

# Trabalho Aula 13

**Hariel G.<sup>1</sup>, Leonardo Gubert<sup>1</sup>, Matheus S. Redecker<sup>1</sup>**

<sup>1</sup>Pontifícia Universidade Católica Rio Grande do Sul (PUCRS),  
Avenida Ipiranga, 6681. Prédio 32, CEP 90619-900. Porto Alegre, RS-Brasil

{hariel.dias, leonardo.gubert, matheus.redecker}@acad.pucrs.br

## 1. Crítica em relação as questões

A maioria dos sistemas de hoje em dia não levam em consideração a noção de tempo. Isso faz com que aplicações de tempo real se tornem de difícil modelagem e análise. O *Cyber-Physical System* (CPS) é utilizado para integração de estados físicos com o virtual. A noção de *system state* é ter descrições que representam os estados do sistema ao longo de sua execução. O *system state* não se torna viável em CPS pelo fato de que ele não consegue tratar facilmente ações não determinísticas, e em CPS as variáveis de tempo real são necessárias e isso torna o sistema não determinístico. A noção de tempo torna a modelagem e as execuções desses problemas em uma *system state* praticamente inviável, pois teríamos que ter para cada estado de tempo todas as ações possíveis. Nós concordamos com o autor, pois como podemos ver em aula, o modelo de *system state* deve ter estados bem definidos, enquanto o CPS pode não ter estados assim, e até mesmo ter estados que não foram modelados, mas deve existir algum tratamento para que o sistema não falhe por completo.

O problema da utilização de *threads* em sistemas embarcados se dá pelo fato de ser humano ter certa dificuldade em pensar de forma concorrente. Por essa razão a modelagem desse tipo de problema se torna mais difícil e trabalhosa, pois precisamos perder mais tempo pensando nos possíveis casos de falha, algo complicado de ser prevista de forma antecipada devido a estocasticidade de alguns eventos. Visto que os projetos de sistemas embarcados custam muito caro e a sua falha pode ter consequências catastróficas, a sua modelagem é muito importante em sistemas embarcados. Nós concordamos com o autor, pois como podemos ver em aula, sistemas concorrentes quando escritos em uma lógica sequencial abrem margem para diversos erros de modelagem, como casos não previstos.

Existem trabalhos de pesquisa que mostram algumas direções a serem seguidas em CPS. Como por exemplo, colocar a noção de tempo nas linguagens de programação, hierarquia e gerenciamento de memória com previsibilidade, entre outros. Há ainda, algumas modificações que devem ser realizadas nas construções atuais, como repensar a divisão de hardware e software e a divisão dos sistemas operacionais com as linguagens de programação. Nós concordamos com o autor, pois se pensarmos que as aplicações quando feitas para um propósito específico tem uma performance melhor. Muito dos conceitos utilizados como base, foram propostos nos anos 90, e os problemas atuais tem diferenças em grande escala, o que torna necessário a remodelagem dos conceitos.