ARTIST2 - FRESCOR Workshop on Requirements
Massy, June 16, 2006
Minutes
Gerhard Fohler, TUKL

Attendants

- Zdeněk Hanzálek, Pavel Píša, Czech Technical University, Prague
- Julio Medina Pasaje, CEA, France
- Javier Gutiérrez García, Michael González Harbour , University of Cantabria, Spain
- Alfons Crespo, Joan Vila, Universidad Politécnica de Valencia, Spain
- Alejandro Alonso, Universidad Politécnica de Madrid, Spain
- Francisco Gómez Molinero, Francisco Javier Cabello, Visual Tools, Spain
- Virginie Watine, Vincent Seignole, Thales, France
- Gerhard Fohler, TU Kaiserslautern, Germany
- Luis Miguel Pinho, Polytechnic Institute of Porto, Portugal
- Paulo Pedreiras, Ricardo Marau, University of Alveiro, Portugal
- Paolo Gai, Evidence, Italy
- Tomaso Cucinotta, Scuola Superiore S. Anna, Pisa, Italy
- Alan Burns, University of York, UK
- Guillem Bernat, Rapita Systems Ltd., UK
- Thorbjörn Neander, ENEA AB, Sweden (via presentation on June 14)

Chair:
Michael González Harbour , University of Cantabria, Spain

Agenda:

Application environments
- U. Kaiserslautern: Multimedia applications
- Visual Tools: Media processing applications
- U. York: Artificial intelligence
- Thales: Telecommunication applications
- EVIDENCE: Automobile applications
- U. Cantabria: Industrial automation applications

QoS management
- SSSA Pisa: Energy-aware quality of service
- Tech. U. of Madrid: Quality of Service
- Polytechnic Inst. of Porto: Dynamic Quality of Service

Support for component-based design methods
- Thales: Component-based framework
- CEA: Real-time components

Specific schedulable resources
- Rapita Systems: Worst-case execution time
- U. Aveiro: Real-time networks & distribution
- Czech Technical University in Prague: FPGAs, reconfigurable architectures
- EVIDENCE: Multiprocessor platforms
- U. of Valencia: Memory management

Input to minutes
Michael González Harbour, Javier Gutiérrez García, Gerhard Fohler

1 Enea AB

Thorbjörn Neander
June 14

- applications sit ontop of OSE which sits ontop of HW
- would prefer to have
  - number of appliactions
  - same platform: OSE, Linux – one “virtual CPU” each
  - HW
  - each applications “sees” its own CPU, memory, etc
- would like to have more general, single virtual processor
- memory management is not prime interest, as long as it is save
  - eg memory pool, applications request individual amounts

2 Introduction

Michael González Harbour

- opens and gives objectives of meeting
- agenda of presentations

3 Application environments

3.1 University of Kaiserslautern: Multimedia applications

Gerhard Fohler

- End-to-end streaming involves variations in sources, e.g., streams, and resources, e.g., wireless networks,CPUs
- Application notions of Quality. Provide some means to the application so that it can adapt to the available resources. For each application domain there are different requirements.
- Facilitate applications means to adapt to the available resourcesDifferences between hard & soft real-time, definition of QoS
- Several dimensions : amount by which I can miss the deadline,
  effects of missing a deadline, N in M

3.2 Visual Tools: Media processing applications

Francisco Gómez Molinero

3.2.1 Video surveilance applications

- not only video streaming, but also heavy-crunching video processing
- PC in control center, IP network
- video streaming:
  - LAN or ADSL (big variation)
  - QoS in Internet is an issue, should be able to specify smoothness control
  - video transmitted to
    - PC

- mobile phone
    - pda
    - virtual video matrix
  - several applications on same Linux machine
  - sharing systems resources
  - framework to guarantee QoS
- control:
  - zoom in cameras etc, telecontrol of devices
- video recording and retrieval:
  - video indexing, annotation, labeling images for retrieval.
  - synchronize with data from other devices, timestamping, and results of image analysis
- video display locally:
  - processor bus banwidth
- real-time image analysis:
  - for video annotation, end user data extraction (counting people), safety critical operation
  - examples: object tracking, human behavior, statistical analys
  - people counting, human behaviour: articulated models, face detection & recognition
  - detecting incidents (before it occurs) timing problems
- resource related requirements
  - CPU
    - efficient use, optimize cost
    - ensure that critical functions are executed first, QoS issue
  - processor bus
    - guarantee image capture rate, interference between high bandwidth devices
  - memory
    - optimization of required amount
    - worst-case usage indentification
    - no schedulable requirements
  - network
    - latency more important than sustained bandwidth
    - jitter eeliminated through buffering
    - telecontrol requires controlled latencies

3.2.2 others:

  - synchronzation of video and data
    - traffic
    - retail: sweethearting problem
    - electrical substations

3.2.3 wish list of VT for FRESCOR

- guarantee CPU to maintain framerate
- process video queries
- guarantee network bandwidth
- no procedure to test worst case scenarios
- no way to check global performance of application
- response time with high average load
- latency control
- influence parameters for encoding

- perhaps develop board inside FRESCOR

3.3 University of York: Artificial intelligence

Alan Burns

- Multi-agent systems
  - RoboCup
  - Real-Time pattern recognition
  - RT path finding
- Deliberation and planning
  - anytime algorithms

- Regular reassignment of resources
  - application does allocate CPU time to agent tasks (not possible to do inside OS)
  - use spare capacity constructively

- Contract model
  - need to recalculate budgets dynamically and efficiently
  - need to negotiate for a collection of agents (threads)
  - simultaneously
    - hierarchical contract mechanism

3.4 Thales: Telecommunication applications

Vincent Seignole

- Hard RT in synchronization between peers
- types of systems
  - user terminals
    - different kinds of I/Os (voice, video, data, network, internet)
    - multimedia
    - multi-mode, moving standards
  - base stations
    - multi-user, connected to core network
    - complex signal processing algorithms, smart antennas, macro
  - diversity
    - cluster of MPs, each having RF7DSP/GPP/FPGA
  - core networks

- new technologies needed: component orientation

- requirements
  - run-time admission
  - intelligent allocation/deallocation agent
  - heterogenueous multiprocessor, follow model of distributed system
  - low footprint, GPP 200K+500K for RTOS DSP: 10-20K RTOS less
  - GPP, DSP, FPGA
  - dynamic frequencly scale for power
  - integrated with components
  - platform independence

3.5 EVIDENCE: Automobile applications

Paolo Gai

- Real-Time control
  - SW 250-state machines in Simulink footprint, 200-500K ROM, 16KB RAM
  - system on chip

- integrated on a network 60-100 CPU
  - multicore system on chip, integration in the same unit
  - network CAN, FlexRay

- OSEK/VDX OS
  - very small
  - fixed priorities, IPC, stack sharing
  - 2Kb kernel

- Autosar
  - integration of different applications in the same system
  - software components on top, integrated through AutoSar
  - www.autosar.de

- SW development process
  - Matlab/Simulink
  - generates a function for each block.
  - a thread is a sequence calling functions, with different frequencies

- shared resources
  - 20-100 tasks, requiring microsecs to execute
  - 1,2,4 ms for high frequency, 500ms for low
  - hundreds or more shared resources,lots of access to each
  - tasks usually non preemptive to save on shared resources.

- end-to-end deadlines on the CAN bus
  - car integrator responsible for analyzing traffic on the network
  - enforced by specification and testing

- some functions that have deadlines with RPM of the engine, others with more static characteristics
- scarce resources, modes of operations, when the car goes fast, some things are no longer done

- extensions to OSEK for memory/timing protection
- execution times
  - in some cases are really important, in other cases not
  - usually statistically profiling.

- Granularities
  - quite different
  - tend to use harmonic frequencies
  - Thales – Telekom: less 1ms in DSP, up to 100 ms
  - Visual tools - Video: 50 ms, related to frames
  - Alan - AI: minimum, 20 ms up to seconds
  - some telecoms: microseconds, done in HW

- Requirements:
  - integrate different components in the same ECU
  - scheduling on distributed level (CAN, Flexray..)
  - source code and binary in the same system
  - hard and soft requirements
  - protection between aplication
  - small constraints
- INTEREST project IST

3.6 University of Cantabria: Industrial automation applications

Michael González Harbour

- distributed applications
  - RTOS
  - scheduling FP immediate ceiling
  - timing requirements
- problem:
  - a component does not know its contract (in relation to the use of shared resources and other

components)
- some higher-order entity must prepare the contract and do the negotiation

4 QoS management

4.1 SSSA Pisa: Energy-aware quality of service

Tomaso Cucinotta

- QoS
  - soft RT
  - resource level
  - appplication level
  - user level
  - design support: composability

- Applications
  - multimedia
  - control
  - overload conditions
  - energy

- Requirements for QoS:
  - similar to FSF contracts
  - + global optimizations for overloads and saving battery, if requested

- Application-level requirements mapping (probes, benchmarks)
  global decissions

- QoS metrics: scheduling, latency jitter, misses, frames per second, AV delay, subtitles...

4.2 Technical University of Madrid: Quality of Service

Alejandro Alonso
- QoS & Component framework
  - power
    - optimize power consumption, meeting user quality
    - power information in the components
    - power affects global behaviour
    - power saving: dynamic frequency, voltage scaling, HW devices
  - additional info for negotiation
    - PM policy
    - apps. power info
    - power status
  - additional output: device power state
- quality configuration, availability of resources
- separation of platform, resource manager

4.3 Polytechnic Inst. of Porto: Dynamic Quality of Service

Luis Miguel Pinho
- QoS specification layering
- quality tradeoffs

5 Support for component-based design methods

5.1 Thales: Component-based framework

Vincent Seignole

- Component based for RT:
    - modelling and analysis
    - execution platforms
    - integration of both above
- low footprint, high performanc

5.2 CEA: Real-time components

Julio Medina Pasaje

- transactional approach for analysis and design
- components as structural elements
- self tunning of resource requirements on a particular platform
- specific schedulable resources

5.3 University Valencia: Memory management

Alfons Crespo

- dynamic memory management
- parameters:
    - maximum memory to allocate
    - maximum holding time
- fragmentation can be decreased
    - explictiy compacting memory
    - implicitely garbage collector

- Requirements:
    - specify memory needed
    - allocate
    - deallocate
    - negotiate new memory requirements
    - allocate memory in a temporal way

5.4 ENEA

Thorbjörn Neander, presented by Michael González Harbour

- Relations deadlines/length of buffers and memory/CPU requirements

5.5 Rapita Systems: Worst-case execution time

Guillem Bernat

- wrong asssumptions
- execution time profiles
- not all applications need safe estimations
- impact of misspredicting the WCET
- analyzability vs predictability (analyzable may be enouogh
- profiles, predicted WCET
- probabilistic analysis assumes WCET independence.

5.6 University of Aveiro: Real-time networks & distribution

Paulo Pedreiras

- QoS adaptability
- graceful degradation
- Solutions
    - Master/Slave paradigm for flexible control

- centralized architecture, but replicated masters for fault tolerance.

5.7 Czech Technical University in Prague: FPGAs, reconfigurable architectures

Zdeněk Hanzálek

- Problem complexity
  - monoprocessors, multiprocessors,
  - dedicated processors, variable number of processors
  - reconfiguration, interconnection, routing
  - 1D or 2D partitioning, pre-configuration,
  - tools and their functionality

5.8 EVIDENCE: Multiprocessor platforms

Paolo Gai

- Multicore systems
  - high variability of available approaches
  - in the future there will be multicore systems
    - granularity is not known yet
    - e.g. SUN processors, ARM symetric processors, pico 300 asymetric dedicated multicores
    - shared memory or message passing to communicate processors
  - What kind of OS? POSIX-like, OSE-like

- from point of view of RT-scheduling, the future will be assymetric as Power issues will be better solved this way.

- EVIDENCE
- asymetric multiprocessor
- partitioned approach
- no cache coherence
-
- It seems that with the large variation in HW architectures it will be difficult to provide platform independence.