

Birla Institute of Technology & Science, Pilani

Instruction Division

Department of Computer Science & Information Systems

First Semester: 2015-2016

Indicative Solution to the Comprehensive Examination (Regular)

Course No.:	CS G 541 / SS G 531	Course Title:	Pervasive Computing
Nature of Exam:	Part-1: Closed Book (10%)	Part-2:	Open Book (20%)
Total Weight:	30%	Max. Marks:	30
Duration:	Part-1: 1 Hr., Part-2: 2 Hrs.	Total Duration:	3 Hrs.
Date of Exam.:	Dec 7, 2015	Timing:	AN: 2-5 PM

Note: Please make suitable assumptions, where necessary and briefly justify them. You cannot, however, change the nature of the question itself by making any assumptions.

Part-1 (Closed Book, 1-Hour)

1. Why is that:
 - a. Not all Operating Systems designed for Sensor-Compute nodes (sensor nodes) have a Kernel.
 - b. Some Embedded Operating Systems, may have kernel-space as well as user-space support for Device Drivers.
 - c. Interaction Protocol Designs involve specifying Individual Messages or Patterns of Multiple Messages.
 - d. Transparency of computing, sensing and network communication elements as well as inter-device interactions is essential for effective Pervasive Computing environments.

1x4=04

Indicative Solution:

- a. *Not all Operating Systems designed for Sensor-Compute nodes (sensor nodes) have a Kernel due to following reasons:*
 - i. *Operating Systems, in general, can be of many types in terms of architecture: monolithic, modular, layered, componentised, hybrid and other than the monolithic variants, other types may feature monolithic kernel, micro-kernel, exo-kernel and other variants of core of the OS architecture. However, not all types of Operating Systems need to have a Core or Kernel per se.*
 - ii. *A very small OS designed for a very resource-constrained computing device is to be conceived with an architecture that keeps the overheads as low as possible while keeping the design simple yet optimally planned keeping the actual intended set of bare minimum functions in view.*
 - iii. *This is why some of the Operating Systems like Tiny OS, specifically designed for Sensor Network Nodes (Sensor-Compute Nodes as used in WSNs) need not and do not have a Kernel; whereas some others do have.*
- b. *Some Embedded Operating Systems, like Microsoft Windows Mobile / CE (other variants) may have kernel-space as well as user-space support for Device Drivers whereas some others either have only the Kernel-space Device Drivers (most common) or only the User-space Device Drivers (relatively rare). This is because*
 - i. *Embedded systems often benefit from either a Limited-Purpose Operating System (LPOS) or an Application-Specific Operating System (ASOS).*
 - ii. *LPOS or ASOS, often benefit from smaller boot images and thus keep the minimum required functionality keeping the purpose in view.*
 - iii. *While, more often than not, only kernel-space device drivers are supported in such cases keeping need-frequency, consequent optimal use and security in view, in some cases, it does help if the applications get the flexibility to*

handle device(s) directly, even if infrequently and particularly in cases wherein it is less likely that other applications would compete in getting access to that given device. Such a provision helps in two ways: a)- it does not force the kernel-size / boot image to grow unduly; and b)- it allows certain degree of flexibility in application / user space.

- iv. Consequently, a few Embedded Operating Systems do support both modes of Device Drivers. (There is, also, a variant that supports what is some times referred to as kernel mode and user mode Device Drivers.)*

c. Interaction Protocol Designs involve specifying Individual Messages or Patterns of Multiple Messages because of the following reasons:

- i. At the fundamental level, interaction may be seen to be comprising of multiplicity or cardinality of communication-specific relations between involved nodes / entities.*
- ii. Multiplicity of Interaction might need to handle resource-contention issues in case multiple nodes, processes or people need to have such interactions, apart from increasing interaction-complexity and adversely affecting the control of interaction as well as determinism of it. This leads to the need of having one or more mechanisms and strategies of effecting control on the interactions. (Multi-Agent Systems, often, use some such mechanisms.)*
- iii. It is, thus, evident that specifying Interaction Protocols between Intelligent Systems may have to handle two separate aspects: (a)- specifying Individual Messages and (b)- specifying patterns of multiple messages.*

Messages themselves may have several types (like Request Message, Response Message, Acknowledgement Message, Error Message etc.). IN practice, Individual Messages are often exchanged as part of 'conversation' that represents an interaction. However, when multiple Messages are involved in a situation representing multiplicity, instead of of being represented as individual messages, they are often represented as Patterns of Messages. For instance, Information-Sharing Message Pattern(s) and Task-Sharing Message Pattern(s) are used two of the many possible patterns in an Inter-Intelligent System Interaction Protocol Design situation.

d. Transparency of computing, sensing and network communication elements as well as inter-device interactions is essential for effective Pervasive Computing environments since:

- i. In absence of such a provision, user would have to know details of all or many of the elements involved and would require explicit knowledge about how to access a particular resource or make use of a specific service; something, that is clearly undesirable in true Ubiquitous Computing environments.*
- ii. In absence of these, invisible computing would no longer be possible in true sense and this would also mean that non-technical / layperson user would need to either use eHCI or iHCI unlike what such a paradigm prefers.*
- iii. Transparency does bring in certain indirect benefits like low-end security becomes simpler to be assured as unless adequately knowledgeable, most common users would not be able to pose select kinds of security threats that become easier to get exposed to when each element is visible to all users. However, there do exist counterviews to this view.*

2. List the constituent elements (hardware, software, service, functional elements, if and as applicable) of the following:
- Smart Dust (original concept)
 - Smart Watch (take any generic or specific example)
 - Plug-and-Play Protocol (any generic or specific PnP may be chosen)
 - Hybrid Intelligent System (in generic sense)

1x4=04

Indicative Solution:

- a. *Smart Dust (original concept) attempted to investigate the feasibility of use of MEMS technology in such a way that in about a cubic millimetre volume an autonomous system could be created that could involve computing, sensing and communication elements in terms of involved hardware as well as corresponding embedded operating system and embedded single-purpose application software as software elements.*

Smart Watch (in a generic case) attempt to provide a set of smart / intelligent services to the wearer which could include features like select physiological parameter monitoring, select types of physical activity tracking, PIM, computing, storage, communication etc. by involving hardware elements like processing, storage, sensing, communicating, I/O device elements with or without smart card elements (in form of SIM / USIM etc.) as well as software elements like embedded Operating System and Multiple Embedded Application Software elements.

- c. *Plug-and-Play Protocols may be of several different types, as usable in typical Ubiquitous Computing scenarios. One such protocol is the Universal Plug-and-Play Protocol (UPnP). UPnP involves a Device Discovery Framework as one of its major elements which has gradually evolved over the years from its use in automatic blinds to connected homes and beyond. UPnP features a Service Layer element that can be customised to enable the emerging technologies like IoT by the way of allowing device, network and service level interoperability. (Use of architecture-neutral data interchange provision too comes in handy.) In its form known as UPnP+, it has ready support elements for bridging, sensing, cloud-based access, data modelling and security amongst other things.*

- d. *Hybrid Intelligent System (in generic sense) may be interpreted from two orthogonal perspectives:*

- HIS may be seen as an IS that is hybrid in the sense unlike pure hardware-only IS or Software-only IS, it comprises of both hardware and software elements. This perspective, however, was the one that was more popular about a decade ago than now.*
- HIS, today, often refers to an Intelligent System that, in fact is combination of multiple IS models. For instance, an HIS might involve multiple IS elements based on Reflex, Proactive, Learning models. (This list is not exhaustive.)*

Autonomy, learning and cooperation are three common features of modern Intelligent Systems and HIS variants do inherit all these features with greater focus on cooperation. Hybrid Intelligent systems are considered suitable for being used in partially observable, non-deterministic, sequential, dynamic ubiquitous computing environments.

3. What are the kinds of mobility-specific considerations in the context of pervasive computing environments designed to support the elderly? Please briefly justify your answer.

1x2=02

Indicative Solution:

Part-2 (Open Book, 2-Hour)

4. Explain in brief, one of the possible ways / techniques to identify, represent, store and manipulate appropriate parameters of the context as applicable to any given situation that warrants use of Pervasive Computing / Ubiquitous Computing environment and associated services. You may choose to take a simple real-life example situation and illustrate the steps involved.

1x4+02=06

Indicative Solution:

5. Consider the recent grave flooding situation that was faced by Chennai due to heavy rains and certain factors that had to do with the way most of our towns and cities are allowed to grow and function.

Keeping in view, the *Smart City* concept that is, often, interpreted / misinterpreted in a variety of ways, identify any specific problem faced during the referred Chennai Flood that you feel could benefit from the *Ubiquitous Computing* technology such that hardships faced by people in the affected areas could be either eliminated or reduced to a great extent, if the city ever again faces similar fury of nature.

Explicitly state the problem / sub-problem(s) chosen by you and suggest both, a broad approach to solve it as well as mention select appropriate computing, sensing and communication elements that would allow relevant context-aware services to help tackle the chosen problem(s) while keeping economics, user-acceptance, user-safety, ease of implementation and a reasonable degree of privacy in view.

5+2=07

Indicative Solution:

6. Consider a situation wherein you have to recommend a simple way to transform existing Central Library (in your respective campus) into an entity that would allow intuitive user interfaces and context-aware pervasive computing services for all stakeholders providing the following features:
- User (Library Administrator/Faculty/Student/Visitor etc.) Authentication and Authorization with support for multiple-profiles in case of select privileged users,
 - Context-based pre-loading of the study or research material in appropriate media format in the user's PC, Laptop, Netbook, Chrome book, Tablet or Smartphone wirelessly (or over wireline network as the case might require);
 - RFID Support for Book / Disk / other Physical Media issue, tracking and management,
 - Ambient reading environment with authorization-based protected, profile-based digital resource access,
 - Assistive-services for select categories of the differently abled users.

What broad solution approach and consequent choices would you recommend and why? Please, also, briefly mention and justify your choices in terms of hardware, software, firmware, protocols and/or sensory and actuation elements.

1x5+2=07

Indicative Solution:

