Introduction

Ferruccio Damiani

Università di Torino
www.di.unito.it/~damiani

Mobile Device Programming
(Laurea Magistrale in Informatica, a.a. 2018-2019)

Alcune frasi che potrebbe essere utile ricordare

- *Per me l'uomo colto è colui che sa dove andare a cercare l'informazione nell'unico momento della sua vita in cui gli serve.*
- *Tutti aspiriamo al meglio ma abbiamo imparato che talora il meglio è nemico del bene, e dunque negoziando si deve scegliere il meno peggio.*
- *Definite sempre un termine quando lo introducete per la prima volta. Se non sapete definirlo evitatelo. Se è uno dei termini principali della vostra tesi e non riuscite a definirlo piantate lì tutto. Avete sbagliato tesi (o mestiere).*
- *Per non apparire sciocco dopo, rinuncio ad apparire astuto ora. Lasciami pensare sino a domani, almeno.*
- *Il computer non è una macchina intelligente che aiuta le persone stupide, anzi, è una macchina stupida che funziona solo nelle mani delle persone intelligenti.*

Unberto Eco (5 gennaio 1932, Alessandria - 19 febbraio 2016,[1] Milano)

[1] La prima lezione della prima edizione di questo corso si è tenuta lunedì 22 febbraio 2016.

Outline

Outline

Me

- `http://www.di.unito.it/~damiani/`
- Research interests
  - Computational models and languages, Domain specific languages
  - Concurrent, distributed, and mobile systems
  - Cyber-Physical Systems, Internet of Things, Smart Cities, Wireless Sensor Networks
  - Edge/Fog/Cloud Computing
  - Self-organization, Swarm intelligence
  - Static and dynamic analysis techniques
  - System evolution and dynamic software updates
  - Variability modeling and software product lines

You

- Who are you?
- What do you hope to get out of this class?

Please subscribe to the Moodle of the course!

Outline

Learning objectives

The course provides a basic knowledge of design principles and foundation for mobile application development by presenting Android development and introducing some aspects of iOS and cross-platform development. Moreover, it presents the aggregate programming paradigm for programming distributed systems (like, e.g., set of mobile devices) by specifying the global behavior and automatically derive the local behaviors.

Required incoming expertise

Knowledge of the basic concepts and notions introduced in the courses of the "Laurea in Informatica" at the University of Torino or in the courses of any other degree that satisfies the requirements for obtaining the GRIN certification (called "bollino GRIN"). In particular:

1. Programming (good object-oriented programming skills);
2. Algorithms and data structures;
3. Computer architecture;
4. Operating systems;
5. Databases;
6. Formal languages and translators;
7. Software engineering;
8. Human-computer interaction and web development.

Learning outcomes

Knowledge of the basic design principles and on the basic development notions for mobile applications. Basic knowledge of the aggregate computing paradigm. The student will be able to:

1. design and develop Android mobile applications;
2. design and develop very simple iOS mobile applications;
3. design and develop very simple cross-platform mobile applications;
4. write simple programs in a language that supports the Aggregate Computing paradigm.

Come si svolgono le lezioni (in Italian)

Sono previste 24 ore di lezione frontali e 24 ore di attività in laboratorio, che seguono il programma riportato più avanti, integrate da casi di studio e da esercitazioni volte ad illustrare l'applicazione pratica dei concetti appena studiati.

Course Syllabus

- Introduction to the design of mobile applications.
- Mobile application development:
  - Android platform;
  - iOS platform (hints);
  - cross-platform (hints).
- Brief introduction to the aggregate programming paradigm.

Teaching material

See the Moodle web site of the course.

Un consiglio (in Italian)

Alcune opinioni su come scrivere bene potrebbero esservi utili per scrivere la relazione.

- Un libro di Beppe Severgnini: *L'italiano. Lezioni semiserie.*
- I 10 consigli di Josh Bernoff:

  `https://www.linkedin.com/pulse/10-top-writing-tips-psychology-behind-them-josh-bernoff`
- Le cose che Claire Le Goues continua a ripetere:

  `https://clairelegoues.com/2016/08/23/things-i-keep-repeating-about-writing/`

Controllo dell'apprendimento durante il corso (in Italian)

Come parte integrante delle attività in laboratorio il docente discuterà con gli studenti i casi di studio proposti a lezione e il modo in cui sono stati svolti gli esercizi assegnati.

Modalità di verifica dell'apprendimento (in Italian) [1 / 3]

Gli studenti si organizzeranno in gruppi (costituiti di norma da 1 o 2 persone, al massimo da 3). Durante il corso ciascun gruppo negozierà con il docente un progetto di esame (volto alla realizzazione di una applicazione mobile):

- ogni gruppo dovrà proporre un'idea;
- industrie e/o docenti potrebbero presentare le loro esigenze durante il corso;
- la negoziazione dovrà, di norma, avere termine prima della fine del corso.

Modalità di verifica dell'apprendimento (in Italian) [2 / 3]

L'esame verterà sul progetto realizzato:

1. Prima dell'esame (di norma, almeno due settimane prima) il gruppo dovrà consegnare al docente una relazione che illustri il progetto, motivi le scelte progettuali, e contenga la documentazione dell'applicazione mobile realizzata.
2. Durante l'esame ogni gruppo dovrà:
   - fare una breve presentazione (circa 20 minuti, usando delle slides) per introdurre l'applicazione mobile realizzata e illustrare (motivandole in modo adeguato) le scelte progettuali effettuate,
   - presentare un dimostratore (su un dispositivo reale o su un simulatore),
   - discutere il progetto con il docente (durante la discussione ciascun componente del gruppo dovrà dimostrare di padroneggiare i concetti e le nozioni presentate durante il corso), e
   - dimostrare di conoscere le basi del paradigma "aggregate programming" e saper scrivere semplici programmi in un linguaggio che supporti tale paradigma.

Modalità di verifica dell'apprendimento (in Italian) [3 / 3]

Il voto finale sarà stabilito considerando la valutazione di:

1. relazione che illustra progetto e presentazione del progetto all'esame [peso 30%];
2. implementazione del progetto (completo delle sue parti secondo quanto descritto nella relazione) e presentazione del dimostratore all'esame [peso 30%];
3. colloquio (discussione del progetto e domande sulle nozioni presentate a lezione, incluso il paradigma "aggregate programming") [peso 40%].

L'esame farà riferimento al materiale didattico indicato sul sito moodle dell'ultima edizione del corso.

Outline

Some comments

- Project: a well-engineered significant app
  - Some screens (significant application flow)
  - Multiple threads
  - Interaction with external services (not just Facebook)
  - “Nice” look and feel
  - Support for different classes of devices
  - Testing
- Exam:
  - Earlier does not mean higher grades
  - Details matter
  - No marketing strategies
  - Synergies are encouraged

Outline

1951 - The UNIVAC I (UNIVersal Automatic Computer I) was the first commercial computer to attract widespread public attention.

The first UNIVAC was accepted by the US Census Bureau on March 31, 1951. A total of 46 systems were eventually built and delivered (sold at more than $1 million each). UNIVAC I used 5,200 vacuum tubes, weighed 13 metric tons, consumed 125 kW, and could perform about 1,905 operations per second running on a 2.25 MHz clock. The Central Complex alone (i.e. the processor and memory unit) was 4.3 m by 2.4 m by 2.6 m high. The complete system occupied more than 35.5 m^2 of floor space. The main memory consisted of 1000 words of 12 characters. When representing numbers, they were written as 11 decimal digits plus sign.

1977: The Apple II was one of the first highly successful mass-produced microcomputer products.

The first computers went on sale on June 10, 1977 with a microprocessor running at 1.023 MHz, two game paddles, 4 kB of RAM, an audio cassette interface for loading programs and storing data. The video controller displays 24 lines by 40 columns of monochrome, upper-case-only text on the screen. The original retail price of the computer was $1,298 USD (with 4 kB of RAM) and $2,638 USD (with the maximum 48 kB of RAM).

1983: The Motorola DynaTAC 8000X.

The first commercially available cellular phone small enough to be easily carried. A full charge took roughly 10 hours, and it offered 30 minutes of talk time. It also offered an 8-character LED display for dialing or recall of one of 30 phone numbers. It was priced at $3,995 in 1984. Dimensions: 300mm $\times$ 44mm $\times$ 89mm. Weight: 800g. [Martin Cooper of Motorola made the first publicized handheld mobile phone call on a prototype DynaTAC model on April 4, 1973 (the photo on the right is a reenactment in 2007).]

Available in August 2015: Nexus 5

Available in August 2015: Nexus 6

Available in August 2015: Nexus 9

Available in August 2015: iPnone 6 plus, iPhone 6, iPhone 5 [S], iPhone 5 [C]

Available in August 2015: iPad Air 2, iPadAir, iPad mini 3, iPad mini 2

Available in August 2015: Apple Watch, Apple Watch Sport, Apple Watch Edition

Available in August 2015: Pebble Time

Available in August 2015: ...

A classification

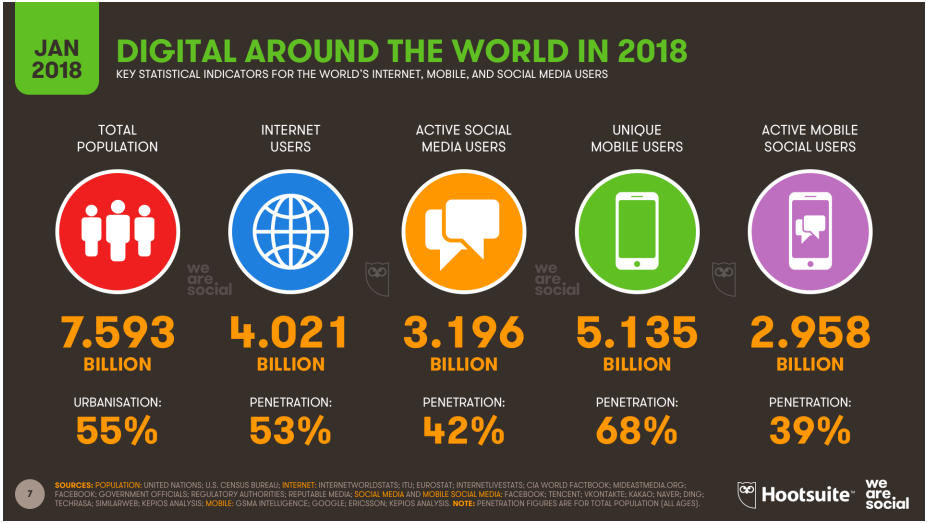
Mobile Backend-As-A-Service (MBaaS)

Quoting from WIKIPEDIA:

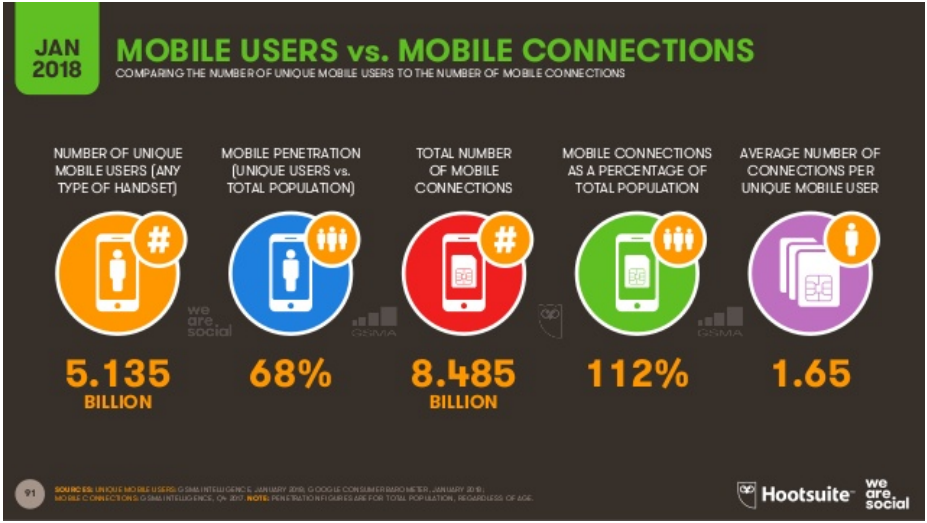
> *A model for providing web app and mobile app developers with a way to link their applications to backend cloud storage and APIs exposed by back end applications while also providing features such as user management, push notifications, and integration with social networking services. These services are provided via the use of:*
>
> - *custom software development kits (SDKs) and*
> - *application programming interfaces (APIs).*
>
> *Although similar to other cloud-computing developer tools, such as software as a service (SaaS), infrastructure as a service (IaaS), and platform as a service (PaaS), BaaS is distinct from these other services in that it specifically addresses the cloud-computing needs of web and mobile app developers by providing a unified means of connecting their apps to cloud services.*

Outline

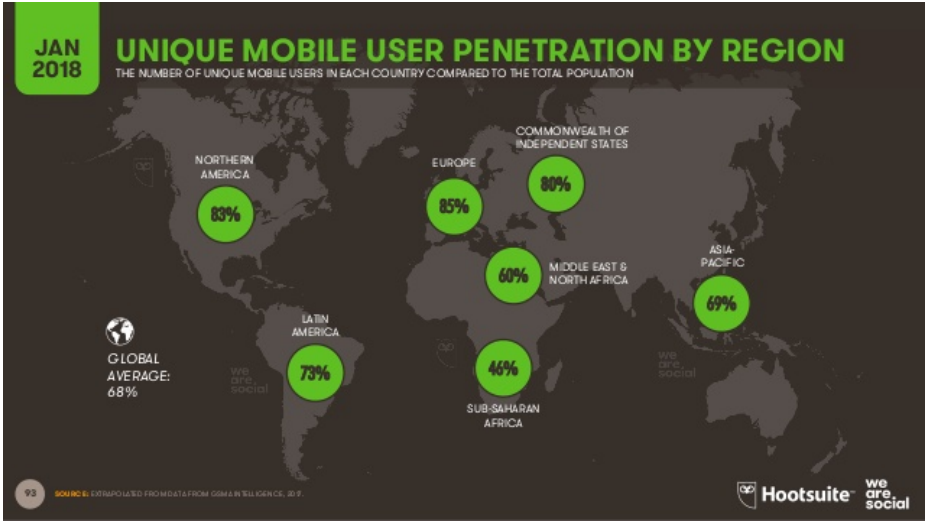
[https://wearesocial.com/uk/blog/2018/01/global-digital-report-2018]



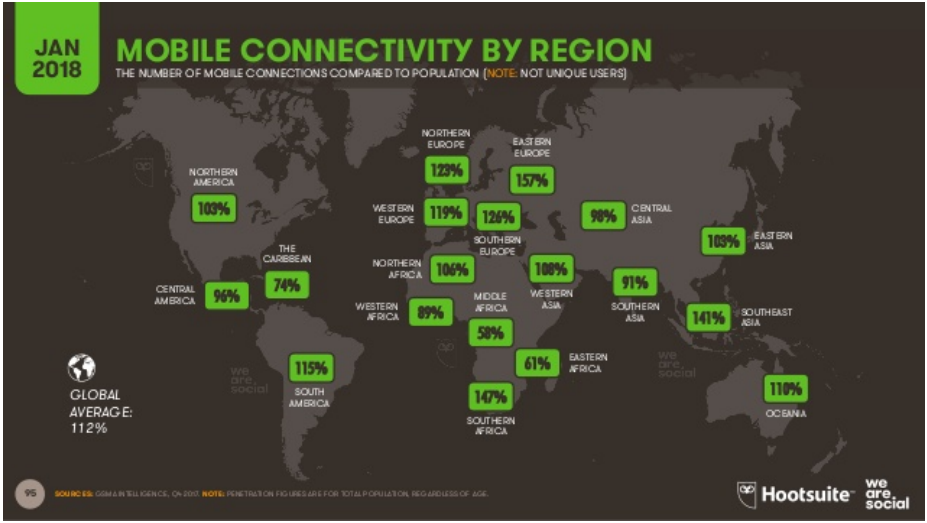
[https://wearesocial.com/uk/blog/2018/01/global-digital-report-2018]



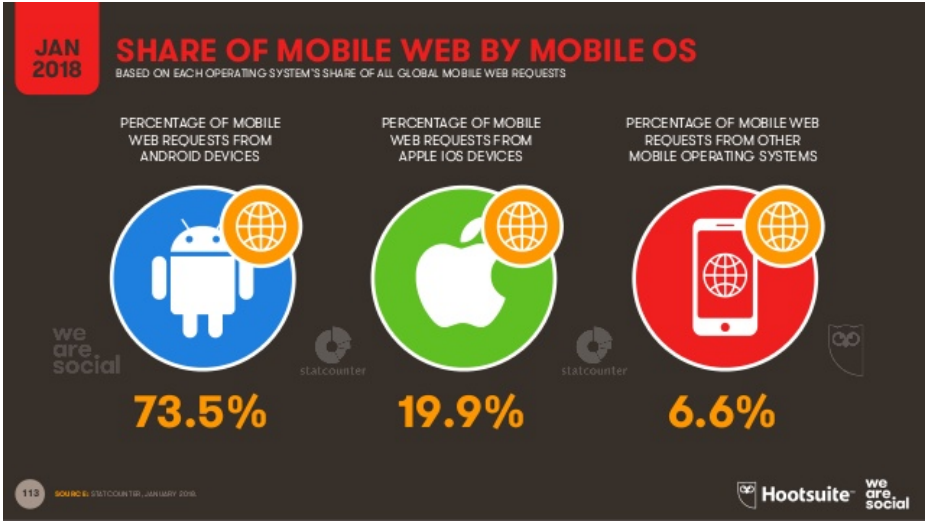
[https://wearesocial.com/uk/blog/2018/01/global-digital-report-2018]



[https://wearesocial.com/uk/blog/2018/01/global-digital-report-2018]



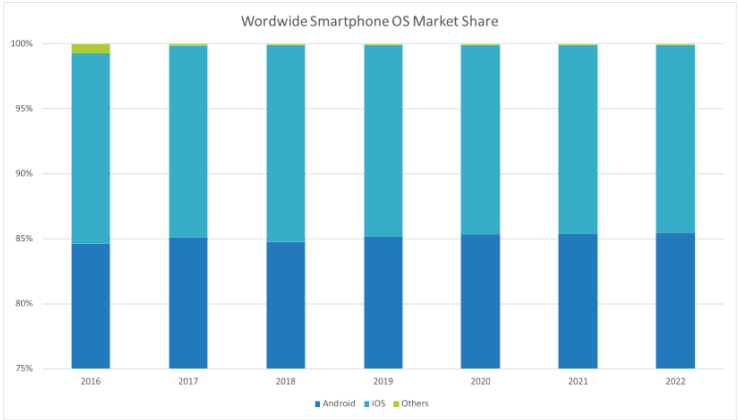
[https://wearesocial.com/uk/blog/2018/01/global-digital-report-2018]



[https://wearesocial.com/uk/blog/2018/01/global-digital-report-2018]

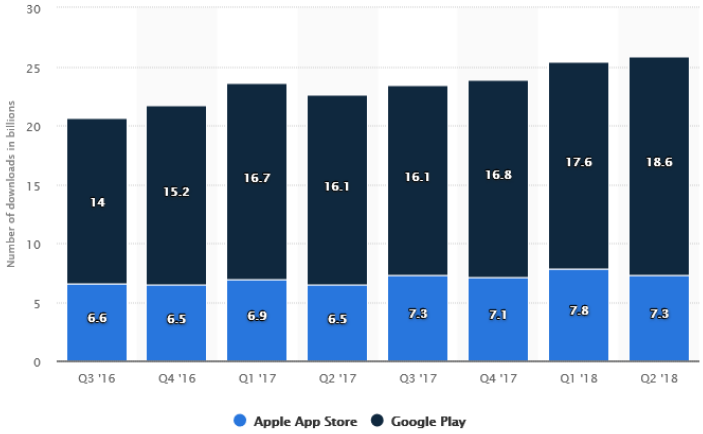
[https://www.idc.com/promo/smartphone-market-share/os]

While the worldwide smartphone market is expected to decline again in 2018, IDC believes the market will experience low single-digit growth from 2019 through the end of its forecast in 2022. The International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker forecasts worldwide smartphone shipments to decline 0.7% in 2018 to 1.455 billion units, down from 1.465 billion in 2017. Read More ▼



[https://www.statista.com/statistics/695094/quarterly-number-of-mobile-app-downloads-store/]

Number of iOS and Google Play mobile app downloads worldwide from 3rd quarter 2016 to 2nd quarter 2018 (in billions)



DOWNLOAD SETTINGS SHARE

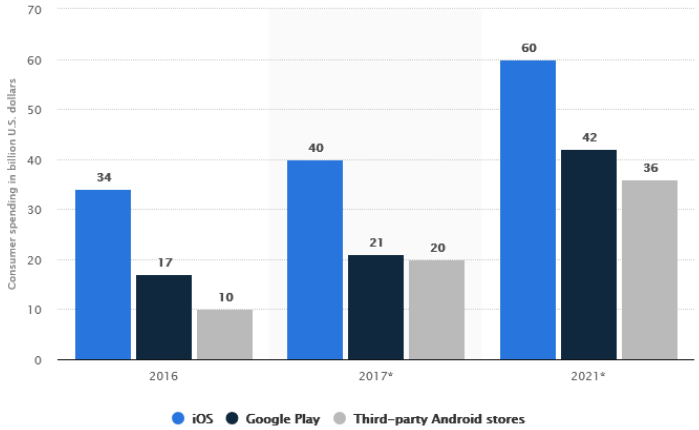
PNG + PDF + XLS + PPT +

DESCRIPTION SOURCE MORE INFORMATION

This statistic presents the number of mobile app downloads worldwide as of the second quarter of 2018, sorted by app store. In the most recently measured quarter, users downloaded 18.6 billion apps from Google Play

[https://www.statista.com/statistics/747489/annual-consumer-spend-mobile-app-by-store/]

Gross mobile app consumer spending worldwide in 2016, 2017 and 2021, by store (in billion U.S. dollars)



DOWNLOAD SETTINGS SHARE

PNG + PDF + XLS + PPT +

DESCRIPTION SOURCE MORE INFORMATION

This statistic presents a forecast regarding the mobile app consumer expenditure worldwide in 2016, 2017 and 2021, sorted by app store. In 2021, consumers are projected to spend 42 billion U.S. dollars on mobile apps from the Google Play Store.

Rapid evolution (and questions becoming dated)

- What about smartwatches? [http://sysrun.haifa.il.ibm.com/hrl/demobile2015/program.shtml]
- Will smartwatches kill smartphones? [http://www.wired.com/insights/2014/08/will-smartwatches-kill-smartphones]
- What about smartglasses?
- More in general, what about wearables devices?
- ...

Outline

Software

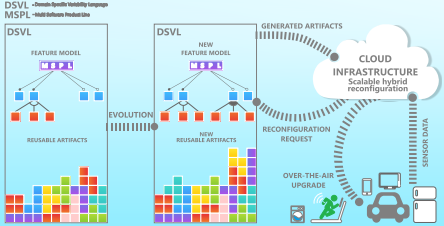
Today

- Software is a fundamental part of many products
  - Mobile phones, cars, washing machines, ovens,...
- The software is not the product, it is a part of the product
  - It must be engineered with the rest of the product
- Sealing the product/project specification is often unfeasible
- Change and evolution are often unavoidable
  - Software is part of social and business processes
  - those are continuously evolving
- Open issues on software updates
  - E.g., how is it possible to connect a car to Internet in a safe and effective manner?

HyVar

Scalable Hybrid Variability for Distributed Evolving Software Systems

http://hyvar-project.eu



HyVar proposes a development framework for continuous and individualized evolution of distributed software applications running on remote devices in heterogeneous environments processes.

This framework combines:

- **Domain specific variability language** to describe evolution as software product line.
- **Scalable cloud infrastructure** for monitoring and individualized customization of software upgrades for the remote devices.
- Continuous software evolution using **over-the-air upgrade** technologies.

The framework ensures upgrades will be seamless and sufficiently nonintrusive to enhance the user quality experience, without compromising the robustness, reliability and resilience of the distributed application instances.

Reply
santer
UNIVERSITÀ DEGLI STUDI DI TORINO
Technische Universität Braunschweig
UiO : University of Oslo
atbrox
MAGNETI MARELLI

"HyVar project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644429"

Outline

Mobile Device Programming

- Different screens (from small large ones)
- Different OSs and different versions
- Scarce resources (memory, disk, battery)
- Unreliable and mutable connectivity (GSM, WiFi)
- Data transfer: costly, slow, high latency
- Priorities (what if a phone call comes in?)
- Inter-app communication
- Development model (cross compilation)
- Distribution model (store)
- ...

Complex devices

- Accelerometer
- Gyroscope
- Digital compass
- Global Positioning System (GPS)
- Barometer
- Ambient light
- Proximity Sensor
- ...

Mobile Applications

Software that can be pushed to a mobile device or downloaded and installed locally to serve some needs

- Browser-based applications are developed in a markup language
- Native applications are compiled solutions (device has a runtime environment)
- Hybrid applications are aimed to exploit the best of both worlds (a browser is needed for discovery)

Mobile application design

- A mobile app should do one thing and do it well
- A mobile app should be as simple as possible, but not simpler
  - Minimalist design (e.g., no quit button)
  - Usability design
  - Standards compliance
  - ...
- Different versions (families of applications)
- ...

Many different languages

- Kotlin/Java (Android)
- Swift/Objective-C (iOS)
- Javascript (cross-platform development)
- HTML5 (Tizen, cross-platform development)
- C# (Windows Mobile, cross-platform development)
- C++ (Tizen)
- ...

Many different languages

- Kotlin/Java (Android)
- Swift/Objective-C (iOS)
- Javascript (cross-platform development)
- HTML5 (Tizen, cross-platform development)
- C# (Windows Mobile, cross-platform development)
- C++ (Tizen)
- ...

We should try to identify key concepts—also by looking at how ideas and technologies originated, evolved and are going to evolve.

Outline

The advent of the Internet of Things (IoT) is bringing us a dramatic increase of density of computational devices deployed in our cities and living and working environments. This kind of network which we refer to as a (spatially) situated network, poses novel engineering challenges:

- Interactions encompass a large-scale system and often need to be opportunistic, context-dependent, and based on physical proximity, which implies they must be adaptive and resilient to nearly continual faults and changes in the network.
- Such networks are also highly heterogeneous in distribution, due to the underlying heterogeneity in human activity and environmental structure that drives the deployment of network devices. For example, a city center is likely to host a high density of cars, traffic sensors, and smart signage, while a country road has very few by comparison. Likewise, an office plaza may host many devices by day, few at night, and massive numbers during sporting events or festivals.

Developing applications for such environments can be extremely difficult, as conventional development methods require that a programmer simultaneously address networking protocols, coordination mechanisms, and the application itself.

Outline

Big data is data sets that are so voluminous and complex that traditional data processing application software are inadequate to deal with them. Big data challenges include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, updating and information privacy. There are three dimensions to big data known as Volume, Variety and Velocity. [https://en.wikipedia.org/wiki/Big_data]

- Many opportunities—see, eg., the Big Data Value Association (BDVA) [http://www.bdva.eu/], and
- dark sides too [https://datacloud.di.unito.it/index.php/s/BPaEQKFdPuwJee6]

Outline

The aggregate programming paradigm

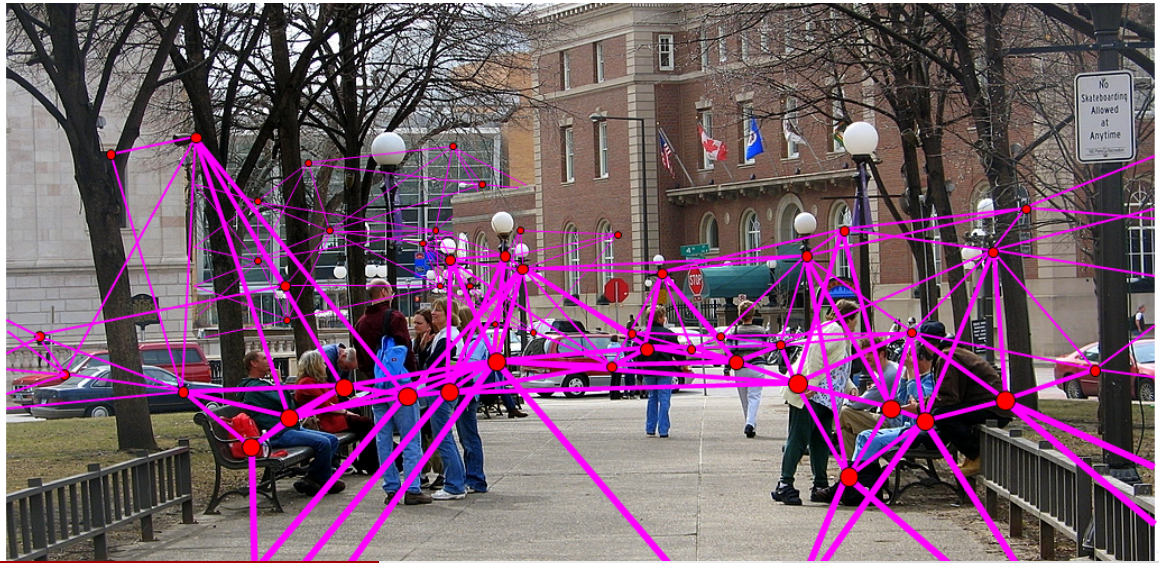
- The aggregate programming paradigm for programming distributed systems (like, e.g., set of mobile devices) by specifying the global behavior and automatically derive the local behaviors.
  - Computational fields
  - The domain specific language ScaFi (`https://github.com/scafi/`)

Networked devices are filling our environment...

...


No
Skateboarding
Allowed
at
Anytime
STOP

...How do we program aggregates robustly?



Outline

Acknowledgements

Some of the slides presented in this course are based on slides by Luciano Baresi.