Realtime and Robotics:

- mutually dependent?
- successfully ignored so far?
- mandatory to achieve next level in robotics?
State of the Art in Robotics

- it is commonly accepted that a controller requires (hard) real-time
- thus, most manipulators / compliant motion systems are based on (hard) real-time
- interestingly enough, most mobile robotic systems are based on best-effort approaches

=> why do we think we can bridge the gap between lab prototypes and real-world applications without considering (hard) real-time in robotics?

=> how can we prove robustness / safety and other non-functional and obvious requirements of robotics applications without addressing (hard) real-time in all robotic systems?

=> what are the patterns how we so far (more or less successfully) circumvented (hard) real-time in many robotic systems?

=> how do we achieve real-time robotics behavior?
   Are those patterns generic and reusable?

=> is taking into account time as design parameter just a bit different or turns this all architectures and frameworks upside-down?

=> ...
What the discussion will be about?

- the gap in knowledge:
  - how can this community contribute to solve the problem?
  - actually, we feel you have all the experience to solve the problems since you all have working systems
  - how can we share / make reusable that knowledge?

- the objective:
  - making explicit that knowledge somehow

- we would like to get some feedback:
  - which of the presented topics should be addressed?
  - are those topics interestingly enough to you to contribute?
  - are those topics relevant to the robotics community in general?

=> thus, presentation of different views and topics ...
Possible issues to be discussed at SDIR III

- hints on which kind of real-time support is needed for which kind of robotics system
  - what do we really need?
- hints on how we can extract and represent so far established approaches on real-time in robotics
  - how can we preserve and reuse existing experience in a way independent of implementational technologies?
  - what is about model based approaches?
- hints on why real-time often is ignored in robotics
  - is (hard) real-time the key towards marketable service robots due to resource awareness, safety guarantees, ...?
- ...

- hints on which topics related to real-time robotics behavior are most urgent
  - how could addressing real-time in robotics behaviors look like?
  - what needs most urgent to be done until next SDIR workshop?
How to identify real-time requirements in robotics?

- **Bottom-Up View:**
  - from control loops to high-level task planning
  - can we get rid of (hard) real-time requirements at a certain layer?

- **Architectural View:**
  - can (hard) real-time be addressed locally or is it a design parameter throughout the system?

- **Framework and Development Process View:**
  - what kind of (hard) real-time support is needed in a robotics framework?
  - how should such software be developed in the future?

- **Application View:**
  - do all robotics applications require (hard) real-time?

- **Other Communities:**
  - why do we not just reuse results of the DRE community?
Block 1: Bottom-Up View

What is the Bottom-Up View?
- from control-loops to high-level task planning
- from sensor/actor loops to high-level control loops

Some Examples
- odometry / PID controllers are hard real-time
- in many cases even the next level like obstacle avoidance is already best effort
- vision feedback for approaching / object following closes a control loop across components
- coordinated axis control in manipulators is hard real-time
- mission planning is not hard real-time
- ... 

Major Questions
- what are the reasons why we can ignore (hard) real-time at a certain layer?
- or is it just that the robotics community does not know enough about (hard) real-time?
- where (and why there) did you put the borderline?
Block 1: Bottom-Up View

Q1.1: What kind of real-time do we need for various components? (hard/soft/no/...)
- pose labels for laser range scans?
- laser based collision avoidance?
- vision based approaching maneuver?
- sensor data fusion?
- task net execution and behavior coordination?

Q1.2: Are there generic patterns how the robotics community solved the previously described uses cases so far?
- what type of real-time have you used in which robotics domain?
- where was it not necessary to address real-time?
- why was it not necessary?

Goal
- understand how we handle time in robotics in different domains so far

Possible Outcome
- identify generic solutions / patterns related to handling time
- understand where time is not an issue and why
- is it worth to write down YOUR specific design pattern with its pro/cons based on your implemented robotic system?
Block 2: Architectural View

What is the Architectural View?
- from reactive systems to multi-layer architectures
- from priority driven to time triggered
- loosely coupled / data centric / state based / ...

Some Examples
- pipes and filters (data flow)
- publish / subscribe (data centered)
- ...  
- service-oriented?
- reactive?
- multi-layer architectures?

Major Questions
- is (hard) real-time addressed locally or is it a design parameter throughout your whole system?
- does time turn every known robotics architecture upside down?
- what kind of architecture did you use in which robotics domain for which type of realtime?
- would you do it the same way again?
Block 2: Architectural View

Q2.1: **What kind of robotics architecture is suited to handle which kind of real-time requirements?**
- do we have examples of robotics architectures tailored to real-time?
- can we add real-time to known robotics architectures?

Q2.2: **Is real-time local to components or spans it across components?**
- what about QoS at component interfaces?
- are data centric architectures with loosely coupled components the solution?

**Goal**
- extract how time is handled in robotic system architectures so far
- identify how much / less we already know about (hard) real-time robotics architectures

**Possible Outcome**
- identify how scalable real-time approaches for complex robotics architectures are
- identify whether (hard) real-time and components (loose coupling as mean to handle complexity) fit together in robotics
- **does it make sense to ask YOU to write down YOUR use cases / requirements / experience related to time in a certain robotics architecture?**
- **can we already reflect known robotic architectures under the perspective of time and write down their pros / cons?**
Block 3: Framework / Development Process

Object-Oriented Information View
- Local object-model extending the distributed DCPS data-model
- Manages relationships and supports native language constructs

Distributed QoS-driven Information Management
- Fault tolerant and global persistence of selected data
- Guaranteed data availability supports application fault-tolerance
- Content-aware filtering and dynamic queries:
  - reducing application-complexity
  - improving system-performance

Real-time pub/sub messaging:
- Asynchronous ‘one-to-many’ real-time data communication
- Dynamic data-flow based on ‘current-interest’ (pub/sub)
- Platform independent data-model (IDL)
- Strong-typed interfaces for multiple languages
- Information Ownership management for replicated publishers

Figure 5: DDS Fundamental Model
DDS connects Publishers of Topics to Subscribers to Topics. Publishers offer, and Subscribers request Quality of Service (QoS) parameters. If the offer can satisfy the request, communication is established. Arrival of data or change in status can be monitored through listeners or with blocking waits.
What is the Framework / Development Process View?

- other domains like DRE (distributed real-time embedded systems) already address many of the questions related to hard real-time as needed in robotics
- other domains like DRE already built their tool chains but we in the robotics domain still lack systematic engineering processes
- furthermore, many software technologies like Real Time Java provide suitable means to implement robotics systems

Major Questions

- what kind of (hard) real-time software support is needed in a robotics framework?
- what kind of contribution is needed by roboticists?
- who should build such frameworks?
- what kind of tool-support is needed?
- where should we just take advantage from experts in relevant fields?
- are robotics frameworks (like OMG Robotics Task Force) behind current technology in other domains?
Block 3: Framework / Development Process

Q3.1: Who knows about the projects in the DRE community?
   - Automotive, Avionics: isn't robotics about the same?

Q3.2: Can these tools / frameworks be used in robotics?
   - Programming Languages: Real Time Java?
   - Real-Time Middleware (OMG DDS, OMG Robotics Task Force, ...)

Q3.3: Do we need a separate UML robotics profile?
   - are DRE UML profiles sufficient?

Goal
   - extract what is specific in robotics and what is different compared to what is done in other domains

Possible Outcome
   - what kind of execution container / technology / ideas of other domains is needed in robotics?
     - periodic tasks / time driven / QoS ???
   - can we identify what is different and thus needs to be done by robotics people and what is better done by others?
   - do we need a roadmap for “robotics software frameworks”?
   - do we need “tool support” or a “UML robotics profile”?
   - should we start modeling our best practices?
Introductory Statement

- in many domains like avionics and automotive, it is mandatory to validate and verify overall system behavior based on models and assisted by tool-chains.
- Interestingly, service robotics wants to establish new markets, but we still build software systems without knowing anything about required resources, correctness, etc.

Q4.1: What is about guarantees in robotics?

- Resource awareness
- Safety issues and guaranteed timely reaction
- Model-based approaches to verify timeliness, etc.

Q4.2: How to achieve these?

DREAM: Distributed Real-Time Embedded Analysis Method

http://www.ics.uci.edu/~gabe
Q5.1: Is there something different in robotics?
Q5.2: If there is something different:
- what is it?
- is it necessary to describe the differences compared to DRE?
- what are the appropriate means to describe this?
Q5.3: Does it make sense to focus on the problems of (hard) real-time behavior in a specific scenario?
- pro: different groups share a common idea / model
- cons: very soon, contributions are not generic anymore
Q5.4: If yes, how could such a scenario look like?
- Is there a scenario which highlights the open questions related to real-time behaviors in robotics?
- Mobile Manipulation? - Aerial Robotics? - Dynamic Environments?
- Obstacle Avoidance? - Humanoids? - Urban Challenge?
Q5.5: Is that scenario suited for model-based development as well?

What do you think is a scenario that covers all the relevant aspects?
Is that scenario of interest to all of you?
Extracting a roadmap / next activities from the panel discussion

- **Bottom-Up View:**
  - is it worth to write down YOUR specific design pattern with its pro/cons based on your implemented robotic system?

- **Architectural View:**
  - does it make sense to ask YOU to write down YOUR use cases / requirements / experience related to time in a certain robotics architecture?
  - can we already reflect known robotic architectures under the perspective of time and write down their pros / cons?

- **Framework / Development Process:**
  - what kind of execution container / technology / ideas of other domains is needed in robotics?
  - can we identify what is different and thus needs to be done by robotics people and what is better done by others?
  - is there a roadmap for community service “robotics software framework” or “tool support” or “UML robotics profile”?
  - should we start modeling our best practices?
Extracting a roadmap / next activities from the panel discussion

- **Guarantees:**
  - how to address safety issues?

- **The Robotics Domain / A Scenario:**
  - Does it make sense to focus on the problems of (hard) real-time behavior in a specific scenario?
  - if yes, how could such a scenario look like?
  - is that scenario suited for model-based development as well?

- how could a roadmap addressing real-time robotics behavior look like?
- what needs most urgent to be done until next SDIR workshop?
- what are good starting points to work on?

- on what issues should a PhD-Student work on next?