springer, 2006: pp. 246–260. https://doi.org/10.1007/11690634_17.

[12] I. Lanese, D. Medić, C.A. Mezzina, Static versus dynamic reversibility in CCS, Acta Informatica. (2019). https://doi.org/10.1007/s00236-019-00346-6.

[13] B. Aman, G. Ciobanu, R. Glück, R. Kaarsgaard, J. Kari, M. Kutrib, I. Lanese, C.A. Mezzina, L. Mikulski, R. Nagarajan, I.C.C. Phillips, G.M. Pinna, L. Prigioniero, I. Ulidowski, G. Vidal, Foundations of reversible computation, in: I. Ulidowski, I. Lanese, U.P. Schultz, C. Ferreira (Eds.), Reversible Computation: Extending Horizons of Computing - Selected Results of the COST Action IC1405, Springer, 2020: pp. 1–40. https://doi.org/10.1007/978-3-030-47361-7_1.

[14] G. Cox, SimCCSK: simulation of the reversible process calculi CCSK, Master's thesis, University of Leicester, 2010. https://leicester.figshare.com/articles/thesis/ SimCCSK_simulation_of_the_reversible_process_calculi_CCSK/ 10091681.

[15] C. Aubert, D. Medić, Explicit identifiers and contexts in reversible concurrent calculus, in: S. Yamashita, T. Yokoyama (Eds.), Reversible Computation - 13th International Conference, RC 2021, Virtual Event, July 7-8, 2021, Proceedings, Springer, 2021: pp. 144–162. https://doi.org/10.1007/978-3-030-79837-6_9.

[16] J.-F. Gillet, D. Willame, Calculus of Communicating Systems: A web based tool in Scala, Master's thesis, Université de Namur, 2017. https://researchportal.unamur.be/files/30127909/ GILLET_WILLAME_Memoire.pdf.