Anatomy of Requirements

UQAM – 19/02/2020

@SmartModelTeam

https://github.com/smart-researchteam

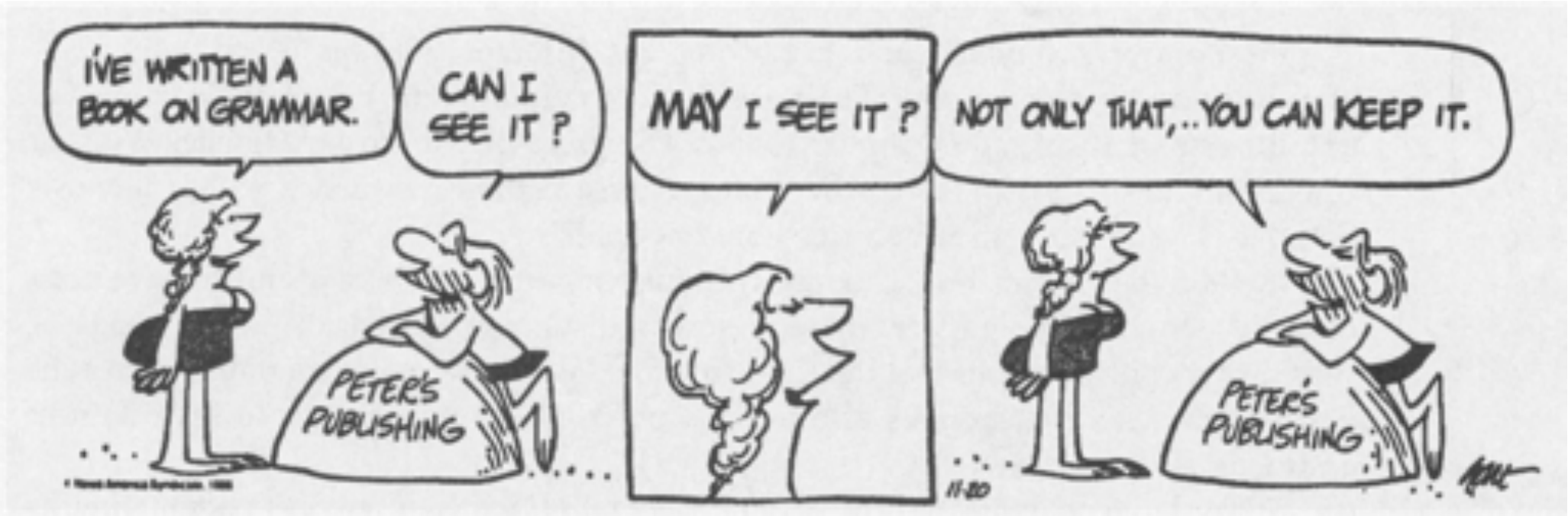
Outline

- Context
- Categories of requirements
- Categories of inter-requirements relations

http://bit.ly/reqs-anatomy

Disclaimer

We are not prescriptive!



https://noharmspilt.com/2015/10/09/descriptive-vs-prescriptive-grammar/

Outline

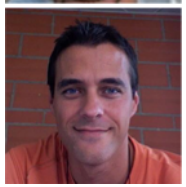
- Context
- Categories of requirements
- Categories of inter-requirements relations

Context

Joint effort...

- Innopolis University
  - Alexandr
  - Bertrand

- IRIT/SM@RT team
  - Florian
  - Sophie
  - JMB



IEEE/SWEBOK/ISO definition of a Requirement

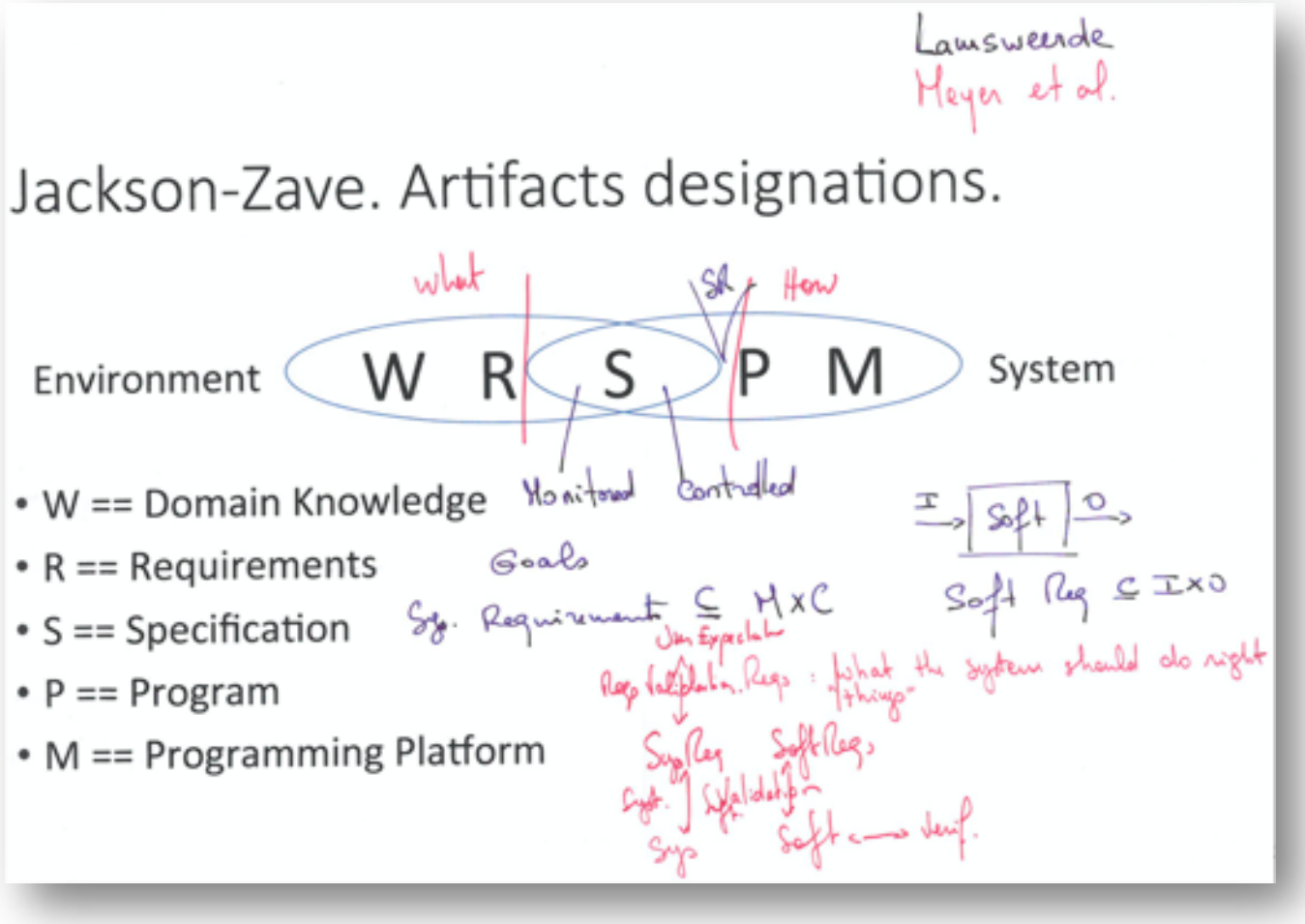
“A **1.1 Definition of a Software Requirement**

At its most basic, a software requirement is a property that must be exhibited by something in order to solve some problem in the real world. It may aim to automate part of a task for someone to support the business processes of an organization, to correct shortcomings of existing software, or to control a device—to name just a few of the many problems for which software solutions are possible. The ways in which users, business processes, and devices function are typically complex. By extension, therefore, the requirements on particular software are typically a complex combination from various people at different levels of an organization, and who are in one way or another involved or connected with this feature from the environment in which the software will operate.

”

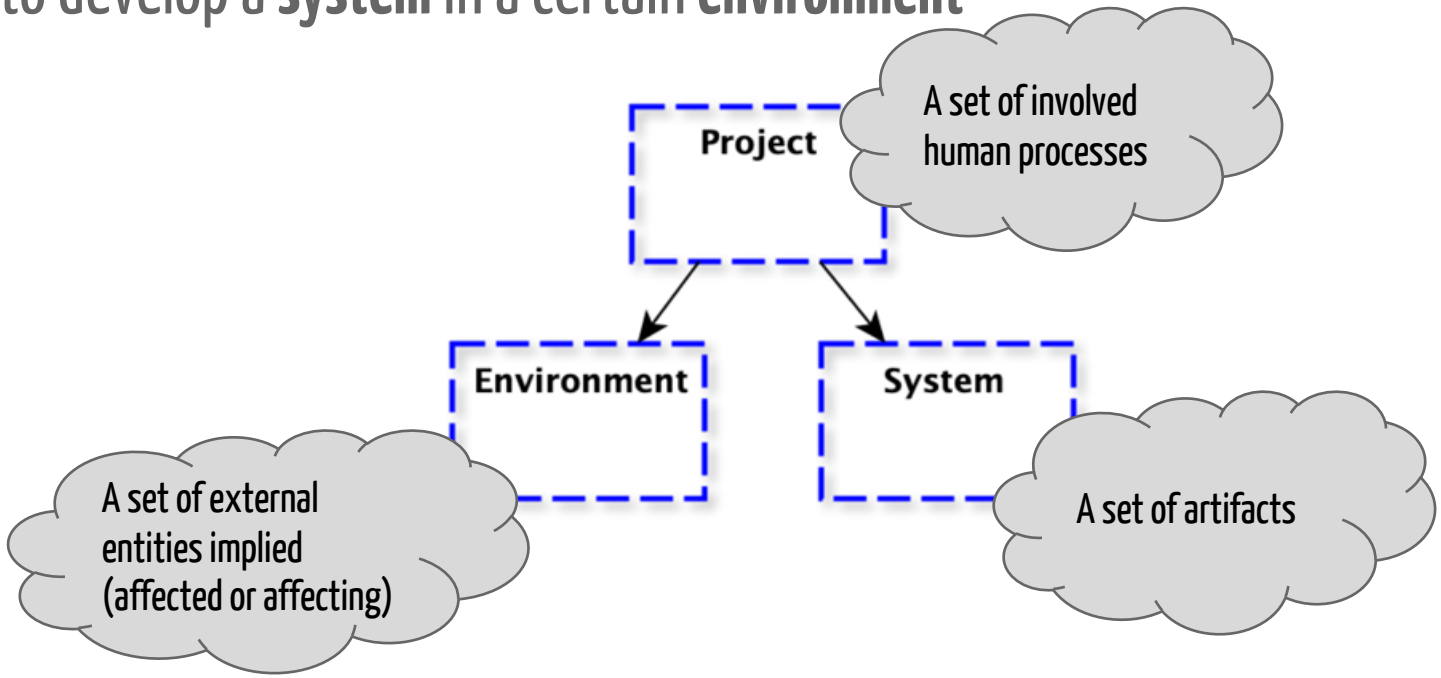
http://swebokwiki.org/Chapter_1:_Software_Requirements

Context (universe of discourse)



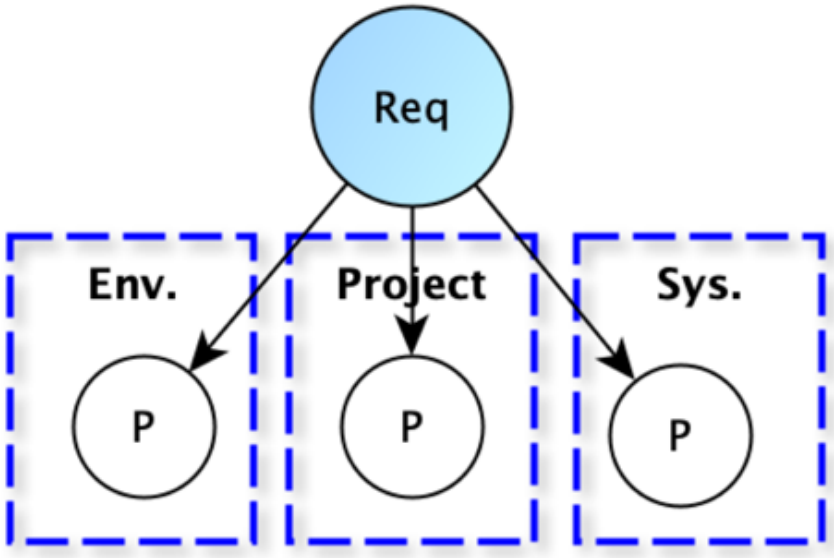
Context (universe of discourse)

"a **project** to develop a **system** in a certain **environment**"



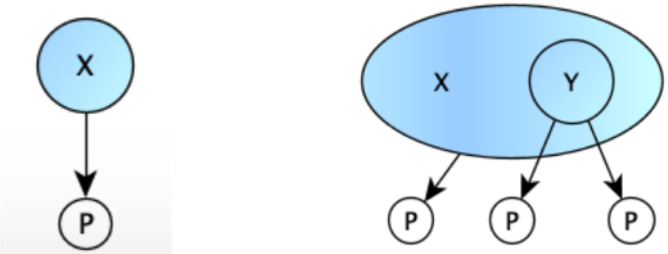
General definition of a Requirement

"A requirement is a **statement** of a relevant project, system or environment **property**"



Requirement

Can be **Atomic** or **Composite**



The **type** of a requirement is the notation in which it is expressed (English, SysML, Eiffel, ...)

"The LGS has three components."

Some basic concepts

Property: boolean predicate (on a project, system or environment)

Statement: human-readable expression of a property

Note: A property or statement can be atomic or composite (homogeneous or heterogeneous)

Additional concepts

We distinguish the different stage of a System:

- The system itself (mainly to talk about its components)
- The running system (mainly to talk about its behavior)
- The system in development (mainly to talk about phases and artifacts)

Outline

- Context
- Categories of requirements
- Categories of inter-requirements relations

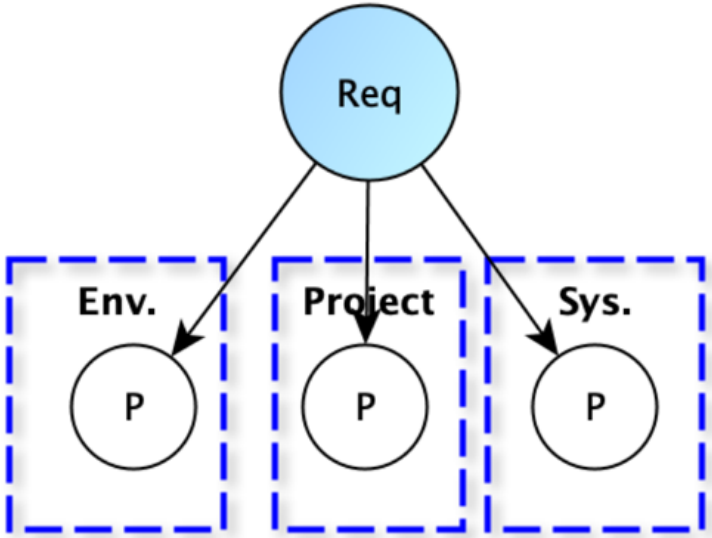
Categories of requirements

Categories of requirements (main)

- Component
- Goal
- Behaviour
- Task
- Product
- Constraint
- Role
- Limit
- Meta-requirement

Goal

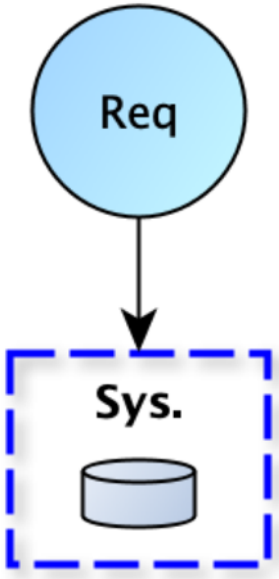
(an objective of the project or the system, in terms of their effect on some entity external to them)



"The goal of the system is to allow any user to book a flight."

Component

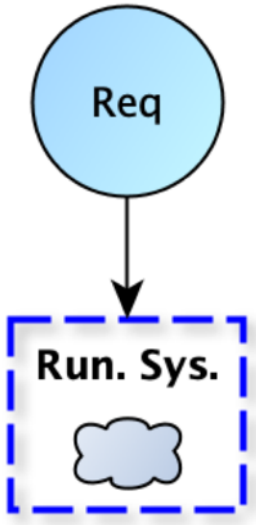
(the property that the system, project or environment includes a certain part)



"The Landing Gear System is composed of three parts."

Behavior

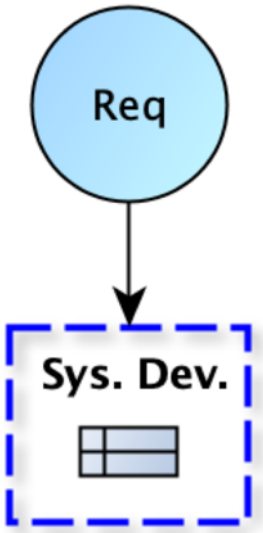
(a property of the effects of the operation of the system or some of its components)



"The system should allow to open and close the door safely."

Task

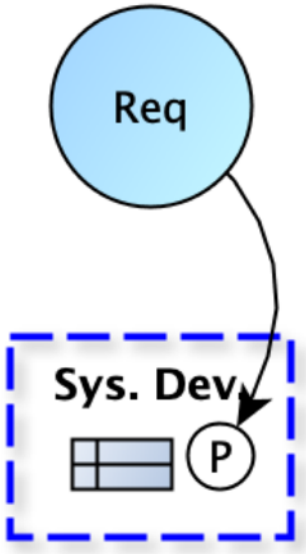
(the property that the project includes a certain activity)



"The team should meet in a daily basis, called daily meeting."

Product

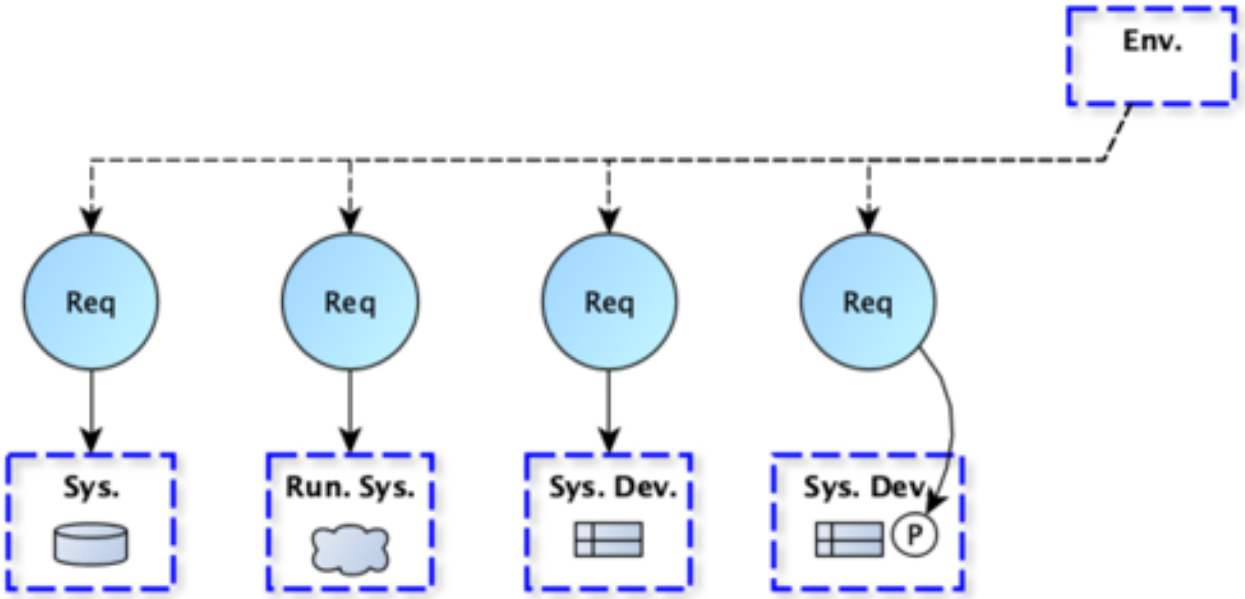
(the property that a task uses or produces a material or virtual object)



"The following test plan is provided:..."

Constraint

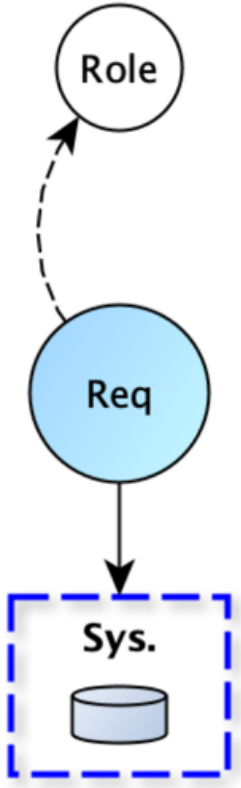
(an environment property that may affect components, goals, behaviors, tasks or products)



"Every transfer over 10.000€ requires an authorization."

Role

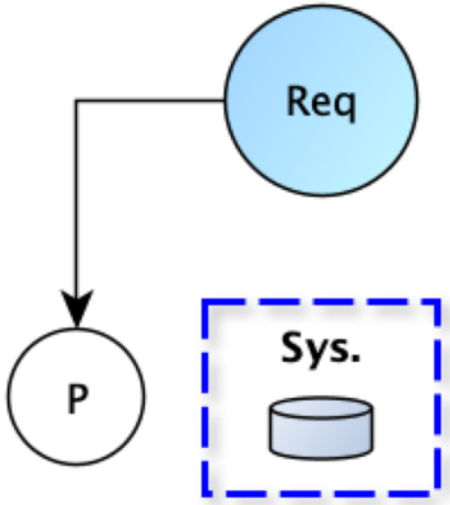
(the property that a component carries some or all of the responsibility for a behavior or task)



"Authorizations are provided by the head of the audit department."

Limit

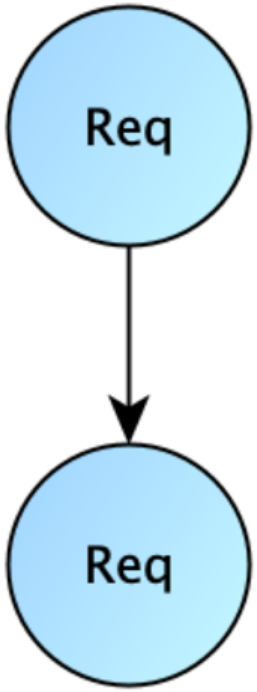
(the property that the project, system or environment does *not* include a requirement of any of the preceding kinds)



"Integration testing will be performed in a follow-up project."

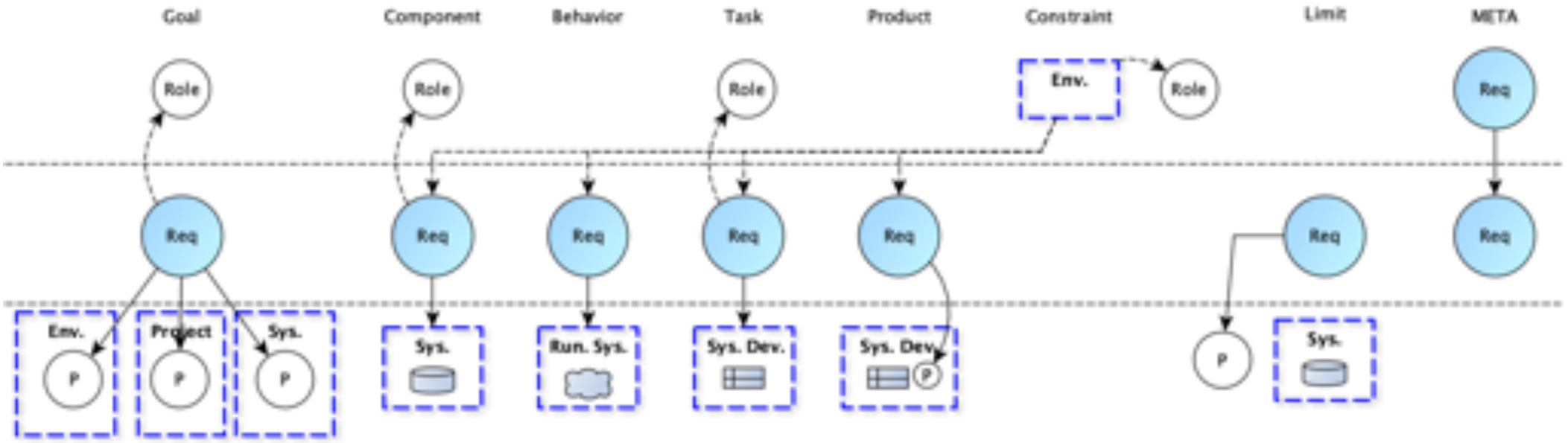
Meta-requirement

(a property of requirements themselves)



"The details are provided in Fig. 7."

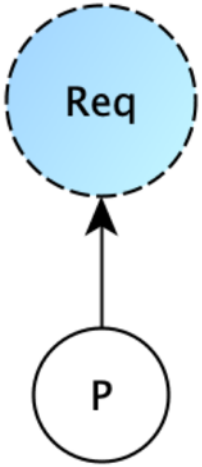
Classification (overview)

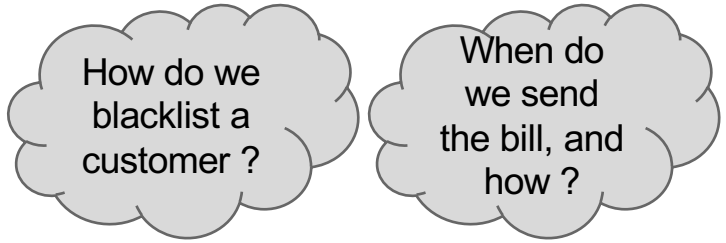


Missing requirements (to be discussed)

Might be useful to identify them.

Category of properties that were not covered by any requirement.





"The system should send the bill to the non blacklisted customers."

Categories of requirements (derived)

- Actor (from Component)
- Justification (from Meta)
- Responsibility (from Role)
- Assumption (from Constraint)
- Obstacle (from Goal)
- Obligation (from Constraint)
- Invariant (from Assumption & Behavior)
- Functional (from Behavior)
- Non-Functional (from Behavior)

Outline

- Context
- Categories of requirements
- Categories of inter-requirements relations

Quizz

Quizz

http://bit.ly/jmb-quizz-classifications

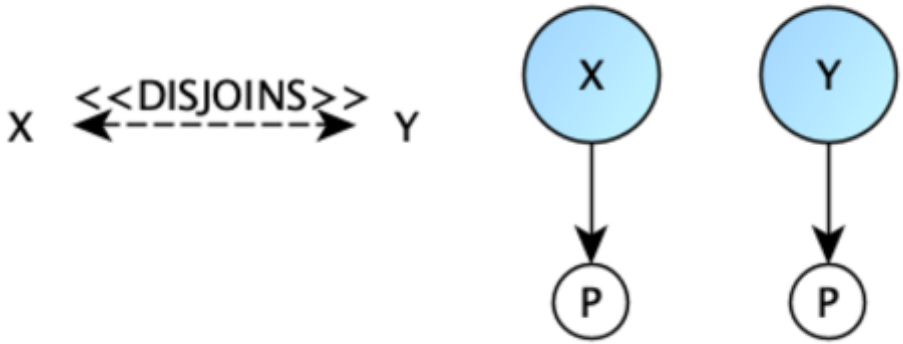
Categories of inter-requirements relations

Relations between requirements

- Disjoins (X || Y)
- Belongs (X ⊆ Y)
- Repeats (X ⇔ Y)
- Contradicts (X ⊕ Y)
- Extends (X > Y)
- Excepts (X \\ Y)
- Constraints (X Y)
- Characterizes (X → Y)

X and Y are unrelated

X || Y

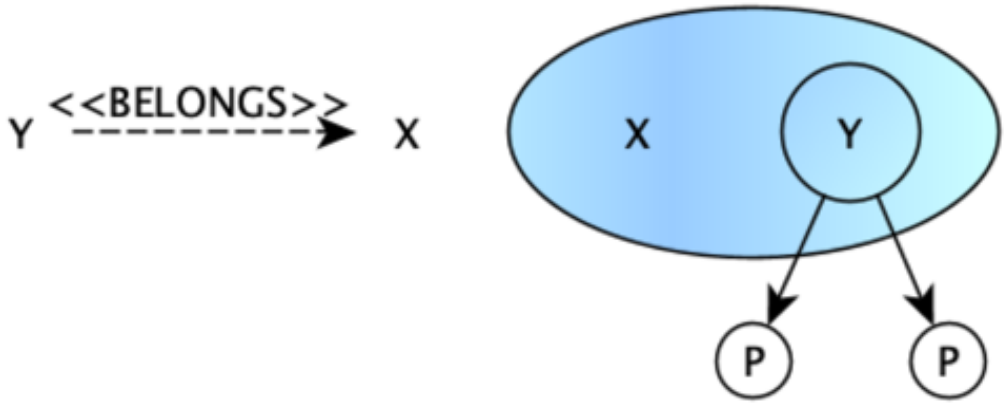


"The system is composed of three components."

"The car should be as economic in fuel consumption as possible."

Y is a sub-requirement of X

$Y \subseteq X$

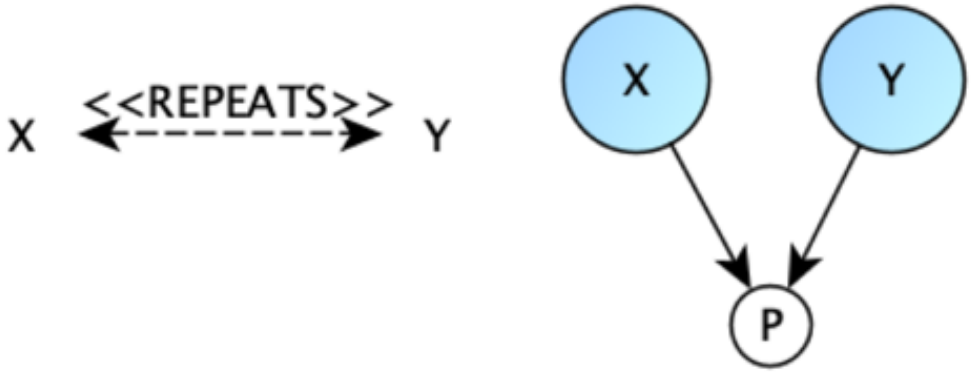


"4.3. System Externals"

"A customer is any user of the system that has not identified himself as an SBE employee."

X specifies the same property as Y

$X \Leftrightarrow Y$

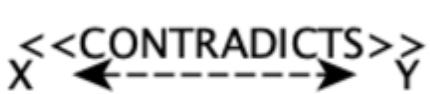


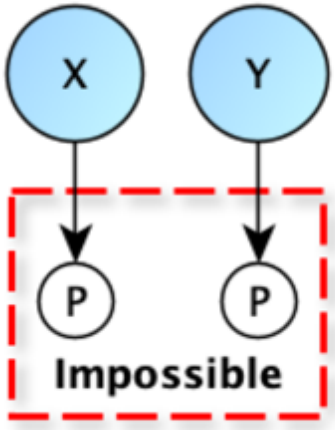
"The system is composed of three components."

"Here are the descriptions of the three parts of the system:"

X specifies a property in a way not compatible with Y

$X \oplus Y$



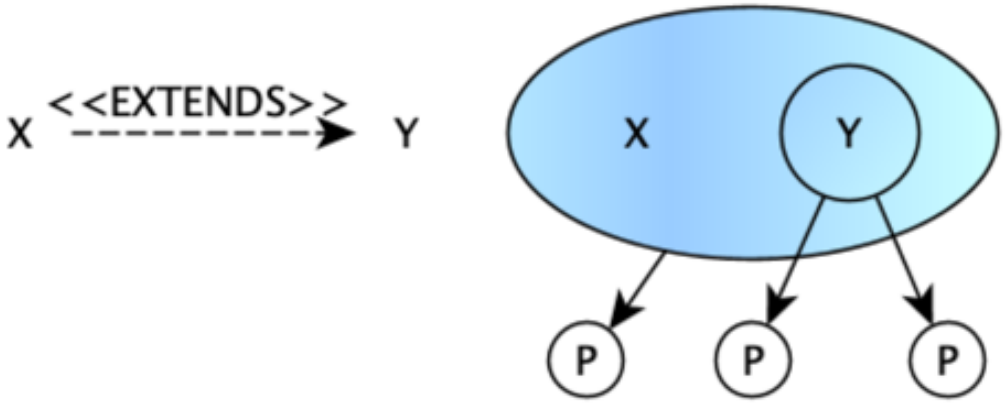


"The car should be as fast as possible."

"The car should be as economic in fuel consumption as possible."

X > Y

X assumes Y and specifies a property not specified by Y

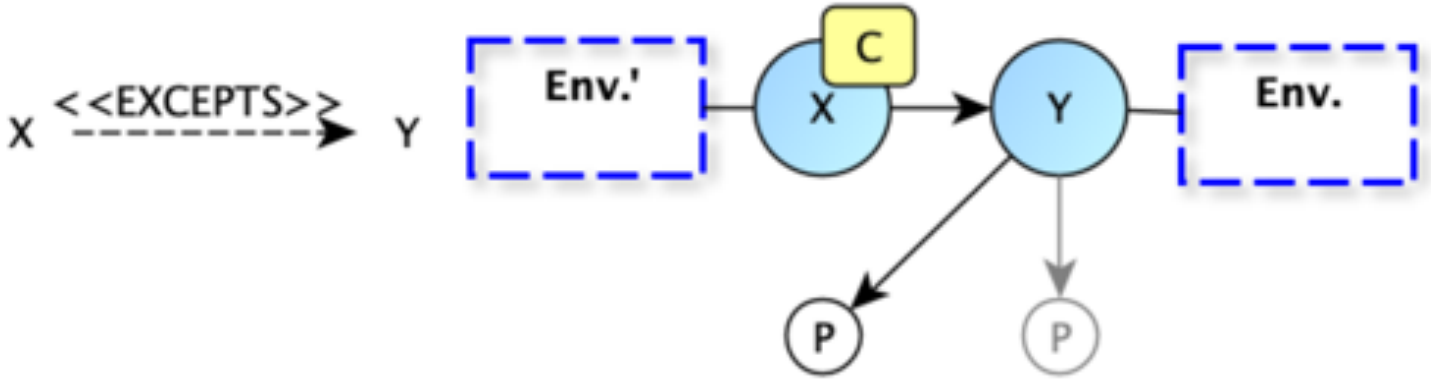


"The system shall provide a search facility that will allow full-text searching of all web pages that the user is permitted to access"

"The system shall allow for online product ordering by either the customer or the sales agent."

X \\ Y

X changes or removes, for a specified case, a property specified by Y

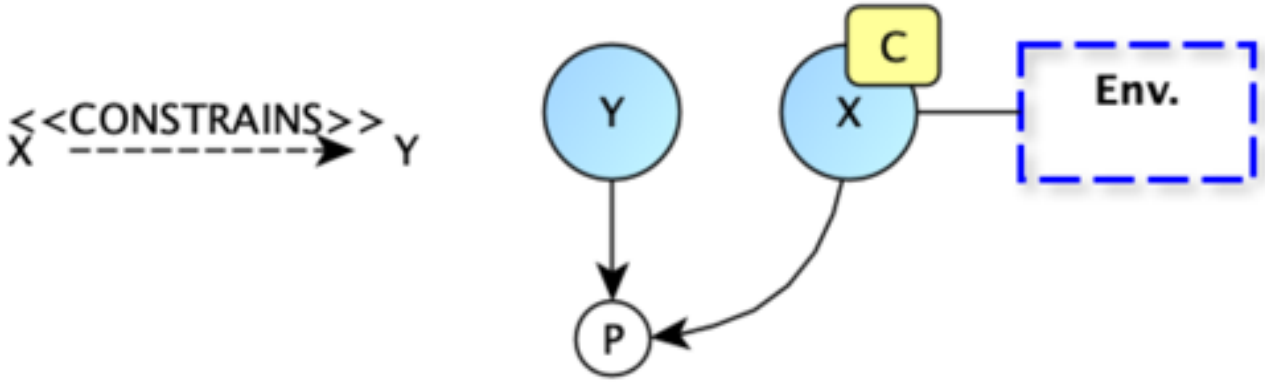


"In case of emergency braking, the system should prevent the wheels from frozing."

"The wheel can be frozen by braking."

X specifies a constraint on a property specified by Y

X Y

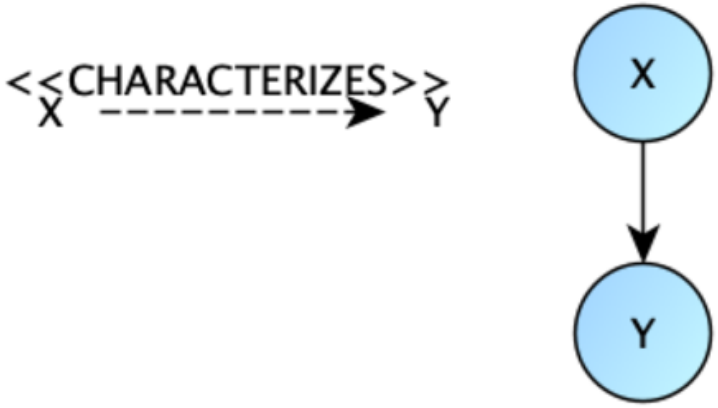


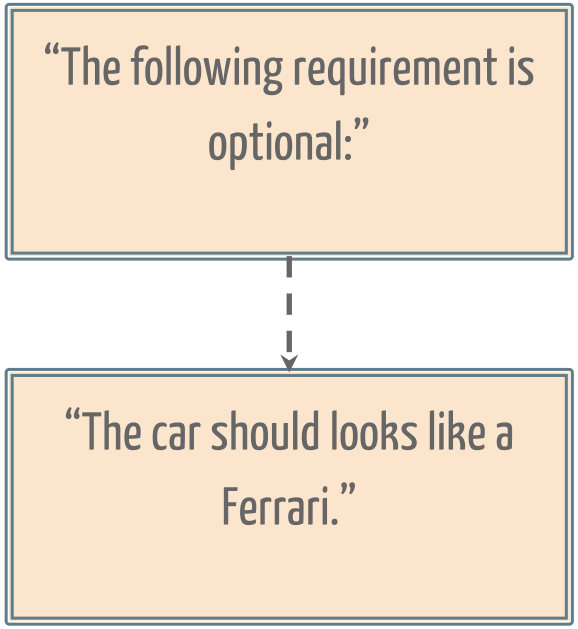
"The user is registered."

"In order to get personalized or restricted information, place orders or do other specialized transactions a user must login so that that the system can determine his access level."

X is a meta-requirement involving Y

$$X \longrightarrow Y$$

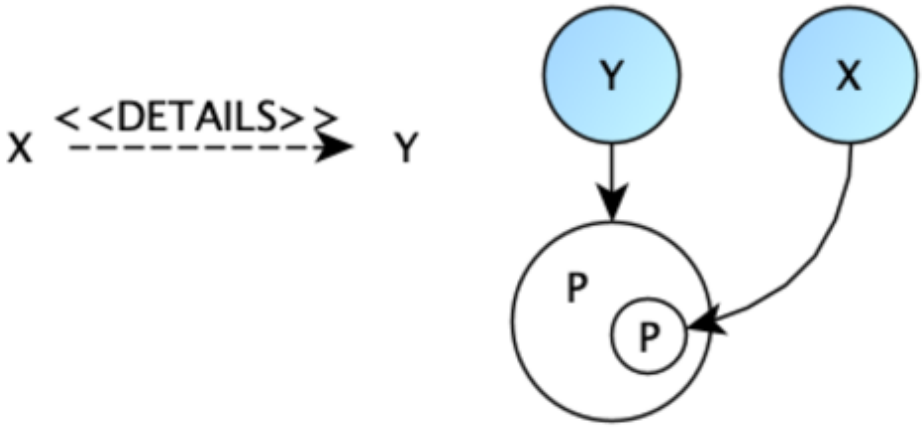




Derived (but useful) relations

X adds detail to a property specified by X

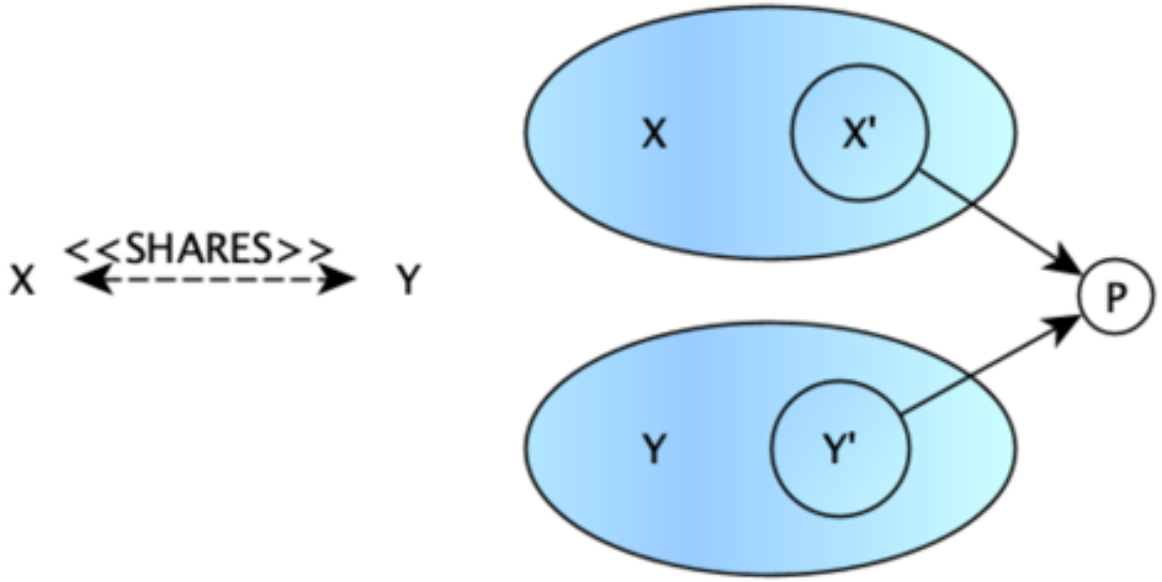
X » Y



"The hot water should be between 27°C and 37°C." - - - → "The shower should deliver hot water."

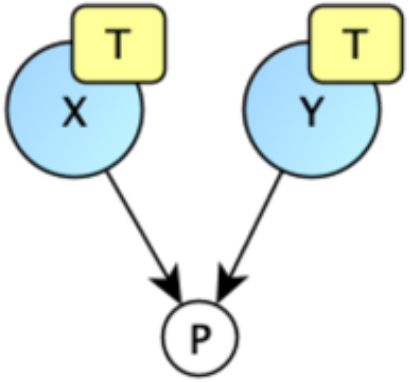
$X \cap Y$

X' ⇔ Y' for some sub-requirements X' and Y' of X and Y



$X \Leftrightarrow Y$, and X has the same type as Y

$X \equiv Y$

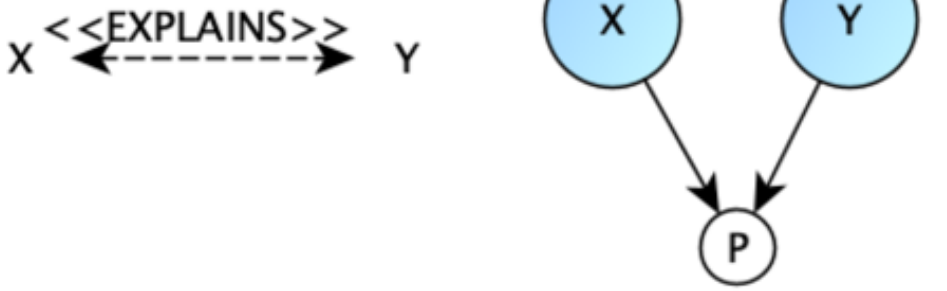


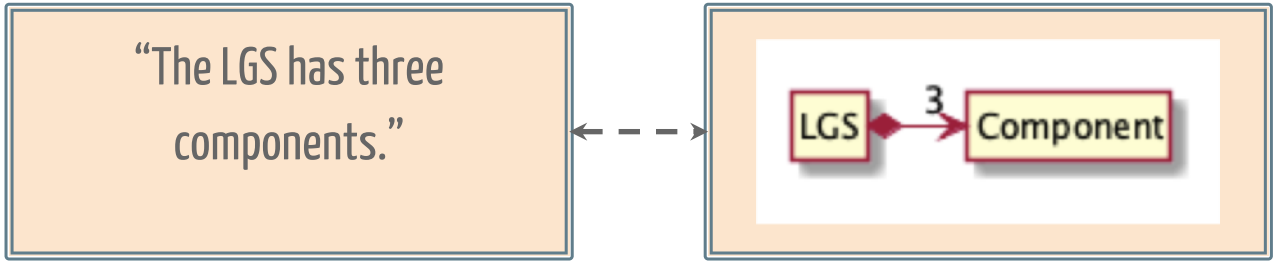
"The system is composed of three components."

"Here are the descriptions of the three parts of the system:"

$X \Leftrightarrow Y$, and X has a different type from Y

$X \cong Y$





What are the benefits ?

Examples of possible prescriptions

No **Duplicates**

Few **Excepts**

...

Contributions

Clarification of reqs concepts

Basic for reqs methodology

Basics for critical analysis of reqs docs

Basis for NLP

...

Quizz

Quizz

http://bit.ly/jmb-quiz-relations

The end...

@SmartModelTeam

https://github.com/smart-researchteam