POSTGRADUATE PROGRAMMES IN

Computer Science

www.cs.le.ac.uk
Contents

Welcome from the Head of Department and the MSc Director 3

Why study for an MSc in Computer Science at Leicester? 4
  Entry Requirements 5
  Applications 5

Research in the Department 6

A Selection of Recent MSc Projects 8

Living at Leicester 10

Content and Structure 12

Our Programmes
  MSc in Advanced Computer Science 14
  MSc in Advanced Computational Methods 15
  MSc in Advanced Distributed Systems 16
  MSc in Advanced Software Engineering 17
  MSc in Agile Software Engineering Techniques 18
  MSc in Software Engineering for Financial Services 20
  MSc in Web Applications and Services 22

How to apply

You have the option of applying by post or online. For detailed instructions and additional information please visit www.le.ac.uk/graduateoffice/pgprospectus

Our staff are happy to advise if you have any queries about your application. Our contact details can be found overleaf.
Welcome from the Head of Department and the MSc Director

We are delighted to be able to offer you a variety of advanced taught postgraduate programmes in Computer Science. The programmes on offer are all aligned with the needs of industry and take full advantage of our research expertise. This means that you will be learning about technologies and techniques that are at the cutting edge of current knowledge and at the forefront of industrial practice.

Our colleagues from industry participate directly in the delivery of some of the modules, including hands-on tutorials that allow you to use some of the tools and methodologies that are shaping the software development landscape.

You will find in the Department many other students studying for a PhD degree, postdoctoral researchers working on several national and international projects, and international visitors from industry and academia; this creates a dynamic and exciting environment to study and develop your own individual project.

A number of workshops will further ensure that you will leave Leicester well-equipped with personal and group skills that employers value very highly. We also provide excellent academic and personal support along with comprehensive careers guidance that includes a personal development plan.

If you need more information after reading this brochure please do not hesitate to contact our MSc admissions team. Our Department is a lively and friendly place; we take pride on the quality of our staff and support that we provide to our students. We hope that you will find the right programme for you and we look forward to welcoming you to Leicester.

José Fiadeiro  
Head of Department

Stephan Reiff-Marganiec  
Director of MSc Programmes
Why study for an MSc in Computer Science at Leicester?

An MSc allows you to specialise in your chosen area of Computer Science. You will gain specific skills at the interface between research and practice, and at the forefront of industrial needs.

**Employability**

Having an MSc increases employability. Most of our MSc graduates have been hired by companies either in Britain or abroad within six months of graduating.

**Tailored Courses**

We offer a rich portfolio of advanced courses and a flexible degree structure that allows you to choose a programme of studies that matches an area in which you want to specialise, a profile suitable for the career that you wish to pursue, or simply a programme satisfies your curiosity and interests.

**Skills Development**

Our courses allow you to develop into a highly skilled professional, well-versed in advanced methods and tools. We will stimulate your creativity and capacity to innovate, and ensure that you acquire key transferable skills.

**Research-led Teaching**

All our academic staff are hired on the basis of their international research excellence, as well as their teaching skills. We are involved in several national and international projects and networks, and we work together with top researchers across the world to extend the boundaries of knowledge in computing. Altogether, we provide a stimulating research environment in which students can frame their projects and plan their future careers.

**Industry Collaboration**

We have an open approach to collaboration with industry. We undertake joint projects, provide consultancy services, and organise events to connect you with potential employers. We also run an Industrial Advisory Board: its members are computer scientists who hold successful careers in industry and who help us ensure that your training is industrially relevant.
Entry Requirements

Candidates should have, or expect to gain, at least a good second class honours BSc degree or a qualification of equivalent standard recognised by the University in a subject with a substantial element of Computing. Applications are treated on an individual basis, however, and so alternative qualifications may be considered, especially in the case of candidates with relevant work experience.

Applications

Applications are handled by the Graduate Office. The University accepts applications online or in paper form.

If you choose the online application process, your application form will be with us very quickly; we then await the supporting material (transcripts, references, etc) before making a final decision. If you want to apply online, visit www.le.ac.uk/graduateoffice.

If you prefer to apply through the post, you can download the application form from the Graduate Office website at www.le.ac.uk/graduateoffice. The form is a pdf file and requires Acrobat Reader for viewing and printing.

If you feel that you would rather have some printed information sent to you by post, send a message to the MSc admissions team (csmsc@mcs.le.ac.uk) with your request. We will also be happy to answer any queries that you may have.

The University has specific information available for International and EU students at www.le.ac.uk/international.

“The city is a lovely multicultural place and this is reflected in the class where many of my fellow students are from around the globe and I really feel this adds to the learning experience.”

Walter Magulike

“Doing this degree is an excellent cross-over point between undergraduate study and working in industry.”

Stephen Gorton

“I have some experience in the IT industry and I found that the course contents are very practical and up to date with the latest technologies and methodologies.”

Bindu Siju
Research in the Department

We are a young and dynamic department that takes a scientific approach to Computing. This is evident in our research where we have people working at the cutting edge of the subject. You will benefit from the fact that researchers at Leicester are playing a major part in developing modern trends in Computer Science; this is reflected in our teaching where we have a strong tradition of high-level and innovative courses.

Our research profile can be characterised by the eight broad themes listed overleaf. Staff and students are usually involved in a number of research grants including projects with other universities and companies as well as other collaborations at national and international level. Funded research positions are often made available for working in specific projects.

Some students on our MSc programmes decide to study for a PhD. You can also apply directly for a research degree (MPhil/PhD). If you are interested in one of our research areas you can directly email csphd@mcs.le.ac.uk.
Algebraic and Categorical Structures and Methods

**TOPICS:** Algebras, coalgebras and categorical structures as mathematical objects, including (co)inductive methods and modal logics. Applications in computer science: Algebraic and coalgebraic specification; Calculi and models of concurrent, distributed, mobile, and context-aware computing; General systems theory and computational models (chemical, biological, etc); Semantics of conceptual modelling methods and techniques; Semantics of programming languages; Validation and verification.

Algorithm Design, Analysis and Engineering

**TOPICS:** Data structures, hierarchical-memory algorithms, approximation algorithms, algorithms for dealing with uncertain data, including online algorithms. Applications to: optimising performance and energy consumption in optical and ad-hoc networks, railroad optimisation, bioinformatics, text indexing, representing semi-structured data, network analysis.

Computational Complexity of Algebraic Structures


Deduction, Rewriting and Transformation

**TOPICS:** Graph transformations; Term rewriting; Theorem proving; Stochastic approaches. Rewriting of biostructures. Operational semantics of concurrency and programming languages.

Interaction Design and Evaluation of Socio-technical Systems

**TOPICS:** Usability and user experience (UX) evaluation methodologies for emergent interactive technologies; Interaction design of computer-supported collaborative work/learning (CSCW/L) environments; Digital educational games; Interoperability issues of social software; Evolution of cross-cultural online communities; Adaptive mash-up services for personal responsive learning environment; Social network analysis of computer-mediated interactions; Creativity and software quality models; Trans-sectorial transfer of design and evaluation methods and tools; Web2.0-enhanced requirement engineering; Engineering of complex socio-technical services: formal modelling and analysis, orchestration and service-level agreements, security and trust; Interaction and context-based technologies for collaborative teams.

Models of Software Intensive Systems

**TOPICS:** Concepts, languages, and semantic models for large and complex software intensive systems. In particular: Architectural dimensions (coordination, distribution, context-awareness); Collaborative systems; Embedded, hybrid, and timed systems; Formal specification, validation and verification techniques; Service-oriented systems. Composition and emergent behaviour/properties.

Optimisation and Heuristics

**TOPICS:** Combinatorial optimisation; scheduling; resource allocation; routing; network design and optimisation; approximation algorithms; meta-heuristics; competent genetic and evolutionary algorithms; evolutionary algorithms for dynamic optimisation; artificial neural networks for optimisation; hybrid intelligent systems; online algorithms and competitive analysis.

Software Evolution

**TOPICS:** Meta-modelling; Model-driven architecture/development; Re-engineering of legacy systems; Aspect discovery and refactoring; Software process engineering; System families. Incremental development; Features and Aspects: policy-driven, ad-hoc composition.
A Selection of Recent MSc Projects

Robust and Energy Efficient Wireless Sensor Networks

Wireless sensor networks are a new technology with a wide range of applications (from environment monitoring to surveillance) but subject to a number of energy constraints, many-to-one flows, and redundant low-rate data. Many routing protocols have been proposed for sensor networks that aim at achieving energy efficiency, but with little or no attention to robustness and fault-tolerance. This project addressed fault-tolerance in ad-hoc sensor network routing, namely the design and development of a robust and energy efficient routing protocol that distinguishes between local and large-scale traffic. A new multipath routing protocol was proposed and simulated which follows the “number of hops” metric and employs a waiting time before transmitting messages to sink.

WSDL/BPEL to SRML Language Transformation

SRML is the Reference Modelling Language of the European project SENSORIA. It provides primitives for modelling and reasoning about business processes in a way that is independent of the languages used for executing them. This project developed a tool for generating SRML models from two specific industry standards: The Business Process Execution Language (BPEL), which is used for specifying business process based on Web Services, and the Web Services Description Languages (WSDL) – an XML format for describing network services. The encoding provides the means to create high-level declarative descriptions of BPEL/WSDL specifications that can be used for building more complex business process models that can include components specified in other languages.

XML Database for Bibliographic Information

Many people in industry and academia use a document typesetting system called LaTeX. Documents may be published in various formats including reports, journal articles, conference proceedings, authored books, and so on. BibTeX is the tool used by LaTeX for formatting lists of references that are to be incorporated within these documents. Although it is very effective, BibTeX has raised a practical concern due to its low-level of abstraction, making it difficult to share bibliographic information among tools or people executing different activities (for instance, between the technical and the marketing staff within a company). In order to address this concern, this project developed an XML-based framework for managing bibliographic information that imports/exports BibTeX references.

“The labs and facilities here are fantastic, it’s a really good place to be a postgraduate student. The course is also in exactly the field I’m interested in.

As well as the support from the Department, the International Office and the Welcome Programme for International Students are fantastic. You get to meet lots of new people from all over the world and it’s a really good way to introduce you to the University.

The city is really multicultural, it’s like a home from home! You can get most of the food and things you can get at home and the culture feels quite similar. The cost of living in the city is also good so your money goes a bit further!

Once I finish the course I hope to start a career in software engineering and spend a couple of years building my skills in the UK.”

Vinod Thota
After completing my BSc at Leeds, I worked at the Alliance and Leicester for around seven years, before choosing to come back and study at Master’s level at Leicester.

The course has been really good. I’d never really done programming before and the course has been really helpful in developing my skills. I’ve done some work on website development and Java programming.

The lecturers have all been extremely helpful and friendly. You can go to them with any queries and they are always happy to help you with anything you don’t understand.

Socially the University is really geared to the needs of international students and we often go out as a course group, when we can find time with all of our studying!

When I finish the course I plan to go back home to Malaysia for a couple of months and then move into a career in web development.

I’ve really enjoyed the course so far and would recommend it to anyone considering studying here.

Nicole Yap
Living at Leicester

About the City of Leicester

Leicester is a lively, multi-cultural city. Its population is nearly 300,000 making it the tenth largest city in England. Leicester has a huge choice of pubs, clubs, restaurants, cinemas and theatres as well as excellent shopping – from one of Europe’s oldest markets to the new Highcross Leicester development which includes a 12 screen cinema, restaurants, cafes and apartments.

Enjoy top class sport at the Leicester Tigers ground at Welford Road (rugby union), at the Walkers Stadium (football) and at Grace Road (cricket).

Take a trip to another galaxy without leaving the planet at the National Space Centre. Get in amongst the buzz of the lively, covered market that has a 700-year history. See one of the largest pieces of Roman architecture in Britain at the Jewry Wall Museum and travel back to the Middle Ages at Leicester’s haunted medieval Guildhall. See contemporary works and a display of printmaking techniques at the City Gallery. Visit the natural history museum at New Walk, and also see the largest collection of German Expressionist art in the country. The shimmering saris and fabulous silks found along Belgrave Road are works of art in themselves.

The range of cafés and restaurants in Leicester is extensive; but Leicester’s greatest culinary strengths is South Asian food – exciting thalis, Indian sweets and delicious fresh samosas are just some of the cuisine on offer. For something altogether more lavish, dine in style on a steam train at the Great Central Railway. As a popular film location, this is a magnet for Hollywood stars including Nicole Kidman and Kate Winslet.

Leicester is in the heart of England with excellent communications by road, air and rail.

As Britain’s first Environment City, Leicester is engaged in a programme of sustainable development to make it a green and pleasant place. It also enjoys some fine countryside within easy reach of the City.

About the University

Education that inspires. Research that changes the world.

The University of Leicester is a leading UK University with a proud past and an exciting future. We deliver high quality undergraduate, postgraduate and professional education and create research that has impact internationally.

Our research changes the world. According to Thomson Scientific, Leicester has the tenth highest number of highly cited researchers amongst the UK’s universities. The discovery of DNA genetic fingerprinting is our most famous research achievement, but our world class research stretches across the arts, science and engineering, medicine, law, education, biological sciences and social sciences.

The University has the joint highest scores for overall student satisfaction in England amongst mainstream universities (National Student Survey). Leicester’s student completion rate is amongst the very highest.
Content and Structure

We offer you seven different programmes of study and, within each programme, a number of options that you can choose from in order to match your preferred profile. Each of the programmes will equip you for high-profile careers in the IT sector: three of the courses provide advanced training in key areas; another three address topical areas of software engineering; and we also offer you the possibility of creating your own path. You can find more details of the programmes on the next few pages and also on our website. The descriptions for the topical SE programmes is more detailed as these are quite specialised.

Programmes

- Advanced Computer Science (Course code ACS)
- Advanced Computational Methods (Course code ACM)
- Advanced Distributed Systems (Course code ADS)
- Advanced Software Engineering (Course code ASE)
- Agile Software Engineering Techniques (Course code SET)
- Software Engineering for Financial Services (Course code SFS)
- Web Applications and Services (Course code WAS)

Progression

MSc and Postgraduate Diplomas are offered for all courses and require 120 credits corresponding to Personal and Group Skills and taught modules. In addition MSc degrees require an individual project as explained below. Postgraduate Certificates require 60 credits of taught modules and are offered for Advanced Computer Science only.

Personal and Group Skills

This compulsory module provides you with skills that are valued by all IT employers: critical analysis, appraisal of evidence, communication, working relationships, managing learning, and autonomous research. The module features seminars by guest speakers, moderated group discussions, collective writing, workshops on transferable skills and career planning. You will also develop your own personal development plan.

Taught Modules

Each course except ACS requires four or five core taught modules, marked in yellow, and two or three options chosen from the whole menu and a list of supplementary modules updated every year (subject to timetabling constraints). Students on the ACS course can compose their own pathway.

Individual Project

If you wish (and qualify) to obtain the full MSc qualification, you will develop an individual project under
the supervision of one of our academic staff members. You will work full-time after your exams and deliver a dissertation on a topic selected according to the profile of your chosen course. The project is expected to contain some element of original work, and may involve informal collaboration with other organisations, subject to the previous approval of the project supervisor. You can find some examples on page 8.

Admissions

Admissions are in October and January.

Students admitted in October follow Semester 1 modules until December. In January, after examinations, they follow Semester 2 modules. Examinations are taken again in May. Students who wish and are permitted to proceed undertake their individual project during the summer and submit their dissertation by the middle of September.

Students admitted in January start by following Semester 2 modules. There are no module dependencies between the two semesters. Examinations are taken in May. Students then break during the summer period and resume in October for Semester 1 modules. Examinations are taken in January after which those who wish and are permitted to proceed start their project. Dissertations are then submitted in the middle of May.

Transfer and Progression

Students may transfer from any of the specialised courses (ACM, ADS, ASE, SET, SFS, WAS) to Advanced Computer Science with the permission of the programme director. Other transfers are possible subject to the relevant requirements being met.

Table of Modules:

<table>
<thead>
<tr>
<th>Main Modules</th>
<th>ACM</th>
<th>ADS</th>
<th>ASE</th>
<th>SET</th>
<th>SFS</th>
<th>WAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO7200:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7104:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7205:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7206:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7207:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7208:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7209:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7211:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7212:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7213:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7214:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7215:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7216:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7217:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO7218:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*yellow = Core Modules*
Postgraduate Taught Courses in Computer Science

MSc in

Advanced Computer Science

Course Aims

Advanced Computer Science (ACS) allows you to supplement your education in a specific set of advanced topics and cutting-edge technologies of your choice, whether you need them for your professional life, or to pursue your studies further to a PhD on a specific theme.

Programme Structure

This MSc offers a broad menu of modules from which you can construct your own pathway with the advice and assistance of a member of staff.

The only core modules for this programme are those shared by all programmes: CO7201 – Individual project, and CO7210 – Personal and Group Skills. You will need to take seven additional modules, of which at most two can be from a supplementary list published every year, including topics such as analysis and design of algorithms, communication and concurrency, compression methods for multimedia, cryptography and internet security, financial mathematics, formal software specification, operations research, semantics of programming languages, software measurement and quality assurance, inter alia.

“I read about the degree on the internet and was very impressed with the course content. The structure of the course and the module topics within it made me decide to come to the University of Leicester. The lecturers are fantastic – they are very helpful and the resources at Leicester are impressive. The computer suite that we used is very modern and I was able to work whenever I needed to.

Studying at postgraduate level is quite challenging. The lecturers allowed the class to drive the course of learning and we shared our opinion and views on aspects of the degree. This allowed us to shape our own thinking.

I have graduated and am now working as an IT project coordinator and developer at a software development company that develops web application. My Master’s degree equipped me with software development process, management and design skills that are vital for my job. I also gained an understanding of the prevalent technologies that are used for creating enterprise software applications. In particular aspects of the course about design and coding software that can be maintained and re-engineered easily have been especially useful to my current career.

The city is a lovely multicultural place and this was reflected in my class where many of my fellow students were from around the globe – I really feel this adds to the learning experience. I have stayed to work in Leicester and I continue to enjoy the City.

My favourite aspect of studying with the Department of Computer Science at Leicester was the academic environment, Library and the availability of the lecturers to assist the students. The interaction I had with my course mates from different nationalities made my experience at Leicester very special. The quality of education and the level of coursework will help any prospective student to acquire more within a short time. It is a place for zealous students and I would certainly recommend Leicester to anyone.”

Walter Magulike
MSc in

Advanced Computational Methods

Course Aims
The field of algorithms is today an extremely important enabling technology. Web search-engines, routing in the Internet, genome analysis, cryptography and image analysis are just a few examples of applications that depend critically on suitable choices of algorithms and data structures. The focus of this MSc is on the design, analysis and engineering of algorithms, covering their use for modelling real-world problems.

Programme Structure

Core modules
In addition to the core modules that are shared by all programmes (CO7201 – Individual project, and CO7210 – Personal and Group Skills), this programme has five core modules:

- CO7100 – Algorithms for Bioinformatics.
- CO7104 – C++ Programming and Advanced Algorithm Design.
- CO7212 – Game Theory in Computer Science.
- CO7213 – Networking and Distributed Computing.

The market
Algorithmics is becoming an essential tool for researchers in many areas outside Computer Science. A 2006 article in the Communications of the ACM notes:

“Algorithmic thinking is transforming both the descriptive sciences and the humanities, bringing them all closer to the mathematical core of computer science.”

One reason for this is the increased reliance on describing complex processes (e.g. the stock market, evolution) as algorithms, and using algorithmic simulations to make predictions.

Applications in industry are also widespread. A 2005 report from the Smith Institute for Industrial Mathematics noted that:

“An extensive survey of leading industrialists has highlighted three broad areas [at the mathematics/computer science interface] to which there should be attached a particular priority: network behaviour, algorithmics and information management.”

Graduates from this programme will be well equipped for careers in financial market analysis or computer network analysis with a view of detecting and resolving problems that require quantitative analysis, simulation and prediction. For instance, data warehousing and internet search companies have been investing heavily in staff with these skills.

Our core modules touch upon some recent and important areas of algorithmic applications, such as bioinformatics and large computer networks. Modules such as Game Theory in Computer Science and Discrete Event Systems provide the tools with which you may model large and complex systems as algorithmic processes, and C++ programming and advanced algorithm design will give you the skills to write the software needed. In your project you have the freedom to explore the full range of algorithmic application areas.
MSc in

Advanced Distributed Systems

Course Aims

The presence of software applications as components of many different kinds of systems with dynamic configurations is increasing at an unprecedented pace. The focus of this MSc is on new methods, architectures and design techniques for software systems that are able to operate, with guaranteed levels of quality of service, across heterogeneous and distributed platforms.

Programme Structure

Core modules

In addition to the core modules that are shared by all programmes (CO7201 – Individual project, and CO7210 – Personal and Group Skills), this programme has five core modules:

- CO7205 – Advanced System Design.
- CO7209 – Software Reliability.
- CO7213 – Networking and Distributed Computing.
- CO7214 – Service-Oriented Architectures.

Where are you coming from?

Andrew Moody took A-levels at Uptonby-Chester High School, before going to the University of Leicester where he graduated with a BSc in Computer Science in 2004 and went on to complete an MSc in September 2005.

What are you doing now?

AM: I've just had my one-year anniversary at RI3K, a small-to-mid sized company in London, developing a trading platform for the reinsurance industry along with project collaboration and contract management products. I'm just about to start my second software development project as a team leader.

Will you take tips from David Brent?

AM: The people I work with are great fun and I get to use some of the most cutting-edge technologies that are out there in a fast paced and exciting environment; this is not The Office, and computer doesn’t always say “no” (sic)!

Do people actually have to take a degree in Computer Science to do your kind of job?

AM: As well as learning new programming languages or how to create advanced databases, one of the key things you’ll develop from CS at a good university like Leicester is the ability to approach a problem from a number of angles and with a number of possible solutions in mind.

But doesn’t Computer Science narrow your choice of job too much?

AM: No! There are no ends to the career paths you can follow from a CS background. Graduate schemes abound and most welcome CS students. There are, however, a number of graduate schemes and positions out there for which CS is one of the few, if not the only, background that will afford you access. The bottom line is that a degree in Computer Science is a qualification that’s going to open far more doors to you than it's going to close.
**Course Aims**

This MSc focuses on the methodologies and technologies that address the challenges that companies are facing for competing in the volatile markets of today: How to generate applications from high-level business models to reduce time-to-market and development costs? How to evolve legacy systems and promote business processes in an economy dominated by the need to offer and integrate, new services, on demand?

**Programme Structure**

**Core modules**

In addition to the core modules that are shared by all programmes (CO7201 – Individual project, and CO7210 – Personal and Group Skills), this programme has five core modules:

- **CO7205** – Advanced System Design.
- **CO7206** – System Re-engineering.
- **CO7207** – Generative Development.
- **CO7208** – Software Process Engineering.
- **CO7214** – Service-Oriented Architectures.

---

**Why did you choose to study Computer Science at Leicester?**

Stephen Gorton: Primarily because I wanted to know more about what computers could do and how they could do it. Then, the University has an excellent reputation! I also liked the city and was happy to live there for the duration of the degree course.

**What were the most memorable parts of the course?**

SG: I enjoyed learning to program in other languages. Although some were harder than others, there was always good teaching support. I think the best part of the course was the 3rd year individual project, where there is considerable freedom in what you can study. I developed an application rather like Skype – it was hard work but very rewarding. Leicester also has a very active social scene and I made some good friends.

**You liked it so much you stayed for the MSc?**

SG: Yes! With the influence of my 3rd year project supervisor and help from my parents, I stayed on to take the Department’s MSc in Advanced Software Engineering. During this course, I found that I liked the advanced topics enough to resign from a job that I’d been offered and pursue a PhD.

**What do you mean by advanced topics? More programming?**

SG: Not at all! There is a lot more to software systems than just the code and this course highlighted that. Current hot topics such as software architecture, legacy systems and reliability were all addressed. One exciting topic was how software could work in a massively distributed way, with components communicating through the Internet. This is known as Service Oriented Computing, and the subject under which I am studying for a PhD.

**How has this enabled you to get a good job?**

SG: Doing a PhD opens up new horizons of work. I know that many people think that doing a PhD is for university lecturers but consultancy, system planning and research in new methods and techniques are all potential careers. This is what I’m doing at ATX Technologies, a company that contributes to the teaching of system re-engineering in the MSc at Leicester.
MSc in Agile Software Engineering Techniques

Course Aims

Today, software is becoming ever more complex, developed with very short time to market, and required to cope with changing requirements. This scenario calls for increased levels of flexibility and agility, both in the technologies used and the processes followed for engineering software. This MSc will provide a sound background on the methods and techniques that can meet these challenges. Hands-on modules based on the Eclipse framework will ensure that this knowledge is framed in practical contexts of usage.

The Market

Gartner, one of world’s leading analysts, have been emphasising the importance of increasing the need for a business’ IT systems to become more agile to deal with change, accepting that these systems have previously been sold and implemented with a static view of operations. They have been predicting that radical change will occur as organisations realise that IT must act as a key driver for business change.

Agile is what both the latest business strategy and software development methodology aspire to be. “There are a lot of companies experimenting with, or about to experiment with, agile software development,” said Steve Gedney, managing director of Borland’s UK operations, to the Financial Times in November 2007.

So what is it all about? The FT article, aptly called “Flexibility takes over from plans in an agile world”, quotes Professor Donald Sull of the London Business School to define it as “a company’s ability consistently to identify and seize opportunities more quickly and effectively than rivals.” What is clear is that agility is having strategic implications in the way companies and organisations are planning to use IT, leading to a new range of technical skills that software engineers need to acquire.

Programme Structure

This programme aims to prepare students for the software industry of today: a highly agile field making use of structured engineering processes and advanced software production technologies to ensure quality in the light of rapid changes. This is achieved by a number of core modules and options drawn from computer science covering both the forward production and the reverse engineering paths.

Core modules

In addition to the core modules that are shared by all programmes (CO7201 – Individual project, and CO7210 – Personal and Group Skills), this programme has four core modules:

- **CO7217 – Domain Specific Languages.** A domain-specific language (DSL) is a programming language or specification language dedicated to a particular problem domain (such as financial products, service oriented architectures, semantic web and ontologies, mobile agents, simulation, robot control, or digital hardware design, to name just a few application domains). In an article published in Technology Review, Charles Simonyi, head of Intentional Programming corporation and former leader of Microsoft’s flagship office applications, explains the relevance of the application of DSLs in industrial practices:

  “Programmers today […] are “unwitting cryptographers”: they gather requirements and knowledge from their clients and then, literally, hide that valuable information in a mountain […] of code. The catch is, once the code is written, the programmers have to make any additions or changes by modifying the code itself. That work is painful, slow, and prone to error. We shouldn’t be touching the code at all.”

  In this module, we teach and experiment with different techniques for defining DSLs and for automatically generating executable code from them.

- **CO7207 – Generative Development.** The ability to generate code automatically from business models, what is usually known as “model-driven architectures” (MDA) is one of the most important enablers of agility in software production. Over decades, companies have been facing the need to cope with a succession of middleware platforms, which incurs huge costs and risks as software needs to be moved from one platform to the next. Dr Richard Soley, Chairman and CEO of the Object Management Group, said: “Companies that adopt the MDA gain the ultimate in flexibility: the ability to derive code from a stable model as the underlying infrastructure shifts over time.”

- **CO7216 – Service-Oriented Architectures.** Guy Lidbetter, CIO of the big European computing services group Atos Origin, said to the Financial Times in December 2007 why SOAs are considered to be the best enablers of agility in IT:
For 2008, as always, companies will be most successful if IT is strongly aligned with the businesses it supports, going on to point out that companies must migrate to an agile architecture if they are to bring products to market that will have a meaningful impact on earnings and revenue: migrating to a Services Oriented Architecture will be the only way to accomplish this.

**CO7206 – System Re-engineering.** Multiple reports issued in the last few years by major analysts such as Gartner and IDC show that the systems in place at the major consumers of IT (such as financial institutions, telecoms, and public administration) still run on legacy technology (e.g. COBOL). This has created a huge market for re-engineering and enterprise integration, which has been growing at 25% a year. All enterprises face the need to engage in some form of re-engineering project, mainly to migrate to SOAs in order to take advantage of the new business channels that are being made available through the internet. It has been estimated that more than 75% of e-business solutions reuse existing systems in conjunction with package software or outsource development.

---

**Research at Leicester: NAOMI**

General-purpose modelling languages, such as the UML, provide a common set of modelling constructs that are not precise enough to capture the concepts associated with particular application domains in a straightforward way.

The lack of a precise understanding of the meaning of those constructs implies that the use of such languages in the design of large software applications often leads to problems that do not surface until system integration. This is why domain-specific modelling languages (DSML) are being favoured by industry and researched by academia so that they can offer primitives that can capture the semantics associated with a chosen domain (e.g. Finance).

In the project NAOMI, Leicester working together with the University of Illinois at Urbana-Champaign and Lockheed Martin Advanced Technology Laboratories in the USA to explore how the precision of multiple DSMLs can be leveraged, leading to more accurate and complete models. NAOMI will enhance semantic interoperability between heterogeneous partial models, which are defined with different DSMLs, that may describe both the structure and behaviour of the system under design from a specific point view, that is, in a particular domain. NAOMI will also provide the means to combine all these partial models based on an agreed semantics and in consistent manner in order to achieve a complete and precise specification of the whole system under study.

One of the keys to address complexity is to use powerful mathematics based on graphs. Leicester is at the forefront of research in graph transformations and their use for model-driven software development. One of the projects in which we are involved is building MOMENT2 – an algebraic model management framework for manipulating models in the Eclipse Modelling Framework (EMF). This framework will enhance the experimentation of practical graph-based model transformation language features: MOF-based concepts, MOF introspection and structural reflection, graph pattern matching, negative application conditions, conditional production rules, support for the Object Constraint Language (OCL), and formal verification techniques such as checking invariants through reachability analysis and model checking. The framework is being written in Maude, an efficient term rewriting system developed at University of Illinois at Urbana-Champaign and SRI International (USA).
Course Aims

Financial services are not only one of the most dynamic sectors of the economy but also one of the two largest customers of IT! This MSc is offered together with the Department of Economics in order to give you both a command of the software technologies that financial institutions require to “embrace the challenge of change” and of the business context and organisational structures that IT systems need to support.

This MSc programme concentrates on architectures for building scalable financial software systems, thus preparing software engineers for a plethora of jobs in the financial industry. In particular it considers technologies and techniques that are particularly relevant for the challenges of the financial market, predominantly a need to migrate from mission-critical, monolithic legacy systems to more flexible architectures that allow speedy reaction to customer and business partner’s needs. The technical aspect must be seen in the context of the business environment, where software engineers typically interact with a world of financial jargon and departments with specialised roles and needs.

The Market

A special volume of the magazine Communications of the ACM dedicated to New Architectures for Financial Services identifies two main challenges that modern financial systems are having to address:

- Advancements in telecommunications in general and the internet in particular are having deep consequences on the way financial institutions make their services available and make business. The availability and diversity of delivery channels requires that the information systems that support the core business are endowed with a robust but flexible software architecture that can accommodate new ways of making business while preserving core business invariants.
- The speed with which new products need to be launched into the market, new regulations come into force, and mergers and acquisitions take place among financial institutions, is not compatible with the monolithic legacy systems that are still operating in most organisations. In response to these circumstances, new ICT-based or ICT-enabled architectures for technology and organisation are beginning to emerge in the financial services industry.

This view is strengthen by the report Business 2010: Financial Services published by the Economist Intelligence Unit, which aptly subtitled “Embracing the challenge of change”. This is what the report concludes:

“Industry leaders are aware that nurturing the ability to adapt to change is the single most important challenge facing their firms. They also recognise the need to be fast, and just as well. Technology is making it possible for small players to launch innovative new services quickly, and for customers to switch to them just as fast. In this sense, IT offers financial industry firms both immense opportunity and certain competitive threats. Executives realise that how they use IT will do much to determine their ability to face up to the competitive challenges of the next five years.”

In summary, there is a big market for software engineering skills directed to the methods and techniques that are required for designing and deploying new software architectures for financial services, and for re-engineering legacy systems to operate on these new architectures.

Programme Structure

This programme aims to support software engineering skills and knowledge with deep understanding of financial organisations. This is achieved by a number of core modules and options drawn from computer science as well as economics offerings.

Core Modules

In addition to the core modules that are shared by all programmes (CO7201 – Individual project, and CO7210 – Personal and Group Skills), this programme has four core modules:

- **CO7206 – System Re-engineering.** The CACM article mentioned above is very clear on the crucial role of re-engineering techniques for the financial services industry to embrace the challenge of change.
  
  This module is delivered with the support of ATX Technologies Ltd, a company that specialises on the modernisation of legacy system, with a long experience in the financial sector.

  **CO7214 – Service-Oriented Architectures.** From SAP to IBM, major IT providers are promoting SOA as the infrastructure that is best positioned to offer the levels of agility required in the financial services industry. This is what Mike Blum, partner and global banking leader for IBM Business Consulting Services, reported to The Financial Times on May 26, 2004:
The opportunity exists for banks to overcome the inflexibility and complexity of their traditional operational structure and become true “on-demand” banks. Technologies are available today that did not exist ten years ago to integrate business components. With middleware and web services we now have the technical enablement in place to take advantage of sharing business components across the enterprise.

- **CO7218 – Financial Services Information Systems.**
  The vast majority of companies worldwide are looking for people that can manage or deploy IT effectively in the operational context of the business that they run. This module will provide an introduction to the key organizational units and functions that IT systems need to support in financial services organisations. Planning and delivery of this module is joint with ATX Technologies Ltd, Mr G.Koutsoukos (Eurobank EFG) also with the collaboration of other colleagues in leading financial institutions.

- **EC7061 – Corporate Finance.**
  This module is delivered by the Department of Economics. It gives you an introduction to the theory of corporate finance so that you can understand the business context in which you will operated. Topics include takeovers, mergers and dividend policy; Debt vs Equity trade-off; Contracts in the venture capital industry; Capital structure and incentives.

---

**Research at Leicester: Leg2Net**

Leicester and the software company ATX received funds from the Marie-Curie Transfer of Knowledge programme to establish an industry-academia partnership. Under this partnership (known as Leg2Net), ATX has been participating in the delivery of the MSc programmes and doing joint research with Leicester aimed at new methods and techniques for re-engineering “legacy” software into service-oriented architectures, and on ways of supporting evolution within those architectures. **But why is this important?**

To many people, software is something that allows us to use a computer to perform certain activities (like writing a piece of text). However, software doesn’t just sit inside the computer: it has a life of its own! For instance, software needs to be continually adapted in order to deliver the same level of satisfaction to the user (or even increase it). This is known as Lehman’s first law of software evolution.

Large organisations such as banks use very complex software applications and evolving them is a much more challenging task. If one is not careful, complexity increases: layers of software are often added without restructuring what was there already, or new applications are coarsely stitched to old ones without taking into account the architecture of the system. In modern terms, we could say that software becomes “obese”: for instance, code that is no longer needed is not deleted, or code gets replicated all over the application in marginally different versions.

As it lets “fat” accumulate, software becomes less and less efficient, more and more difficult to change, and lacking the levels of agility, flexibility, and responsiveness that companies. As with humans, one can make a surgical operation to remove the fat. In software engineering, this requires a careful analysis of the code, breaking it into meaningful chunks so that one can understand what is “fat” and what is “muscle”, and reorganise what is left so that the original functionality is preserved. These rejuvenation techniques (re-engineering in computer science speak) are based on graphs and require sophisticated mathematical operations: it’s a job for specialists, supported by clever software tools!

Again as with humans, if we don’t change our life style, fat will accumulate again. One of the methods that has been gaining popularity for keeping software fit and agile is the adoption of an “architecture”, i.e. a way of organising software as a structure of interconnected (smaller) components and restricting changes on software to reconfigurations that conform or preserve that structure. In other words, evolution has to adhere to a (strict) “regimen”, which allows organisations to plan and optimise the use of resources, as well as control the quality of their systems.
Course Aims

The evolution of web-based technologies has now led to what is known as “Web 2.0”: a semantically enriched information source with advanced potential to provide specialised software applications “on the fly”. A plethora of standard PC-based applications is now appearing online (calendar and diary tools, text editors, spreadsheets, among other) that can be used in a distributed collaborative setting. Developing such applications is particularly challenging, partly due to the wide background required but also the rapid emergence of new technologies. This MSc is intended to equip students with a sound understanding of the area and its emerging trends, while at the same time providing a very hands-on approach to current technologies such as .Net.

The Market

Articles about Web Applications and Web Services can be found daily in the computer press, but also in the Economist or Financial Times, highlighting the fast development of this area in a technical sense, but also its commercial relevance for businesses all over the world.

In 2004 the Radicati Group made the following prediction on the growth of the Web Services market:

“The nascent market for Web services will swell dramatically over the next four years, spreading well into the global arena, according to new research released today.”

Radicati Group’s “Web Services Market 2004-2008” reports that the combined market for Web services solutions, management, integration and security will be worth $950 million in 2004. By 2008, that figure will climb to $6.2 billion.

This prediction was confirmed by IBM in an article published by the Financial Times in April 2006:

“IBM’s business from service-oriented architecture has doubled over the past year […] SOA […] is seen as a fundamental architectural shift that will pull the industry out of its post-bubble slump.”

Programme Structure

This programme aims to support software engineering skills for applications that are distributed on the web, while familiarising students with the practical aspects of relevant software development environments and frameworks. This is achieved by a number of core modules and options drawn from computer science addressing the whole range from implementation to conceptual understanding.

Core modules

In addition to the core modules that are shared by all programmes (CO7201 – Individual project, and CO7210 – Personal and Group Skills), this programme has four core modules:

- **CO7205 – Advanced System Design.** Taking advantage of the business opportunities being made available by the Web requires new methods of building IT systems that rely not on big, integrated programs but on small, modular components. Not that software as we know it will disappear, but we need ways of modelling applications that can shop around for services provided by external parties and bind to them at run time to fulfill business needs. Stuart Feldman, from IBM, offered The Economist the following view of software production in the new age of web applications:

  “Plenty of code will still be needed to make the new world of computing run, just as mainframe computers are still around, though in a much less dominant position. But the computer business will no longer revolve around writing big, stand-alone programs. Instead, it will concentrate on using software to create all kinds of electronic services, from simple data storage to entire business processes.”

- **CO7215 – Advanced Web Technologies.** This module covers the very practical aspect of implementing service oriented systems using Microsoft’s .net technologies. It familiarises the student with skills and tools that are directly relevant to industry, while also introducing more advanced aspects such as business processes and their implementation in BPEL.

- **CO7216 – Semantic Web.** As the industry strives towards assembling applications at run-time from smaller components, it is no longer sufficient to have
interfaces matching (a syntactic aspect) and humans deciding on whether the component fulfils the required task (has the right meaning, or semantics). It becomes predominant that machines can make the decision on the semantic aspect and hence a plethora of requirements open up: how can the semantics of components be captured, how can decisions be made, etc. Semantic web technologies introduce this capability.

- **CO7214 – Service-Oriented Architectures.** “This is the industrial revolution for software,” said Toby Redshaw, vice-president of information technology strategy at Motorola, the US electronics group, to the Financial Times in January 2005. He was talking about the rise of service-oriented architectures (SOAs). The software industry has been quick to sense an opportunity in SOAs. Big software companies such as BEA Systems, IBM, Oracle and Microsoft jumped on the bandwagon 4 years ago. Since then, they have been joined by a host of start-ups offering products to help manage SOAs. According to Peter Sondergaard, Gartner’s head of research, chief information officers’ (CIOs) are being urged to move away from siloed IT and re-organise the infrastructure as a series of processes that link business operations:

  “CIOs should put service-oriented architectures – the process whereby applications and data render software components as services, thereby speeding project deployments and application changes – at the top of their agendas.”

Research at Leicester: inContext

Imagine a very common situation: a project in a large company needs to hold a meeting to make decisions on the next steps ahead. The meeting requires the attendance of the project’s key people – all busy visiting customers around the world – as well as people with specific skills (say, a web designer). In order to get in touch with all these people, find out about their availability, choose a convenient time when they can all attend or make themselves represented, and select the web designer, a secretary will certainly be busy for the best part of several days…

The situation described above is an instance of what is called “collaborative work” – an area in which computer scientists are working to develop innovative technology that can support the work of teams. The challenges are multiple: teams bridge company boundaries, its members move around, people use mobile devices and work on many things at the same time, and so on.

The EU-funded research project inContext ([www.in-context.eu](http://www.in-context.eu)) is developing a platform and techniques that make use of service-oriented computing to integrate existing tools (such as email systems, calendars, project schedulers) into a coherent system that can be used on any device, anywhere in the world, to make collaborative work more productive.

So far, the project has concentrated on the development of a Pervasive Collaboration Service Architecture (PCSA) that allows users to connect from a PC, a mobile phone or a PDA to the system and request services. The system automatically decides which services to offer based on the context of the requesting user and others involved in the activity: where are they? what are they doing? what have they done in similar situations before? Making such decisions is not easy: it involves methods and techniques that support data mining, the gathering and modelling of context information, and reasoning about models in order to derive new facts. The automation of the decision-making process involves sophisticated algorithms and methods.

Many scientific results have by now been produced, and the viability of the PCSA has been demonstrated through a prototype meeting scheduler: when secretaries tell the system that a meeting is required, it will automatically collect names of people who can represent those that cannot attend, find experts in specific areas, and suggest alternative times for the meeting. It will even send invitations to people on the device that they use: email, instant messages or SMS to a mobile phone.
Contact Details:
The MSc Admissions Team
Department of Computer Science
University of Leicester
Leicester LE1 7RH

e: csmsc@mcs.le.ac.uk

t: +44 (0)116 252 3884

f: +44 (0)116 252 3915

w: www.cs.le.ac.uk