



Design Fest 2004

Data Collection
TerraSense Arbor XT

*Design team: Cristiano, Giuliano, Pedro,
Nélson, Lene, Henning, Fábio*



Problem Description

ArborXT

- A system for gathering, storing and analyzing weather information.
- Provides forest fire prediction, historical trend analysis (possibly forecast), etc.
- Consists of hardware and software located at central site and field locations.



Problem Description (cont.)

- Two main interacting parts
 - Central Computer
 - Maintains configuration data for all sensors.
 - Gathers and stores sensor data.
 - Performs various analysis and presents results to the user.
 - Sensors
 - Report measurements of physical data to the Central Computer.
 - Via various telecommunication hardware.
 - At various intervals of time and tolerances.
 - In a wide variety of units.
 - May employ two distinct data flow models: pull and push.



Challenges

- Differences in telecommunication links and their reliability
- Various types of sensors with different configurations
- Extracting the right information given the potentially complex demands of the user
- Adding and configuring sensors



Assumptions

All sensor readings are stored in a database.

Sensor readings are stored in the database in a format that mirrors the messages sent by the sensors: sensor id, value of sensor reading and timestamp

A sensor stores a list of locations and timestamps marking the first time they were put there

A group of sensors is the result of a query to the set of sensors. It is a responsibility of the *plugins* to store this grouping information if they want it stored



Assumptions *(cont.)*

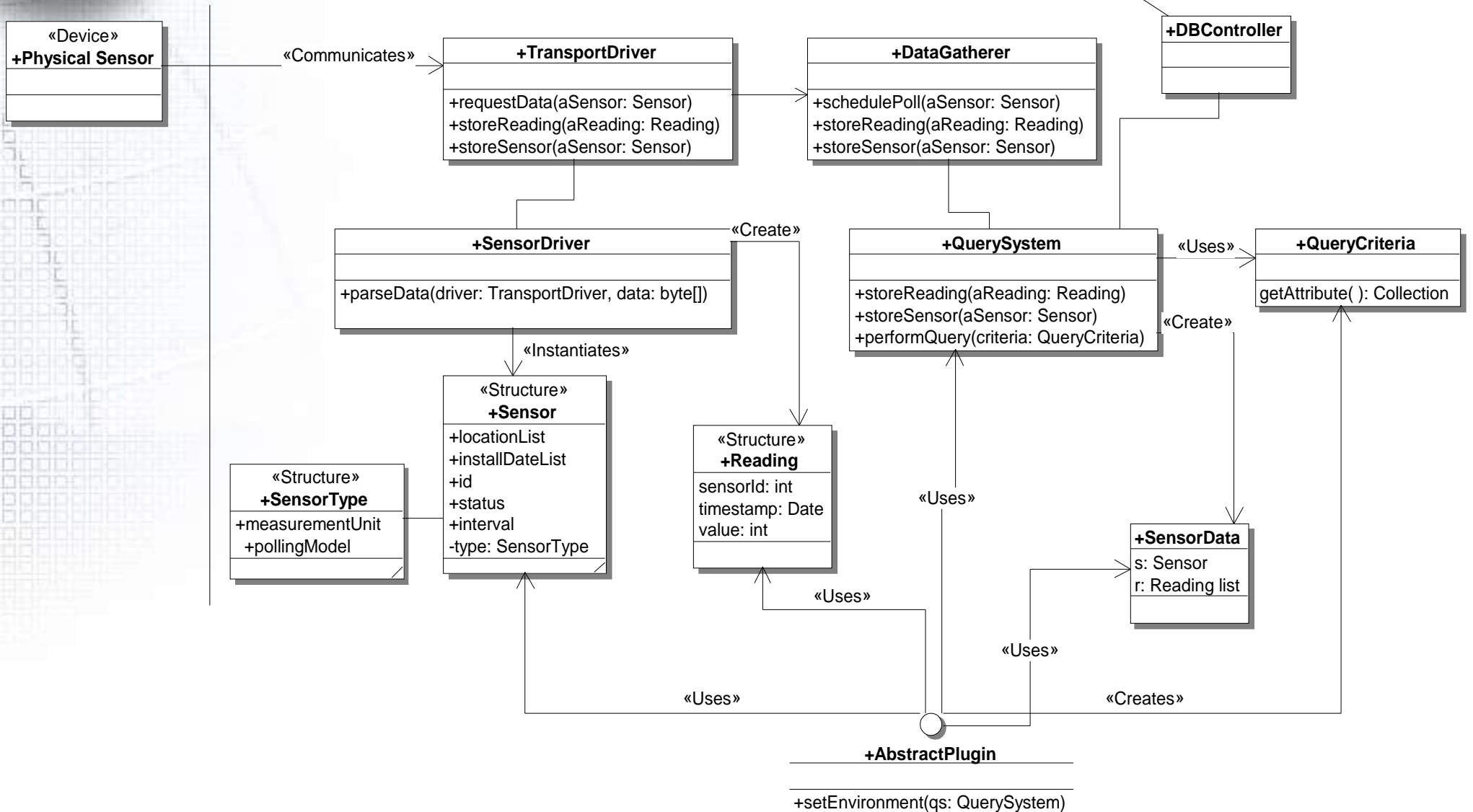
Plugins embed functionality that allows them to build query criteria and pass them to a query system inside the central computer

If a plugin needs to analyze data from a given sensor frequently, it has the responsibility of making periodic queries to the query system to check for changes in the data



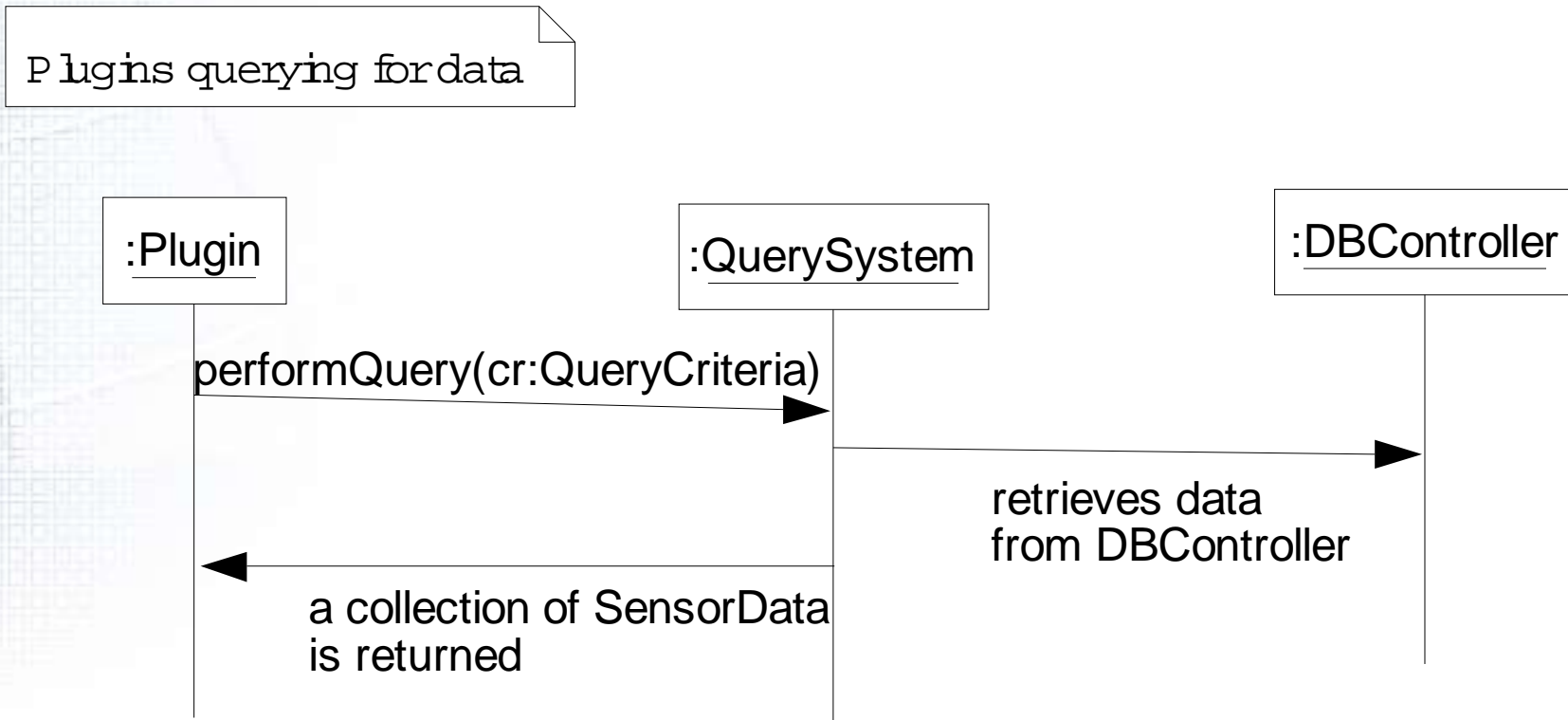
Class model

Standard (relational?) database wrapper.
Could range from a simple façade to a complex implementation such as JDBC.





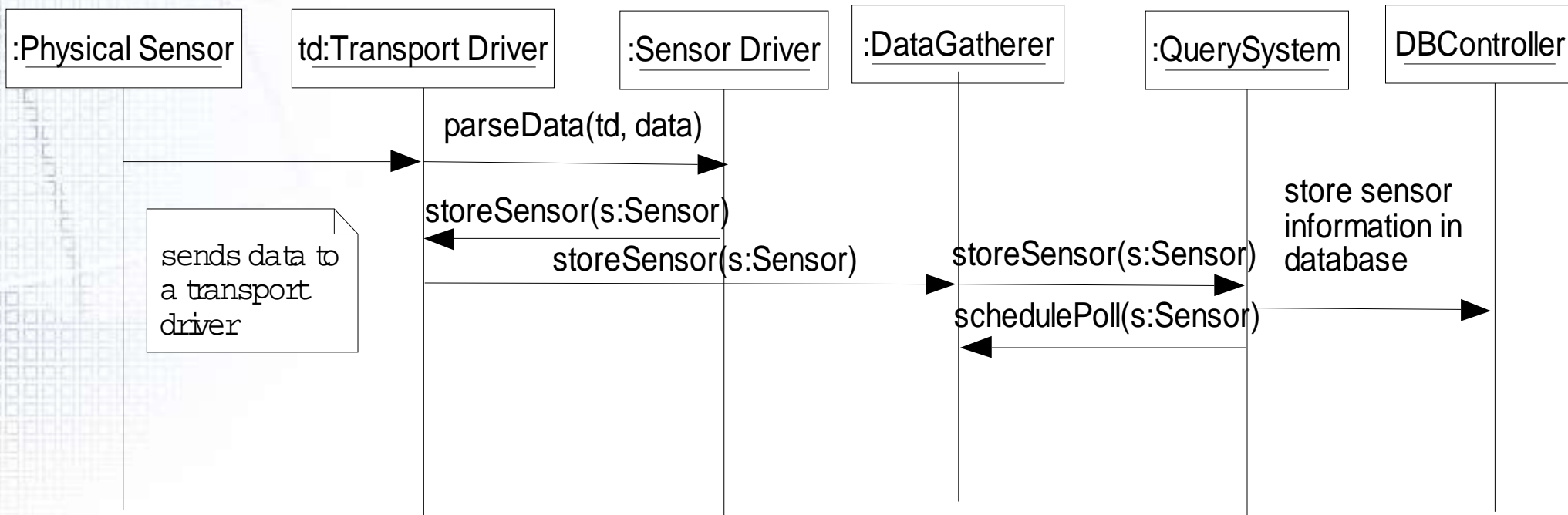
Interactions: *plugins querying for data*





Interactions: *sensor configured automatically*

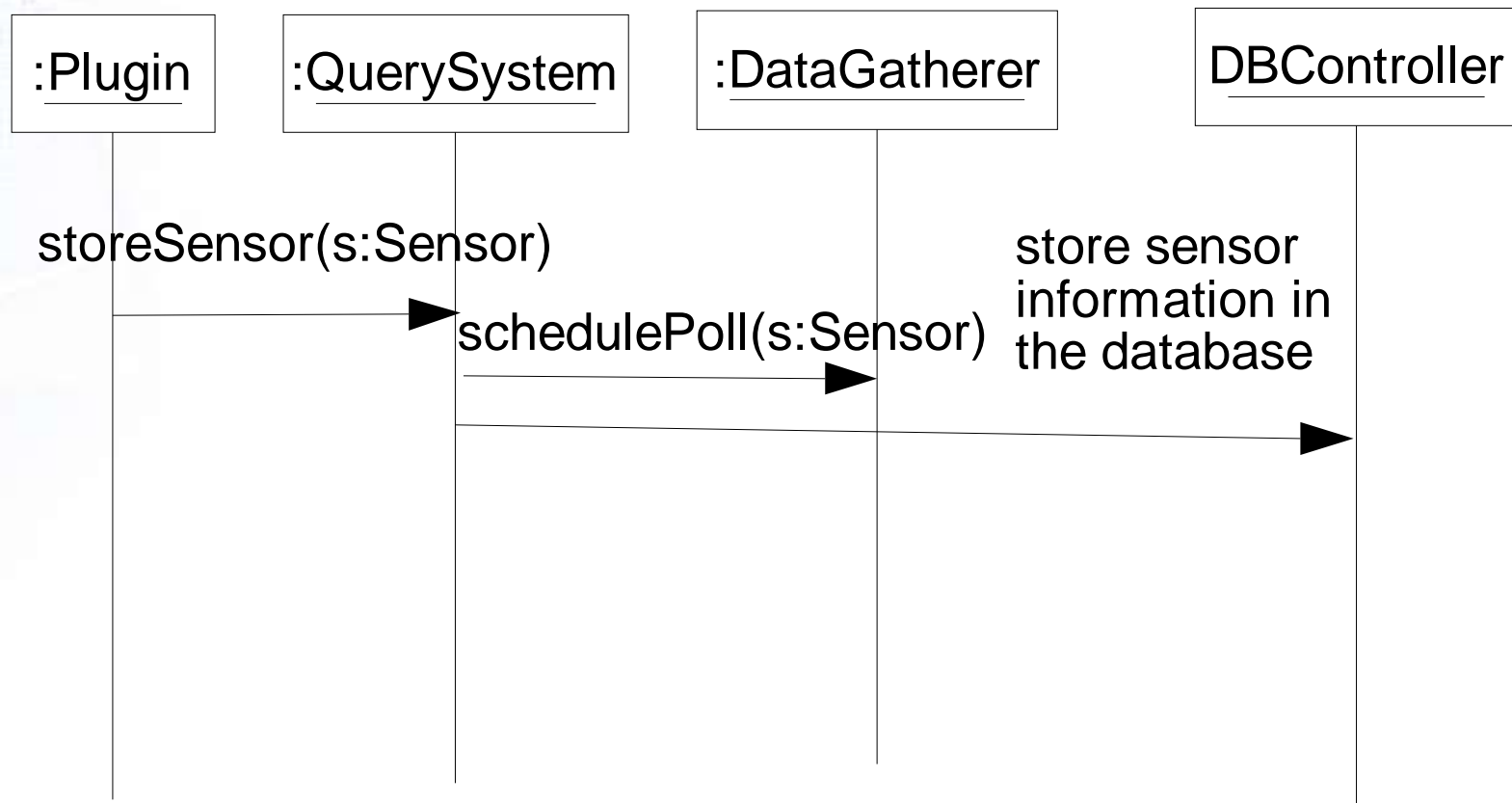
A new sensor is configured automatically





Interactions: *sensor configured by hand*

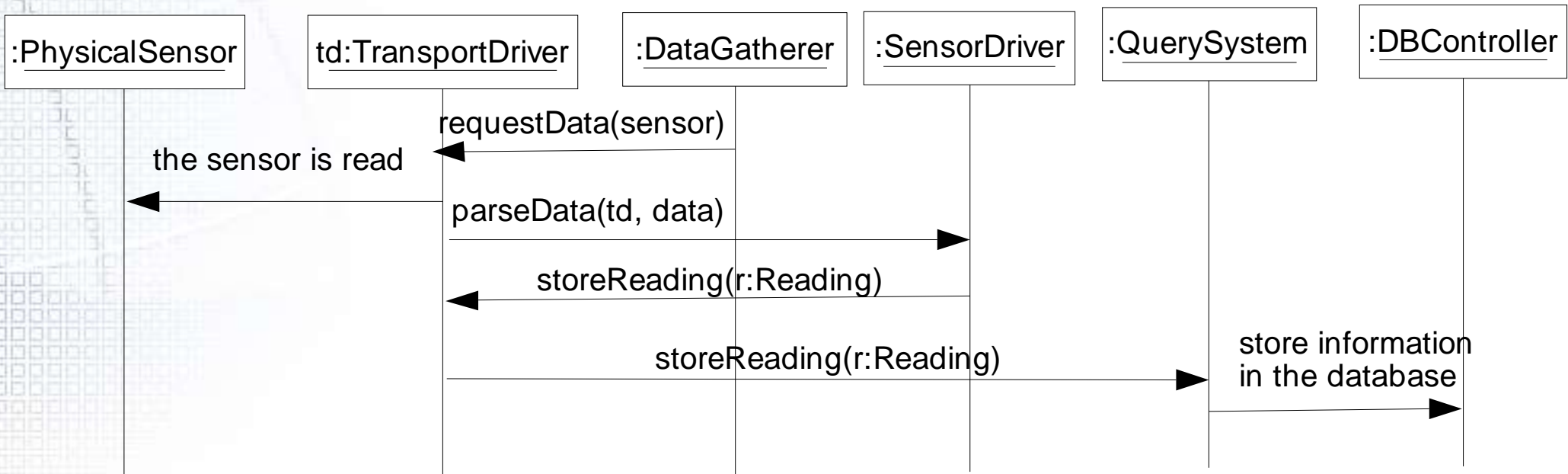
A new sensor is configured by hand





Interactions: *polling of sensors*

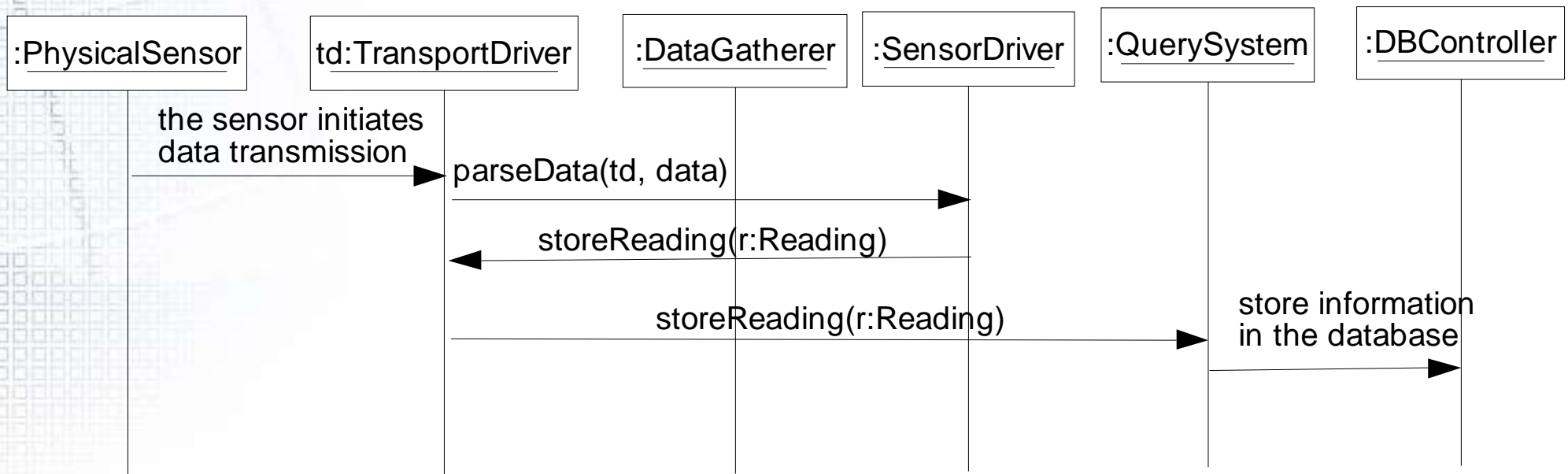
Sensor reading (pull model). Data Gatherer starts sensor readings for given sensors periodically





Interactions: *sensor sends its readings*

Sensor reading (push model).
Sensors send data to the system





Lessons learned

- Process planning
- Read and discuss the case thoroughly
- Keep it simple
- Patterns and communication
- Patterns and design
- Naming of classes