Systems 3

Teodora Ristic

August 2021

1 Basic Terms

1.1 Data, Information, Knowledge

- **Data** is the lowest form and it refers to unstructured facts or figures that lack any kind of organization in form of numbers, letters, images, sound. It's a formalized structure of information that is suitable for communication, interpretation or processing. Data attributes are: accuracy, precision, details, wholeness, age, structure, accessibility, source, data type.
- **Information** is the knowledge derived from data. Information allows one mind to influence another. According to Shannon, it reduces uncertainty after arrival of data. Information attributes are relevance, timeliness, reliability and verification.
- **Knowledge** represents a deeper level of understanding. It's a combination of rules, procedures, ideas and instincts that influence actions and decisions. It can be explicit (programming, reading, writing) or implicit (not explicitly expressed, like intuition, feeling for aesthetics, body language...)

1.2 Information Theory

The theory of information is based on the theory of probability and statistics. The key measure used in information theory is entropy.

Entropy defines the amount of uncertainty that is associated with the value of the random variable or the result of a random process. It's a measure of disorder. With predictability, entropy drops. Bit is a unit of entropy.

1.3 Information Systems

Information system is a set of interdependent components that collect, process, store and distribute data providing users with information relevant to decision making. Information systems can be formal and informal. **Formal IS** have a clearly defined structure where data they operate, procedures, rules, policies and roles are defined. They are based on the organization framework

for the organization. **Informal IS** is based on individuals and their needs and it doesn't have defined rules.

1.4 Informatics and Computing

- **Informatics** is a scientific discipline that explores the design, structure, functions, organization and operation of information systems. It deals with information systems and uses the computer as an end object for achieving goals.
- **Computing** deals with the principles of programming and problem solving using a computer.

1.5 Basics about Systems

System is a group of interconnected components. System elements work together to achieve a goal or perform a certain function. They are interconnected.

Systems consist of subsystems (components). They are all interdependent, not isolated. The boundaries of a system are determined according to the purpose of a system. **The system environment** is a set of elements that aren't part of the system, but there is some interaction between them and the system.

System inputs allow the transfer of physical objects and information from the environment to the system. **System outputs** allow the transfer of physical objects and information from the system to the environment.

Types of systems:

- **Static** = the state of the system doesn't change under the influence of the environment.
- **Dynamic** = the system depends on the environment and it changes. It can change in any part like hardware, software...
- **Simple systems** = they don't need control.
- **Complex systems** = they need control (collecting information on system operation, analyzing operation, deciding and forwarding instructions).

System attributes:

- **Entropy** = measure of chaos in a system. Function of probability of system state.
- **Dynamic balance** = the ability of the system to always find itself in a state of relative stability despite effects and disturbances.
- **Adaptability** = the ability of the system to always find itself or cause change in the environment if the environment isn't optimal.
- **Feedback** = a feedback loop is a mechanism for establishing a dynamic balance in the system. (thermostat)

2 Information Systems building blocks

2.1 Front and Back IS

- **Front-office IS** support business functions that reach out to customers (marketing, sales, customer management).
- **Back-office IS** support internal business operations and interact with suppliers (human resources, financial management, manufacturing, inventory control).

2.2 Information System Applications

1. **Transaction processing systems** are IS applications that capture and process data about business transactions. This includes data maintenance, which provides for updates to stored data.
2. **Management information system** (MIS) is an IS application that provides for management-oriented reporting. These reports are generated on a predetermined schedule and appear in prearranged format.
3. **Decision support system** (DSS) is an IS application that provides its users with decision-oriented information whenever a decision-making situation arises.
    - **Data warehouse** is a read-only informational database that is populated with detailed summary and information generated by other transaction and management systems. It can be accessed by end-users and managers with DSS tools that generate a limitless amount of information in support of unstructured decisions.
4. **Expert system** is a programmed decision-making IS that captures and reproduces the knowledge and expertise of an expert problem solver or decision maker and simulates the thinking of an expert. Expert systems are implemented with artificial intelligence technology that captures, stores and provides access to the knowledge of the experts.
5. **Office automation systems** support the wide range of business office activities that provide for improved workflow and communication between workers. **Personal information systems** are designed to meet the needs of a single user and boost their productivity. **Work group information systems** are designed to meet the needs of a group and boost group productivity.

2.3 Information Systems Architecture

Information systems architecture provides a unifying framework into which various people with different perspectives can organize and view the main building blocks of IS.

Stakeholders:

1. **System owners** pay for the system to be built and maintained.
2. **System users** use the system to perform or support the work to be completed.
3. **System designers** design the system to meet the users' requirements.
4. **System builders** construct, test and deliver the system into operation.
5. **System analysts** provide development of information communications gap between user owners and users and designers and builders.

Goals:

1. **Improving business knowledge**:
   - IS owner is interested in information that adds new business knowledge.
   - IS users know about the data that describes the business.. This data is used to create information and more business knowledge.
   - IS designers deal with database technology that will be used to support business knowledge.
   - IS builders focus on the actual database management system technology to store business data that will support business knowledge.
2. **Improving business processes**:
   - IS owner is interested in the business functions and processes that support a business.
   - IS users specify the business process in terms of requirements for the new system. Those process requirements are policies (explicit rules that need to be followed) and procedures (steps to be followed in completing business processes).
   - IS designers - many companies purchase commercial software instead of building their own.
   - IS builders focus on custom-built applications that automate business processes.
3. **Improving business communications**:
   - IS owner defines the communication scope of IS development project.
   - IS users view communications in terms of IS inputs and outputs.
   - IS designers deal with technical design for user and system-to-system communication interfaces.
   - IS builders deal with technology to implement the communication interfaces.

3 Project Management

- A **project** is a sequence of unique, complex and connected activities that have one goal or purpose that has to be completed in a specific time, within budget and according to specification. A project is successful if: the resulting IS is acceptable to the customer, the system was delivered on time and within budget and the development process had a minimal impact on ongoing business operations.
- **Project management** is the process of scoping, planning, staffing, organizing, directing and controlling the development of a system at a minimum cost within a specific time frame.

3.1 Project Management Activities

1. **Negotiate Scope (Scoping) - Scope** defines the boundaries of a project. What part of the business should be studied, analyzed, designed, constructed, implemented and improved. **A statement of work** is a description of the work to be performed as a part of the project.
2. **Identify Tasks (Planning) - A work breakdown structure** is a hierarchical decomposition of a project into phases, activities and tasks. **Milestones** are events that signify the completion of major deliverables during a project.
3. **Estimate task duration (Estimating)**
   - Estimate the minimum amount of time to perform a task. This is optimistic duration OD.
   - Estimate the maximum amount of time to perform a task. This is pessimistic duration PD.
   - Estimate the expected duration of time to perform a task. This is expected duration ED.
   - The most likely duration is calculated as D = OD + 4*ED + PD / 6
4. **Specify Inter-task Dependencies (Scheduling)**
   - Finish to start - the finish of one task triggers start of another
   - Start to start - start of one task triggers start of another
   - Finish to finish - two tasks must finish at the same time
   - Start to finish - the start of one task triggers finish of another
5. **Assign Resources**
   - People - system owners, users, analysts, designers, builders...
   - Services - such as quality review that may be charged on a per use basis

- Facilities and equipment - rooms and technologies needed to complete a project
- Supplies and materials - pencils, notebooks...
- Money - all of the above translated into a budget

Resource leveling is a strategy that is used to correct resource overallocations by some combination of delaying or splitting tasks. There are two techniques: task delaying and task splitting.

- The **critical path** for a project is a sequence of dependent tasks that last the longest time. It determines the earliest possible completion date for a project. Tasks on a critical path can't be delayed without delaying the whole project, but they can be split.
- The **slack time** for any non-critical task is the amount of delay that can be tolerated between starting and completing a task without causing a delay in completion time of the entire project. Tasks that have slack time can be delayed.

6. **Direct the Team Effort**
7. **Monitor and Control Progress**

3.2 Project Management Tools and Techniques

- **PERT (Project Evaluation and Review Technique)** chart is a graphical network model that shows project tasks and the relationships between those tasks.
- **Gantt chart** is a horizontal bar chart that shows tasks in a calendar. Each bar is a project task.

4 System Analysis

- **Systems analysis** is a problem-solving technique that decomposes a system into component pieces to study how well those component parts work and interact to accomplish their purpose.
- **Systems design** is a problem-solving technique that complements systems analysis and it reassembles system's component pieces back into a complete system, hopefully an improved one. This involves deleting, adding, changing pieces from the original system.

Steps in system analysis:

1. Scope definition
2. Problem analysis

3. Requirements analysis
4. Logical design
5. Decision analysis

4.1 System Analysis Methods

1. **Model-driven Analysis Methods** emphasizes the drawing of pictorial system models to document and validate existing and proposed systems. This becomes the blueprint for designing and constructing an improved system. A **model** is a representation of either reality or vision.
    - **Structured analysis - Data flow diagrams**
    - **Information engineering - ER diagrams**
    - **Object-oriented analysis - Unified Modeling Language UML**
2. **Accelerated Analysis Methods**
    - **Discovery prototyping** is used to identify the users' business requirements by having them to a quickly and dirty implementation of those requirements.
    - **Reverse engineering** technology reads the program code for a database, application program and user interface and it generates the equivalent system model.
3. **Requirements Discovery Methods** include techniques to be used by system analysts to identify and extract system problems and solution requirements from the user community.
    - **Joint requirements planning (JRP)** techniques use workshops to bring together the stakeholders to perform system analysis.
    - **Fact-finding** is a set of techniques used to collect information about system problems, solutions, opportunities and priorities. Those include sampling, research, observation, questionnaires/surveys/interviews.

5 Data Modeling and Analysis

- A model is a representation of reality.
- **Logical models** show what a system is or does. They are implementation independent which means that they depict the system independent of any technical implementation.
- **Physical models** show not only what a system is or does, but also how the system is physically and technically implemented.

5.1 Data Modeling

Data modeling is a technique for organizing and documenting a systems' data. Data model is implemented as a database and that's why it's also called database modeling. The actual model is called entity relationship diagram ERD because it depicts data in terms of entities and relationships.

Design process: world -¿ mental model -¿ conceptual model -¿ logical model -¿ physical model -¿ database

5.2 Data Analysis and Normalization

Data analysis is a process that prepares data model for implementation as a simple, nonredundant, flexible and adaptable database. This technique is called normalization. **Normalization** is a data analysis technique that organizes data attributes such that they are grouped to form nonredundant, stable, flexible and adaptive entities.

- **1NF** - a logical data model is in 1st normal form (1NF) if it doesn't have attributes that have repeating values for a single instance of an entity. Every attribute in an entity has to have only one value per instance for the model to pass 1NF.
- **2NF** - requires data to be in 1NF and data model leads to entities containing attributes that are dependent on the whole identifier.
- **3NF** - model needs to be in 1NF and 2NF and in the entities none of the attributes are dependent on the nonidentifier attribute.

TUTORIAL NOTES

git checkout master -> changing to master branch
git log --oneline --decorate --graph --all -> See commit history
git status -> see which branch + is there something to commit
git branch -> see all branches
git branch xyz -> create xyz branch
----PUTTING XYZ CHANGES ON MASTER----
git checkout master
git merge xyz
git branch -d xyz -> we can remove xyz branch
---
git fetch <remote> -> syncronize with remote
git pull -> contains both fetch and merge.
git checkout -b xyz origin/xyz -> create a local branch from a remote branch
git checkout -u origin/xyz -> setup to track remote xyz from origin
git branch -vv -> see tracking branches you set up.
----------PHP----------
$x = 10; -> variable
echo $x+$y;
echo <h1> . $x . </h1>
---------Print Toyota---------
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo $cars[2] //print toyota
?>
superglobals => They are always accessible from any class or function.
-------NODEJS / REACT--------
DOM = document object model - standard object model and programming interfaces of HTML for how to get, change, add, or delete HTML elements.
"const" is a signal that the variable won't be reassigned.
In JavaScript, class methods are not bound by default. If you don't want undefined when calling, bind. [For ex: this.handleClick.bind (this);]
Slug =>a lowercase version of the page title, with any spaces removed.
$array = array("foo" => "bar", "bar" => "foo",); [comma-seperated key => value pairs]

```
sglbl@SglblPC: /mnt/c/Users/Sglbl/Documents/Özel Office Şablonları
sglbl@SglblPC:/mnt/c/Users/Sglbl/Documents/Özel Office Şablonları$ git clone https://gitlab.com/sglbl/my-website.git   putting from gitlab to pc
sglbl@SglblPC:/mnt/c/Users/Sglbl/Documents/Özel Office Şablonları$ git init .      it means you will start to this git thing to do
sglbl@SglblPC:/mnt/c/Users/Sglbl/Documents/Özel Office Şablonları$ touch index.html  create file
sglbl@SglblPC:/mnt/c/Users/Sglbl/Documents/Özel Office Şablonları$ nano index.html   edit file
sglbl@SglblPC:/mnt/c/Users/Sglbl/Documents/Özel Office Şablonları$ git add .      registering to git system
sglbl@SglblPC:/mnt/c/Users/Sglbl/Documents/Özel Office Şablonları$ git commit -am "Here you write the changes you did"
[master (root-commit) 6072a54] Here you write the changes you did      saving the changes on git system
 1 file changed, 1 insertion(+)
 create mode 100644 index.html
sglbl@SglblPC:/mnt/c/Users/Sglbl/Documents/Özel Office Şablonları$ git push -u origin  putting changes on gitlab (gonna ask you username / password )
sglbl@SglblPC:/mnt/c/Users/Sglbl/Documents/Özel Office Şablonları$ git status  seeing status of difference between gitlab
On branch master                                                               and local files
nothing to commit, working tree clean
sglbl@SglblPC:/mnt/c/Users/Sglbl/Documents/Özel Office Şablonları$ git log    seeing history
commit 6072a54cd78943cecbad094185ebe66b94c5bc74 (HEAD -> master)
Author: sglbl <silo1881@gmail.com>
Date:   Fri Oct 22 16:06:29 2021 +0200

    Here you write the changes you did
sglbl@SglblPC:/mnt/c/Users/Sglbl/Documents/Özel Office Şablonları$ git log > taskX.txt   saving history to a file called taskX.txt
```

Push the commit to the remote repository created on github/gitlab.

```
sglbl@SglblPC: /mnt/c/Araclar/GTU/Year3/System3/AbroadGuideForStudents

sglbl@SglblPC:/mnt/c/Araclar/GTU/Year3/System3/AbroadGuideForStudents$ git init
Initialized empty Git repository in /mnt/c/Araclar/GTU/Year3/System3/AbroadGuideForStudents/.git/
sglbl@SglblPC:/mnt/c/Araclar/GTU/Year3/System3/AbroadGuideForStudents$ ^C
sglbl@SglblPC:/mnt/c/Araclar/GTU/Year3/System3/AbroadGuideForStudents$ git remote add origin https://gitlab.com/sglbl/abroad-guide-for-students.git
sglbl@SglblPC:/mnt/c/Araclar/GTU/Year3/System3/AbroadGuideForStudents$ git add .
sglbl@SglblPC:/mnt/c/Araclar/GTU/Year3/System3/AbroadGuideForStudents$ git commit -m "Initial commit"
sglbl@SglblPC:/mnt/c/Araclar/GTU/Year3/System3/AbroadGuideForStudents$  git push --set-upstream origin master
Username for 'https://gitlab.com': sglbl
Password for 'https://sglbl@gitlab.com':
```

If you don't have a git repo already initiate one with `git init`

1. `git remote -v` Check if any remote already exists
2. If **Yes** then use `git remote set-url origin git@github.com:User/UserRepo.git` to change the origin
3. If **No** then use `git remote add origin git@github.com:User/UserRepo.git` to set new origin for your repo.
4. and finally use `git push -u origin master` to push your code to remote and add upstream (tracking) reference to your remote branch.

NOTE: If you use `-u` flag, its for **upstream**, it enables you to use simply `git pull` instead of `git pull origin <branch-name>` in upcoming operations.

PHP

```
<!DOCTYPE html>
<html>
<body>

<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>

</body>
</html>
```

Learn PHP

Study PHP at W3Schools.com

9

5+4

concatanate

PHP Array

```
<!DOCTYPE html>
<html>
<body>

<?php
$cars =
array("Volvo","BMW","Toyota");
var_dump($cars);
?>

</body>
</html>
```

array(3) { [0]=> string(5) "Volvo" [1]=> string(3) "BMW" [2]=> string(6) "Toyota" }

if you want to print Toyota echo $cars[2]

PHP Global Variables – Superglobals $_PSOT

```
<!DOCTYPE html>
<html>
<body>

<form method="post" action="<?php echo
$_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // collect value of input field
    $name = $_POST['fname'];
    if (empty($name)) {
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>

</body>
</html>
```

Name: Submit Query

123

https://www.studenti.famnit.upr.si/~76210123/week5/intro.php?name=suleyman

Google Facebook You Tube Çeviri Wp TrAlt Tureng Moodle OBS file

Hellosuleyman PHP

array(1) { ["name"]=> string(8) "suleyman" } Test $GET

result by name

www.studenti.famnit.upr.si - PuTTY

```
 GNU nano 2.5.3                File: intro.php

<html>
<body>

<?php
echo "<h1> Hello ". $_GET["name"]  .  " PHP </h1>";
var_dump($_GET);
 ?>
<a href="test_get.php?subject=PHP&web=W3schools.com">Test $GET<

</body>
</html>
```