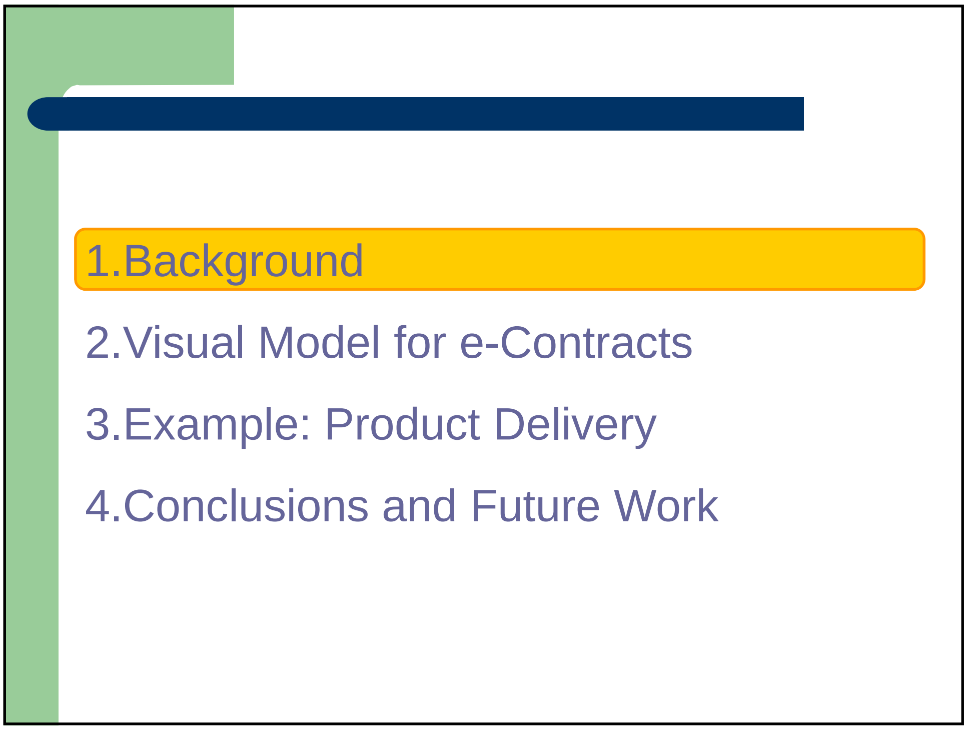
Visual Specification of Timed Contracts

Enrique Martínez

emartinez@dsi.uclm.es

Universidad de Castilla-La Mancha. SPAIN

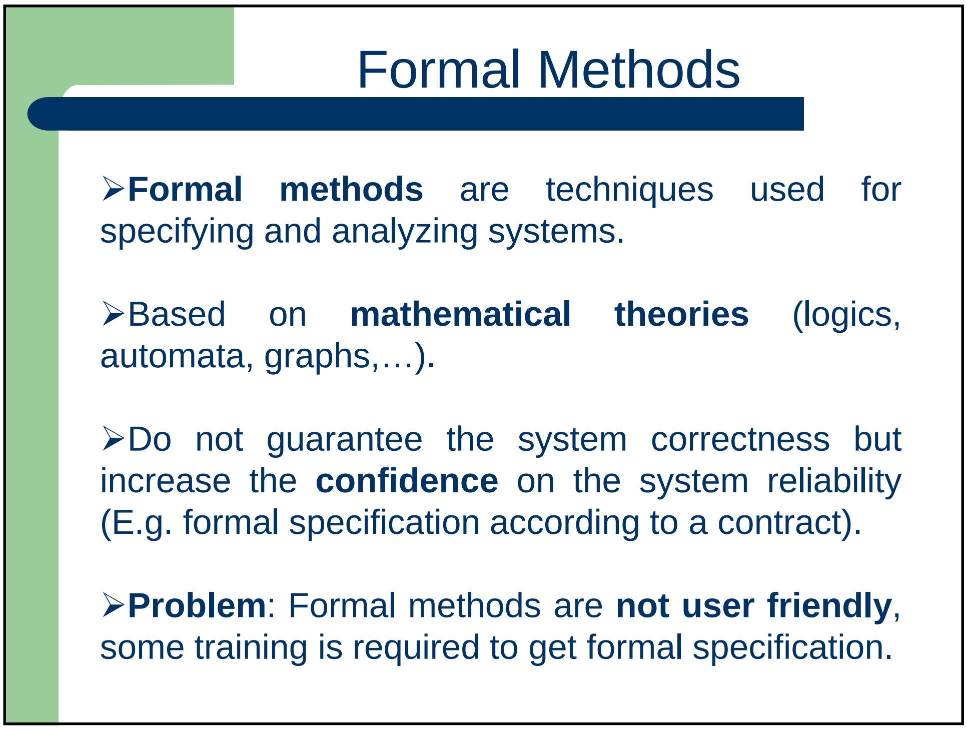


1.Background

2.Visual Model for e-Contracts

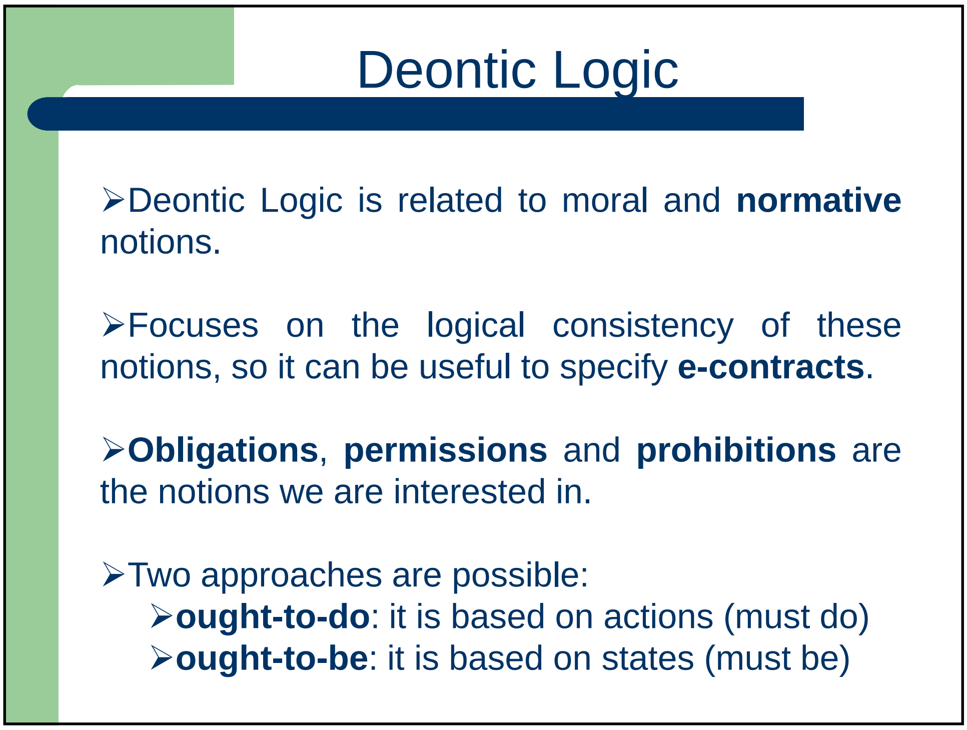
3.Example: Product Delivery

4.Conclusions and Future Work



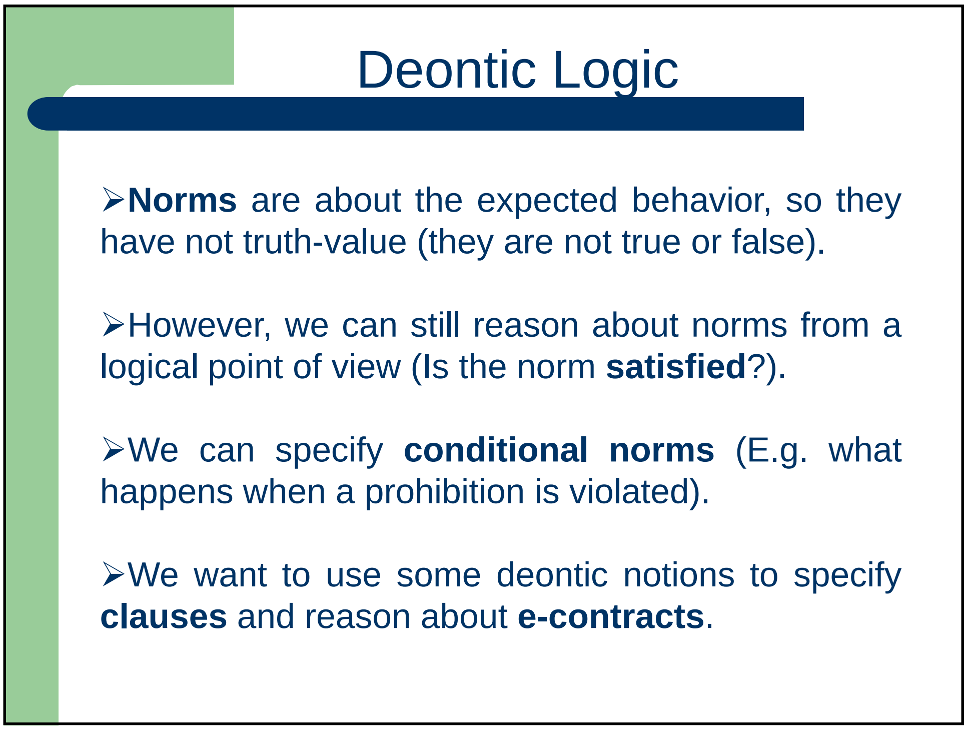
Formal Methods

- **Formal methods** are techniques used for specifying and analyzing systems.
- Based on **mathematical theories** (logics, automata, graphs,…).
- Do not guarantee the system correctness but increase the **confidence** on the system reliability (E.g. formal specification according to a contract).
- **Problem**: Formal methods are **not user friendly**, some training is required to get formal specification.



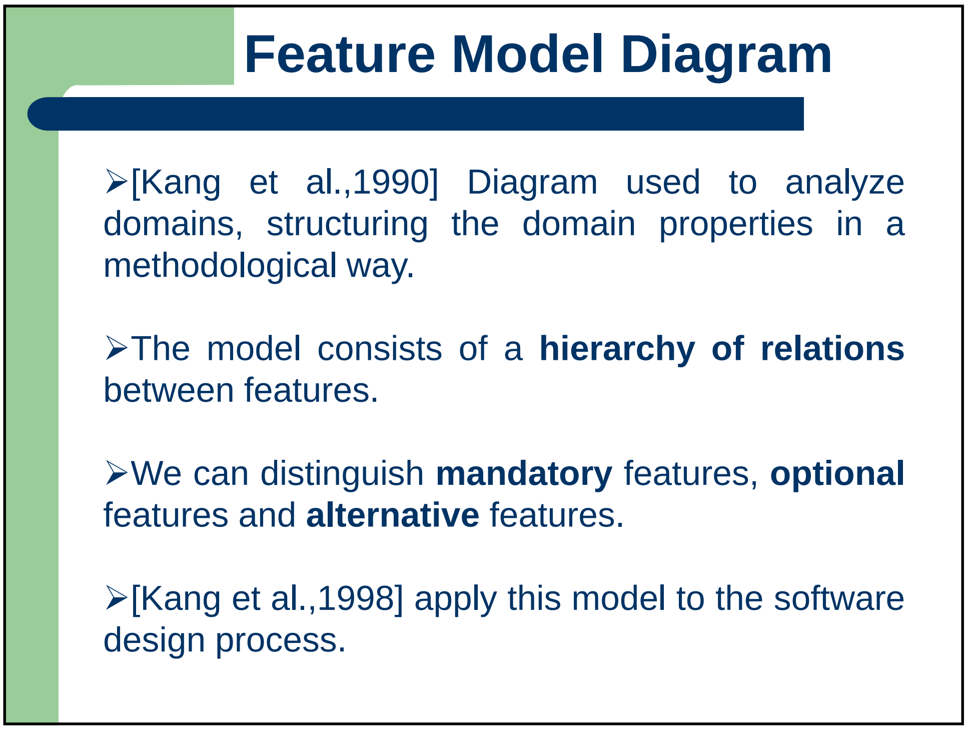
Deontic Logic

- Deontic Logic is related to moral and **normative** notions.
- Focuses on the logical consistency of these notions, so it can be useful to specify **e-contracts**.
- **Obligations**, **permissions** and **prohibitions** are the notions we are interested in.
- Two approaches are possible:
  - **ought-to-do**: it is based on actions (must do)
  - **ought-to-be**: it is based on states (must be)



Deontic Logic

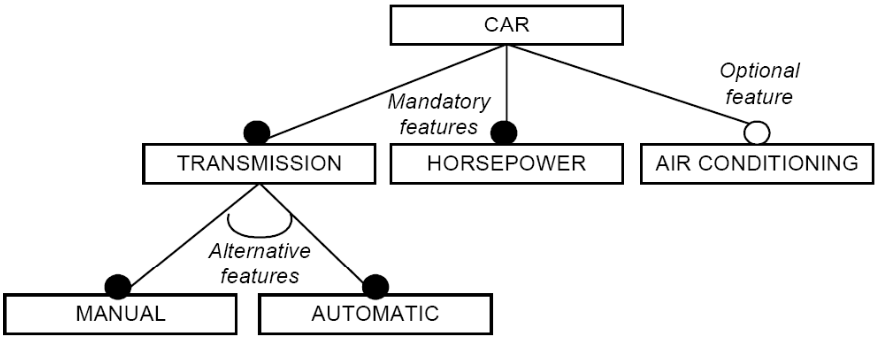
- **Norms** are about the expected behavior, so they have not truth-value (they are not true or false).
- However, we can still reason about norms from a logical point of view (Is the norm **satisfied**?).
- We can specify **conditional norms** (E.g. what happens when a prohibition is violated).
- We want to use some deontic notions to specify **clauses** and reason about **e-contracts**.



Feature Model Diagram

- [Kang et al.,1990] Diagram used to analyze domains, structuring the domain properties in a methodological way.
- The model consists of a **hierarchy of relations** between features.
- We can distinguish **mandatory** features, **optional** features and **alternative** features.
- [Kang et al.,1998] apply this model to the software design process.

Feature Model Diagram



Example: Feature Model for a Car

Feature Model Diagram

➢[Czarnecki et al.,2000] use features for generative programming, modeling the commonalty and variability existing in the domain.

➢[Riebisch et al.,2002] extend this diagram with multiplicities.

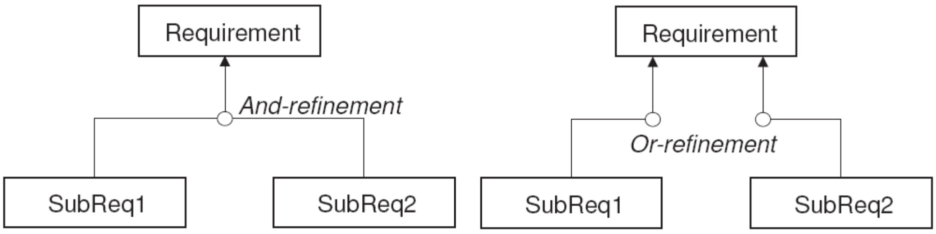
➢[Robak et al.,2003] propose using feature diagrams to model Web Service variability.

➢[Fantinato et al.,2006] describe a feature-based approach to simplify Web Service establishment.

Goal Model Diagram

- This diagram consists of decomposing goals into subgoals through AND/OR refinements.
- AND-refinement means that all the subgoals must be satisfied to satisfy a goal.
- OR-refinement means that at least one subgoal must be satisfied to satisfy a goal.

Goal Model Diagram



Requirement

And-refinement

SubReq1

SubReq2

Requirement

Or-refinement

SubReq1

SubReq2

AND-refinement & OR-refinement

Goal Model Diagram

➢The KAOS methodology [Van Lamsweerde et al. 1993] use this model to analyze the requirements of software systems.

➢The Tropos methodology [Perini et al.,2001] also advocate the use of goal diagrams for requirements analysis.

➢A methodology founded on Tropos for designing Web Services also has been proposed [Lau et al., 2004].

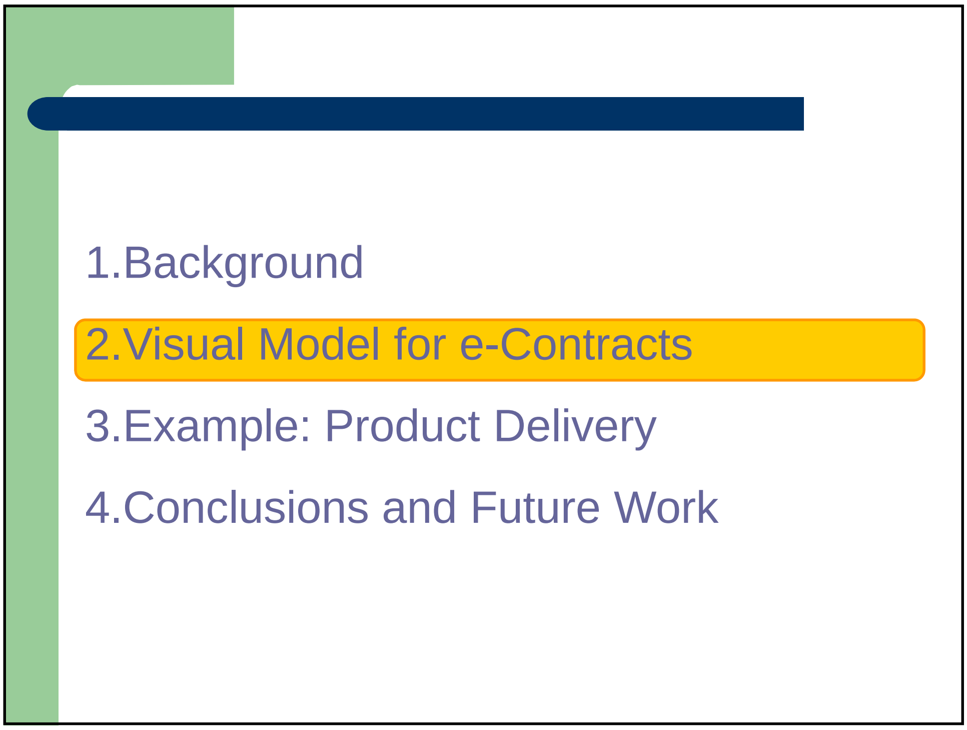
¿What about e-Contracts?

➢We have a set of **clauses** that must be satisfied by the partners of the e-contract.

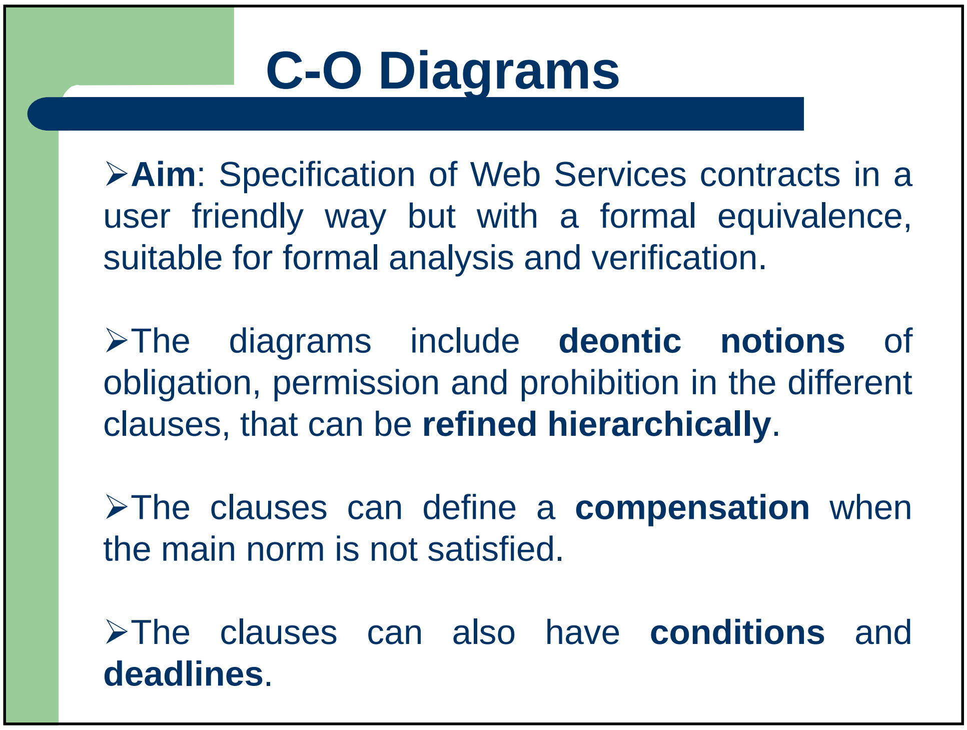
➢These clauses are decomposed into **subclauses** in a hierarchical way.

➢Clauses can also include under which **conditions** are applied and **time restrictions**.

➢A **visual model** similar to feature/goal model can be appropriate to analyze e-contracts, including deontic notions, conditions and time constraints.



1. Background
2. Visual Model for e-Contracts
3. Example: Product Delivery
4. Conclusions and Future Work



C-O Diagrams

➢**Aim**: Specification of Web Services contracts in a user friendly way but with a formal equivalence, suitable for formal analysis and verification.

➢The diagrams include **deontic notions** of obligation, permission and prohibition in the different clauses, that can be **refined hierarchically**.

➢The clauses can define a **compensation** when the main norm is not satisfied.

➢The clauses can also have **conditions** and **deadlines**.

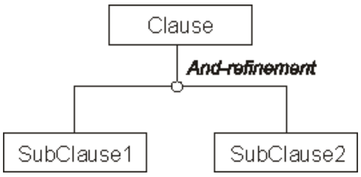
C-O Diagrams

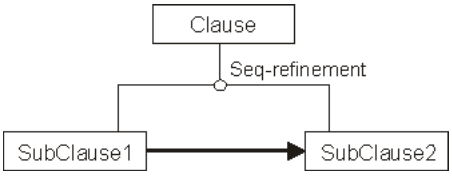
g	P	R
tr		

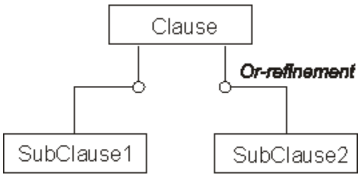
- ➢**g** are the conditions.
- ➢**tr** are the temporal restrictions.
- ➢**P** is the proposition that must be satisfied (Obligation, Permission or prohibition).
- ➢**R** is the reparation/compensation that must be satisfied when **P** is not satisfied.

C-O Diagrams

Three different refinements:







Formal Model

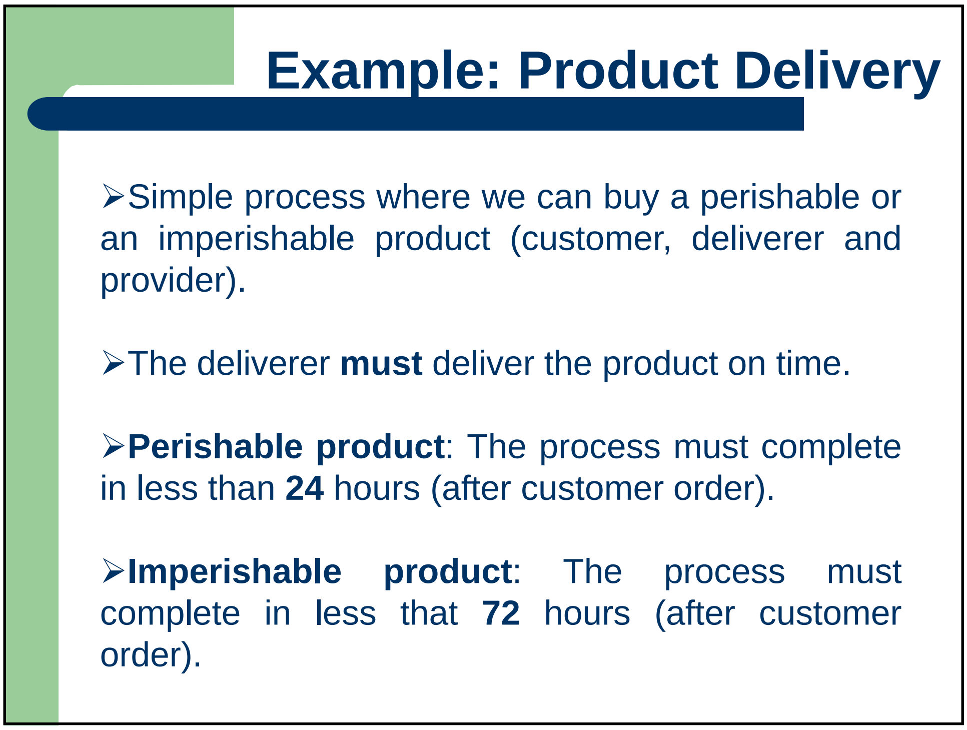
- As we have seen, formal methods increase the **confidence** in our systems.

- We want a **formal specification** equivalent to the visual specification in order to analyze the model.

- We choose **Timed Automata**, because they allow us to specify and verify temporal properties.

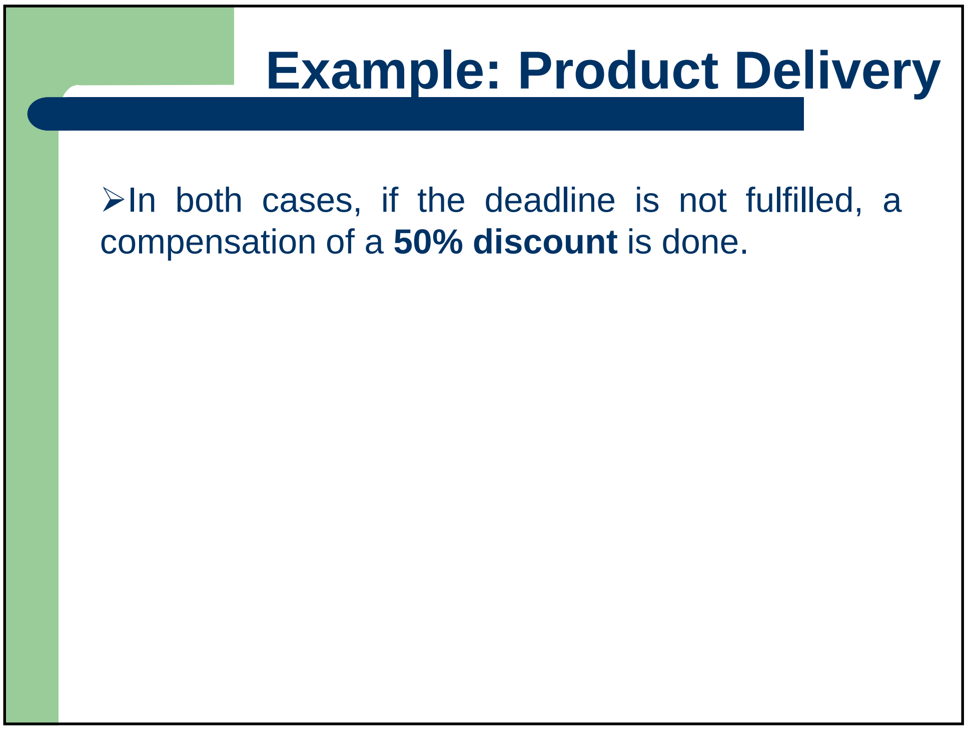
- There are tools like **UPPAAL** supporting this formalism, including a model checker engine.

1. Background
2. Visual Model for e-Contracts
3. Example: Product Delivery
4. Conclusions and Future Work



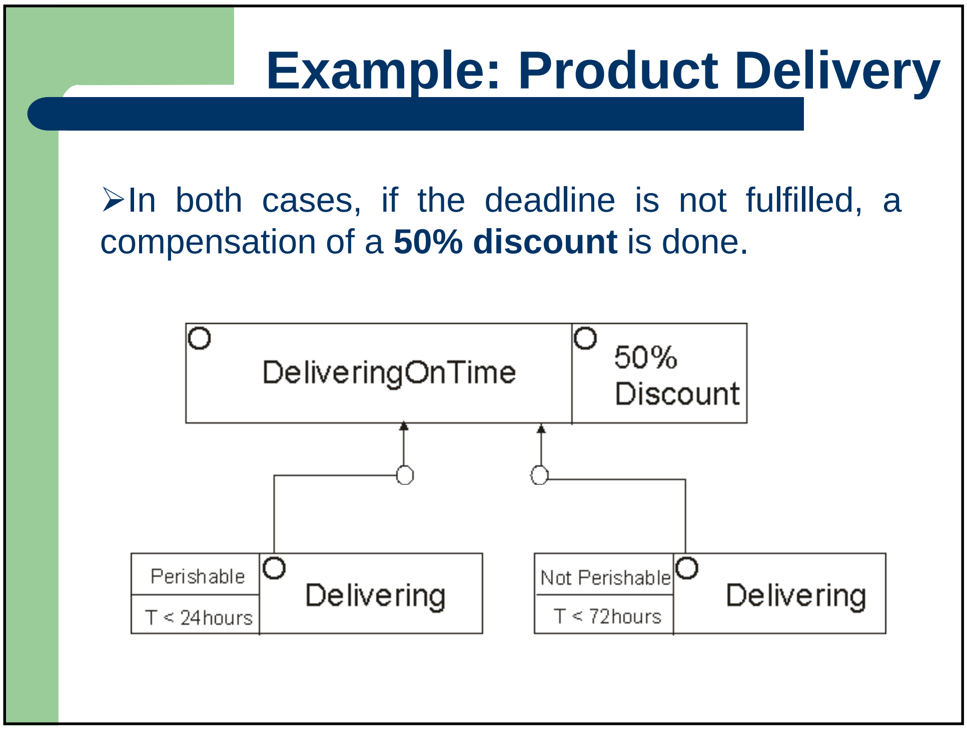
Example: Product Delivery

- Simple process where we can buy a perishable or an imperishable product (customer, deliverer and provider).
- The deliverer **must** deliver the product on time.
- **Perishable product**: The process must complete in less than **24** hours (after customer order).
- **Imperishable product**: The process must complete in less that **72** hours (after customer order).



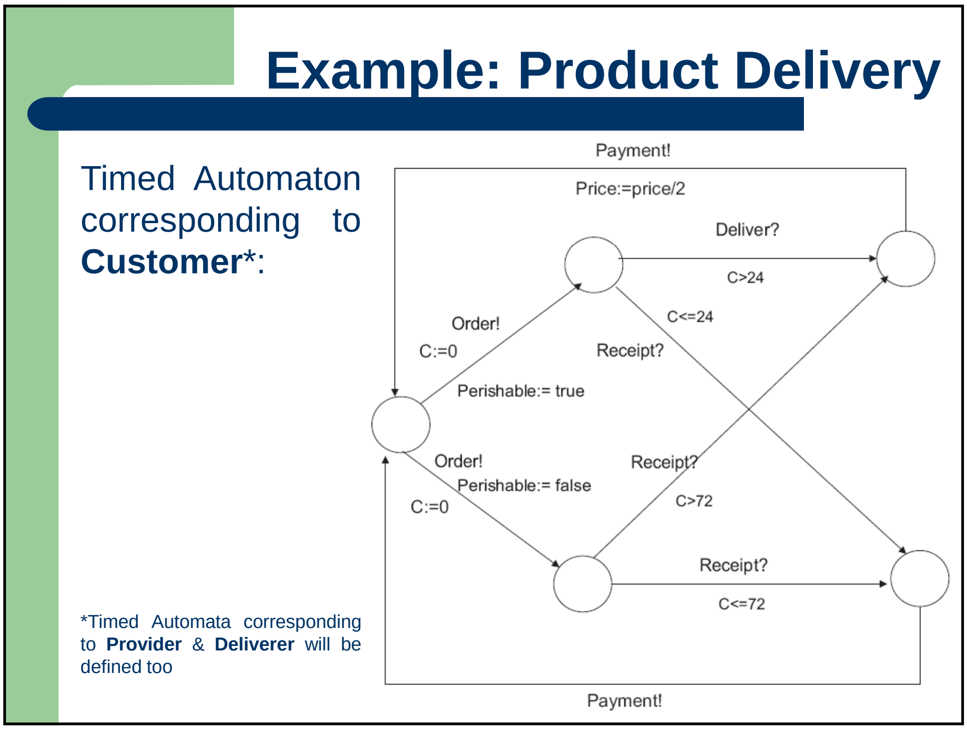
Example: Product Delivery

- In both cases, if the deadline is not fulfilled, a compensation of a **50% discount** is done.



Example: Product Delivery

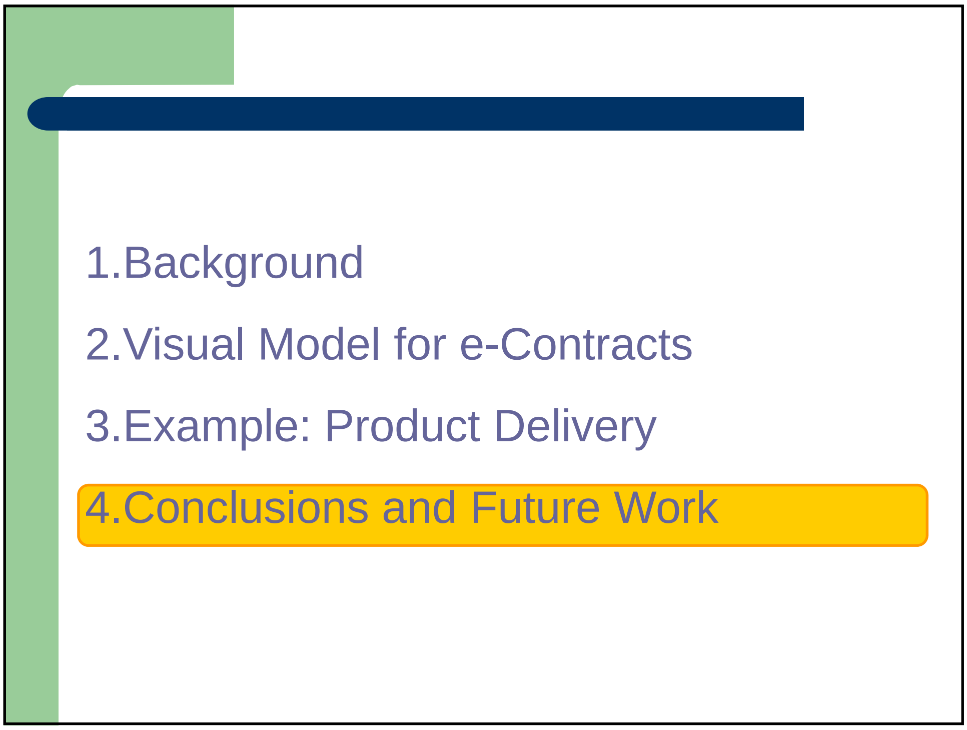
➢In both cases, if the deadline is not fulfilled, a compensation of a **50% discount** is done.



Example: Product Delivery

Timed Automaton corresponding to **Customer***:

*Timed Automata corresponding to **Provider** & **Deliverer** will be defined too

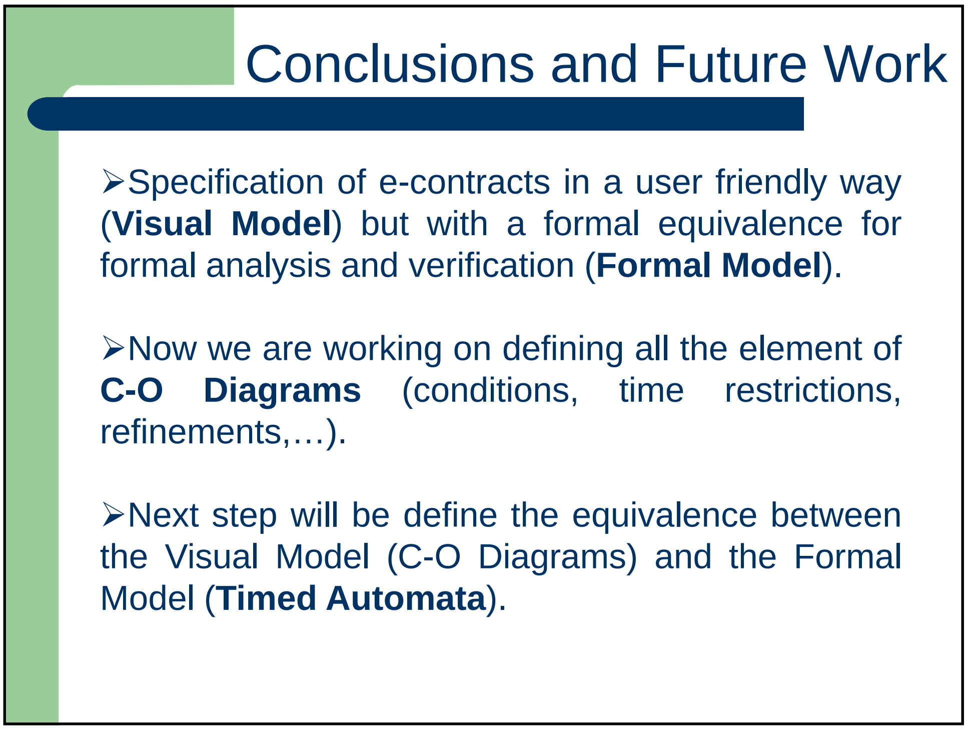


1.Background

2.Visual Model for e-Contracts

3.Example: Product Delivery

4.Conclusions and Future Work



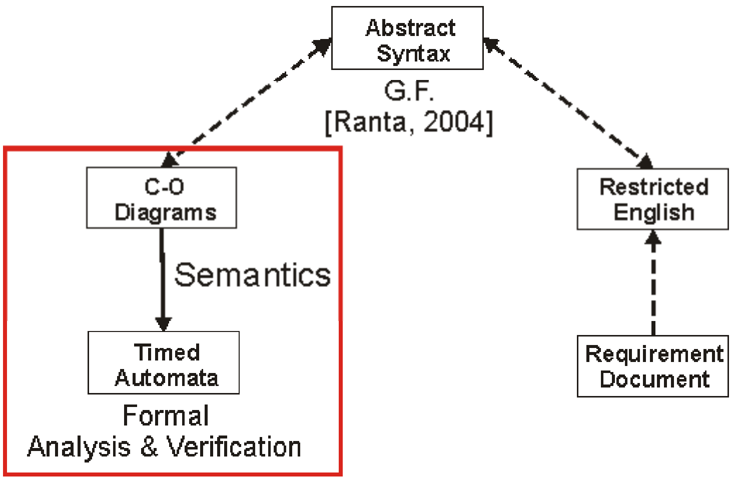
Conclusions and Future Work

➢Specification of e-contracts in a user friendly way (**Visual Model**) but with a formal equivalence for formal analysis and verification (**Formal Model**).

➢Now we are working on defining all the element of **C-O Diagrams** (conditions, time restrictions, refinements,…).

➢Next step will be define the equivalence between the Visual Model (C-O Diagrams) and the Formal Model (**Timed Automata**).

Conclusions and Future Work



Visual Specification of Timed Contracts

Enrique Martínez

emartinez@dsi.uclm.es

Universidad de Castilla-La Mancha. SPAIN