White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of York.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/181402/

Version: Published Version

Article:
Akesson, Benny, Nasri, Mitra, Nelissen, Geoffrey et al. (2022) A Comprehensive Survey of Industry Practice in Real-time Systems. Real-Time Systems. pp. 358-398. ISSN: 1573-1383

https://doi.org/10.1007/s11241-021-09376-1

Reuse
This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

UNIVERSITY OF LEEDS
University of Sheffield
UNIVERSITY of York

Real-Time Systems
https://doi.org/10.1007/s11241-021-09376-1

A comprehensive survey of industry practice in real-time systems

Benny Akesson[1,2] · Mitra Nasri[3] · Geoffrey Nelissen[4] · Sebastian Altmeyer[5] · Robert I. Davis[6]

Accepted: 28 September 2021
© The Author(s) 2021

Abstract

This paper presents results and observations from a survey of 120 industry practitioners in the field of real-time embedded systems. The survey provides insights into the characteristics of the systems being developed today and identifies important trends for the future. It extends the results from the survey data to the broader population that it is representative of, and discusses significant differences between application domains. The survey aims to inform both academics and practitioners, helping to avoid divergence between industry practice and academic research. The value of this research is highlighted by a study showing that the aggregate findings of the survey are not common knowledge in the real-time systems community.

Keywords Real-time systems · Survey · Industry practice

✉ Benny Akesson
benny.akesson@tno.nl; k.b.akesson@uva.nl

Mitra Nasri
m.nasri@tue.nl

Geoffrey Nelissen
g.r.r.j.p.nelissen@tue.nl

Sebastian Altmeyer
altmeyer@informatik.uni-augsburg.de

Robert I. Davis
rob.davis@york.ac.uk

[1] ESI (TNO), Eindhoven, The Netherlands

[2] University of Amsterdam, Amsterdam, The Netherlands

[3] Electronic Systems Group, Eindhoven University of Technology, Eindhoven, The Netherlands

[4] Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands

[5] Department of Computer Science, University of Augsburg, Augsburg, Germany

[6] Department of Computer Science, University of York, York, UK

Published online: 11 November 2021

Extended version

This paper builds upon and extends the preliminary paper "An Empirical Survey-based Study into Industry Practice in Real-time Systems" by Akesson et al. (2020b), adding:

(i) A discussion of potential threats to validity of the survey and its results, as well as the steps taken to mitigate them (see Sects. 3.1 and 3.3).
(ii) A statistical analysis and discussion of the results of the survey, in the context of its five objectives (see Sect. 5).
(iii) A discussion of the results of a quiz aimed at determining if the aggregate findings of the survey are common knowledge in the real-time systems community (see Sect. 6).

Readers wishing to take the quiz themselves are recommended to do so before reading the survey. The quiz can be found online at https://www.surveymonkey.com/r/quiz_real-time. The quiz questions are also given in the appendix, with the answers at the end.

1 Introduction

The real-time embedded systems field covers a broad range of systems from simple control loops on micro-controllers to complex interconnected distributed systems. These systems span many different application domains, including avionics, automotive, consumer electronics, industrial automation, and medical systems, each with its own requirements, standards, and practices. This diversity makes industrial real-time systems and their associated design methods difficult to characterize.

Some fields, such as software engineering (Höfer and Tichy 2007; Wohlin et al. 2012; Kitchenham and Pfleeger 2008) and systems engineering (van der Sanden et al. 2021; Muller 2013), have a history of systematically researching industry practice using surveys, interviews, and literature reviews (Mohagheghi and Dehlen 2008). This provides a view of the perceived relevance, benefits, and drawbacks of different technologies and methods; identifies trends and opportunities for future research; and tracks the adoption of existing research results. By contrast, there is no such tradition of empirical studies into industry practice in the real-time systems field. This omission contributes to a gap, and potentially a divergence, between industry practice and academic research. This paper addresses that gap via an empirical survey-based study into industry practice. The five objectives of the study were to:

O1 Establish whether timing predictability is of concern to the real-time embedded systems industry,
O2 Identify relevant industrial problem contexts, including hardware platforms, middleware, and software,
O3 Determine which methods and tools are used to achieve timing predictability,
O4 Establish which techniques and tools are used to satisfy real-time requirements,

O5 Determine trends for future real-time systems development.

A survey targeting industry practitioners in the area of real-time embedded systems was developed and distributed. The survey comprised 32 questions related to the five objectives. Based on the survey data, we formulated a number of propositions about the characteristics of real-time embedded systems and on current practice in industry.

The four main contributions of this survey are:

1. Insights into the characteristics of real-time systems based on responses from 120 industry practitioners from a variety of organizations, countries, and application domains.
2. Discovery of statistically significant differences between the three largest application domains: avionics, automotive, and consumer electronics.
3. Generalization of the results from the survey data to the broader population that it is representative of, via the use of standard statistical tools.
4. Evidence that the aggregate results of the survey are not common knowledge in the real-time systems community.

The remainder of the paper is organized as follows: Sect. 2 outlines related work. Section 3 describes the methodology used, threats to validity including steps to mitigate them, and the design of the survey. The survey questions and results are elaborated in Sect. 4, along with key observations and a discussion of statistically significant differences between domains. In Sect. 5, we revisit the objectives, providing a number of propositions based on a generalization of the sample results using statistical interference. Section 6 provides empirical evidence that the aggregate findings of the survey are not common knowledge in the real-time systems community. Finally, Sect. 7 concludes with a summary and directions for future work. (Note, the paper makes use of a common terminology and vocabulary, familiar to real-time systems researchers. A glossary of such terms can be found at: https://site.ieee.org/tcrts/education/terminology-and-notation/).

2 Related work

Research methods used for understanding industry practice can be broadly divided into three categories:

1. *Survey-based research* targeting industry practitioners in one or more application domains (van der Sanden et al. 2021; Torchiano et al. 2013; Liebel et al. 2018; Hutchinson et al. 2014; Forward and Lethbridge 2008; Broy et al. 2012; Whittle et al. 2014; Vetro et al. 2015; Hermans et al. 2009). Surveys have the advantage that they can often reach more than 100 practitioners, but the drawback that they only invest 10–15 min answering 20–30 predetermined questions.
2. *Interviews* with industry practitioners, either open or based on a framework of questions, with the answers subsequently analyzed (Whittle et al. 2017; Kuhn

et al. 2012). This approach has the disadvantage that it typically reaches far fewer practitioners, as interviews require more effort to conduct and analyze; however, it benefits from a more dynamic and interactive structure and hence richer responses (Muller 2013).

3. *Literature surveys reviewing case studies* with the goal of categorizing industry experiences (Mohagheghi and Dehlen 2008).

Some works combine both survey-based research and interviews, exploiting their complementary nature to improve overall quality (Hutchinson et al. 2014; Whittle et al. 2014). There are also replication studies investigating how results generalize to other populations (Vetro et al. 2015; Höfer and Tichy 2007).

In contrast to the fields of software and systems engineering, there has been little if any research undertaken into industry practice in the real-time systems field. Instead, the academic community tends to look inwards, surveying and classifying its own work rather than studying industry practice, contexts, and needs. Examples of well-known literature review surveys include those on uniprocessor scheduling (Audsley et al. 1995; Sha et al. 2004), multiprocessor scheduling (Davis and Burns 2011), limited preemptive scheduling (Buttazzo et al. 2013), mixed-criticality scheduling (Burns and Davis 2017), resource allocation and mapping (Singh et al. 2017), timing analysis (Wilhelm et al. 2008), and multi-core timing analysis (Maiza et al. 2019). Recent literature surveys (Maiza et al. 2019; Davis and Cucu-Grosjean 2019a, b) and other initiatives[1] take this one step further and include diagrams illustrating how the number of publications on different research topics has varied over time. This allows hot-topic areas to be identified. While these works may be useful to identify trends in academic real-time systems research, these trends may not be reflected in industry. In conclusion, there is no existing work that systematically surveys industry practice in the real-time systems field. The aim of this paper is to address that omission and to help close the gap between industry practice and academic research.

3 Methodology

The study described in this paper has five objectives **O1** to **O5** (listed in Sect. 1) that focus on industry practice. To meet these objectives, we chose, as the research method, a *survey* asking industry practitioners a set of predetermined questions. As noted in Sect. 2, we found no existing surveys in the relevant area and thus could not reuse any design and assessment of validity and reliability. It was therefore necessary to develop a new survey instrument. In designing the survey, we first considered appropriate validity criteria and their relevant threats (see Sect. 3.1), before making design choices to mitigate those threats (see Sect. 3.2). Note, in the description below, we follow the structure, classifications, and terminology proposed by Kitchenham et al. (Kitchenham and Pfleeger 2008) for survey-based research.

[1] https://github.com/ebni/classify-rt and https://site.ieee.org/tcrts/classification-of-rtss-papers/.

Table 1 Threats to validity

Threats to construct validity	
Threat 1	*Improper measurement instrument and/or process* not measuring the characteristics of interest in the right way
Threat 2	*Improper measured attributes* not selecting the right attributes to represent the characteristics of interest.
Threats to internal validity	
Threat 3	*Self-exclusion bias* participants who are not interested in or not allowed to answer the questionnaire do not fill it in.
Threat 4	*Inclusion of "foreign units"* questions are answered by respondents who do not have the necessary expertise.
Threat 5	*Selecting duplicate units* two different teammates who work on the same system answer a question differently.
Threat 6	*Relying on personal experience* respondents with different levels of experience may answer questions differently.
Threat 7	*Personal bias* respondents who fill in the survey may have a certain bias with respect to the questions.
Threats to external validity	
Threat 8	*Sampling bias* the probability that different members of the intended population from part of the sample is not uniform.
Threat 9	*Geographical sampling bias* geographical location could impact the chance of receiving an invitation.
Threats to conclusion validity	
Threat 10	Low statistical power or improper use of statistical tests and analyses.

3.1 Validity criteria and threats

Four categories of validity can be identified (Wohlin et al. 2012):

(i) *Construct validity* refers to the degree to which a question measures what it claims to measure.
(ii) *Internal validity* reflects whether all causal relations are studied or if unknown factors could affect the results. The main threats to internal validity come from coverage issues (Torchiano et al. 2013), as detailed in Table 1.
(iii) *External validity* is the extent to which the research results can be generalized to the world at large (Kitchenham and Pfleeger 2008).
(iv) *Conclusion validity* is concerned with the ability to draw correct conclusions from the study methods.

Table 1 provides details of threats in each of these categories of validity (Wohlin et al. 2012; Torchiano et al. 2013).

3.2 Survey design, instrumentation, and process

Survey design: The survey was designed as a *cross-sectional study*, i.e. a snapshot taken over a particular period of time, December 2019, to April 2020 in this case. We used a *self-administered questionnaire* on SurveyMonkey[2] that could be answered without the need for intervention, and without providing respondent or company identification. Further, we did not automatically collect any data relating to respondents identities (IP addresses, etc.), thus preserving anonymity. The aim here was to reduce the risk of self-exclusion (Threat 3) by enabling those who work on confidential projects to still answer the survey questions. As an additional guarantee of anonymity, we only release summarized and aggregated results.

In designing the survey, we were cognizant of the trade-off between having a more comprehensive set of questions and increasing the time needed to complete the survey, which could reduce the completion rate. We began by considering more questions, but converged on approximately 30, inline with recommendations for scientific surveys, aimed at avoiding problems of abandonment (Threat 3). The questions were designed specifically to capture information pertinent to the five objectives of the survey.

Survey questions: We focused on *closed questions*, where respondents are asked to select one or more answers from a list of predefined options. Closed questions are typically faster to answer and easier to analyze than open ones, thus they reduce the likelihood that a participant abandons the survey before reaching the end (to avoid Threat 3). The drawback of closed questions is that they limit the range of possible responses.

Phrasings of the questions, the predefined options, and the scales used to code responses can all impact construct validity. To mitigate Threat 1, the questions were carefully formulated to be neutrally worded, as precise as possible, and avoid unnecessary jargon. Key terms were explained where necessary, helping to ensure that the questions and the list of predefined options were as unambiguous and easy to comprehend as possible. We did not, however, manage to completely eliminate ambiguity caused by jargon. For example, the term "distributed system" used in Questions 10 and 11 may have been interpreted in two different ways. (This is discussed further in Sect. 4 in the observations on Question 11).

With closed questions, care was taken to ensure that the predefined options were unbiased, mutually exclusive, and as exhaustive as possible (to avoid Threat 1). We included "Other", where appropriate, to give respondents the opportunity to go beyond the predefined options when necessary. Where appropriate, we also allowed multiple options to be selected to prevent arbitrary choices between equally valid answers. This was particularly important in the context of real-time systems comprising different sub-systems to which different answers could apply. Both of these techniques are listed as best-practices (Kitchenham and Pfleeger 2008). The category "I do not know" was added to a number of questions where we did not expect all respondents to be able to give an answer, despite belonging to the target

[2] https://surveymonkey.com.

population. This is common practice, despite there being some disagreement about it in the social science community (Kitchenham and Pfleeger 2008). The "I do not know" category has the benefit of making the lack of knowledge explicit, and distinct from skipped questions and arbitrary answers.

One of the main challenges to construct validity is striking a balance between usefulness and interpretability of the data gathered. Specifically, we had a choice of asking respondents to consider either one or several real-time systems that were being developed in their organization. We chose the former, since although this approach gathers data about fewer systems, it enables conclusions to be drawn about individual systems, and answers to different questions to be related to one another.

Instructions for participants: The welcome page of the survey was written in a neutral tone, explained the purpose of the survey, defined the target population, and suggested that it would take 10–15 min to answer the 32 questions. It also explained that the survey was anonymous and that the output of the survey would be an academic paper. The incentive for the respondents was the opportunity to shape future research in the area of real-time systems and align it with industry practice and needs.

Survey validation: To mitigate Threat 2, a draft of the survey was validated by a test group comprising 13 domain experts with extensive industry experience. Their independent and concurrent feedback on both questions and possible answers was used to improve the survey and to ensure that it was fit for purpose.

Sampling method: Since there is no list that identifies all industry practitioners who work on real-time embedded systems, it was not possible to perform a *random sampling* to invite the participants. Further, the target population is highly specific and has limited availability, which prevents the use of probabilistic sampling methods. Instead, we used a combination of *convenience sampling* and *snowball sampling*. Convenience sampling means that we reached out to the target population via the authors' combined networks using emails and personal messages on LinkedIn.[3] We sent them a personalized invitation, written in a neutral tone, followed by a reminder a few weeks later, as suggested by Kitchenham and Pfleeger (2008).

To increase the reach of the survey beyond the authors' networks, we applied snowball sampling in two different ways. First, by encouraging those who we invited to take the survey to forward it to other practitioners. However, we instructed them to only forward the invitation to people working on different real-time systems to avoid Threat 4 and Threat 5. Second, we used snowball sampling to mitigate a geographical bias towards contacts from Europe where most of the authors' networks reside (to avoid Threat 8 and Threat 9). We asked 20 academics, primarily based in North America, South America, and Asia, to forward the invitation to members of the target population in their networks. (A separate survey link was created to allow curious academics to open the survey and try out the questions, while making it possible to filter out such responses, and so avoid impacting the results, mitigating Threat 4).

[3] https://www.linkedin.com/.

In convenience sampling, we selected known industry contacts with substantial real-time systems experience, who we expected would be able to provide concrete answers to the questions, representative of the systems in their company's portfolio. We anticipated that these contacts would understand the utility of the survey, and thus be interested in diligently completing it, and in seeing the results. We aimed to avoid selecting contacts working in the same departments, and limited selections within any one company. Overall, the authors directly invited 114 industry contacts, which, including snowball sampling, resulted in 90 respondents starting the survey. Invitations via the 20 academic contacts resulted in a further 30 respondents starting the survey. Of the 120 respondents starting the survey, 97 completed it. Due to snowball sampling, we do not know the exact response rate, i.e. the total number completing the survey divided by the total number invited to take it.

3.3 Further discussions on threats to validity

In this subsection, we focus on threats to internal, external, and conclusion validity (construct validity having been covered in the previous subsection).

Internal validity: To reduce the effects of personal bias (Threat 7), we did not ask questions where the answers depended on personal opinion. We also formulated our propositions in Sect. 5 based on quantitative measures, i.e, statistical analysis, rather than qualitative ones that could be influenced by personal opinion. To reduce the risk of including foreign units (Threat 4), we did not advertise the survey via public channels. Further, the welcome page explicitly asked only those who consider themselves part of the real-time embedded systems industry to complete the survey (Threat 3). To reduce the risk of reliance of personal experience (Threat 6), we only invited, via convenience sampling, participants who were considered to have sufficient experience and competence to answer the survey questions; however, those subsequently invited via snowball sampling may not necessarily have fulfilled that criteria.

External validity: The survey suffers from sampling bias as a result of practical limitations in finding industry practitioners through the author's networks and the networks of their close academic contacts. This impacts generalization and statistical inference, since they are only valid for a population that the sample data is representative of. This population is undoubtedly not "the real-time embedded systems industry as a whole" but rather some portion of it that can be described as follows:

Effective population: *industry practitioners actively developing real-time embedded systems who have first- or second-order links to academic real-time systems researchers.*

As evidenced by the responses to Question 30 in Sect. 5, over 80% of respondents interact with the real-time research community, by reading articles, participating in conferences and research projects, and reviewing papers. Further, the responses to Question 1 indicate that only 15% of respondents are from small companies (< 100 employees). Similarly, there is a large variation in the number of respondents per application domain (see Question 4). Hence, the survey results may be more

representative of the automotive, avionics and consumer electronics industries than of healthcare or space.

Even though the effective population does not cover the whole real-time embedded systems industry, it represents the potential first-hand industry clients for the work published by the real-time systems research community, as evidenced by the responses to Questions 28 and 30, which ask about the number of published papers the respondents read and their type of interactions with the research community.

The survey results, observations, propositions, and statistical inferences are therefore useful in the context of understanding the state of practice, needs and trends in companies developing real-time embedded systems, who interact with the real-time systems research community and are interested in exploiting its research results.

Conclusion validity: To isolate and contain the potential impact of Threat 10, we separate the results that are based purely on observations about the sample data, in Sect. 4, from the propositions that we infer about the effective population using statistical analyses in Sect. 5. Further, to reduce Threat 10 to the statistical analyses used, we have explained our use of statistical inference in Sect. 5, taking into account issues regarding the misinterpretation of confidence intervals (Greenland et al. 2016). Finally, we were careful to avoid the pitfalls of collecting data and then retroactively searching for correlations. With a limited sample size and many possible post-hoc hypotheses (i.e. that answer X to one question correlates with answer Y to another question) this can easily result in false results that appear statistically significant due to random variation.[4] Rather, we only present information about statistically significant differences with respect to the sub-groups identified via Questions 4 and 5. Our working hypothesis, based on prior knowledge, was that different sets of requirements are applied by different industry sectors and by those developing safety-critical systems, which can result in differences in system design and development. Hence, where there is a statistically significant difference based on these subgroups, there is also a reasonable case for a causal link.

4 Results

This section lists all of the survey questions, in the order in which they appeared in the survey, along with graphs of the results, and our observations. The survey was divided into a number of topics, which are separated by horizontal lines in the text below. Where results are given as percentages, unless otherwise stated, these correspond to the proportion of respondents who selected that specific option out of all of the respondents who answered that particular question. The graphs presenting the results are color-coded. Dark red bars are used for questions with distinct alternatives, and hence the total sums to 100%. Light blue bars are used where respondents were asked to "select all options that apply", and hence the percentages sum to more

[4] For a mildly humorous overview of this *multiple comparisons problem* see https://orangedatamining.com/blog/2019/1/4/how-to-abuse-p-values-in-correlations/.

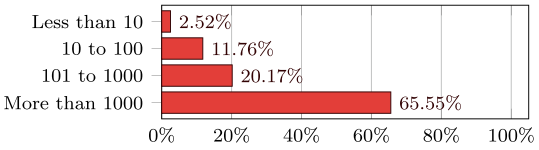
than 100%. Multi-colored bars (e.g. Question 6) indicate the percentage of respondents who selected the corresponding scores or rankings. Where the answers have an ordering (e.g. Question 1), the results are presented in that order. Otherwise, they have been re-ordered with the most popular answer first for ease of reference. Nevertheless, "I do not know" and "Other" are always placed last.

Our observations include a commentary on the results, and a more in-depth look at the data. In some cases, we comment on the results for sub-groups that have been identified via the answers to Question 4 (Avionics, Automotive, and Consumer Electronics) and Question 5 (safety-critical components, and no safety-critical components). We *only* comment on the difference in results between these sub-groups where these differences are statistically significant at the $p < 0.05$ level.[5] Finally, the number X of respondents answering each question is given as ($\mathbf{n = X}$) at the right hand side of each question box.

The aggregated data from the survey is available online (Akesson et al. 2020a), including a breakdown of the information for each of the main sub-groups, Avionics, Automotive, and Consumer Electronics.

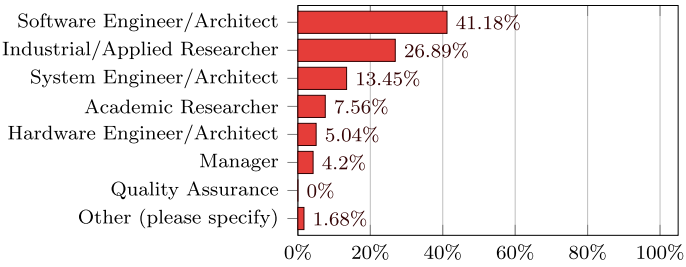
Demographics: This part of the survey asked questions about the respondent's organization and professional experience.

Question 1 How many employees does your organization have? ($\mathbf{n = 120}$)



Observations: Approximately two thirds of the respondents were from large companies (> 1000 employees), with around one third from small and medium sized enterprises (SMEs).

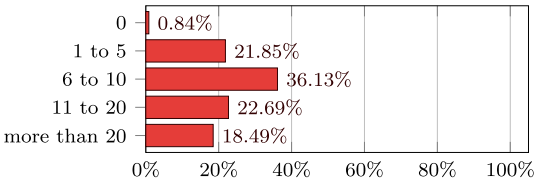
Question 2 Which position best describes your current role in your organization? ($\mathbf{n = 120}$)



[5] According to the analysis provided by the SurveyMonkey toolset, see https://help.surveymonkey.com/articles/en_US/kb/Significant-Differences.

Observations: Approximately 60% of the respondents were directly involved in system development (software, system, or hardware), while approximately 27% were involved in industrial research. The category "Academic Researcher" includes staff on secondment to industry, and staff who recently moved from industry to academia.

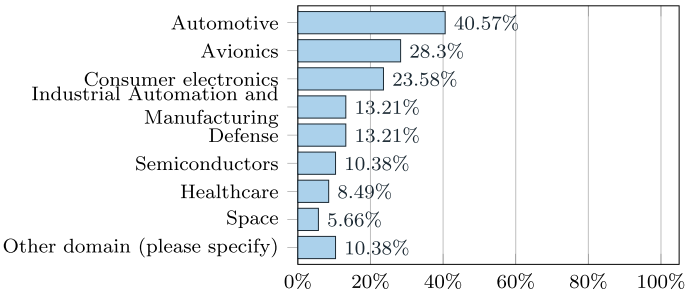
Question 3 How many years of industrial experience do you have? (**n = 120**)



Observations: The majority of respondents had many years of industrial experience, with 41% having more than 10 years experience, and only 23% having five years or less.

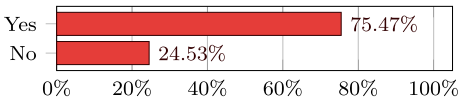
System context: This part of the survey asked questions about hardware, software, and the execution of the system. Respondents were asked to think about a particular system where they were familiar with these aspects, and to consider the same system for all questions to ensure consistent responses.

Question 4 To what domain(s) does the considered system belong? (**n = 107**)



Observations: The survey has broad coverage of the different application domains. Note, that multiple domains could be selected. The largest overlaps were between: Avionics and Defense 9.4%, Automotive and Industrial Automation 6.6%, Automotive and Consumer Electronics 5.7%, Automotive and Avionics 5.7%, and Space and Defense 4.7%. Automotive alone was indicated by 65% of those selecting that domain, similarly, Avionics alone by 60% of those selecting that domain, and Consumer Electronics alone by 56% of those selecting that domain. Of the 11 respondents who indicated "Other domain", five specified "Telecomms." i.e. 4.7%.

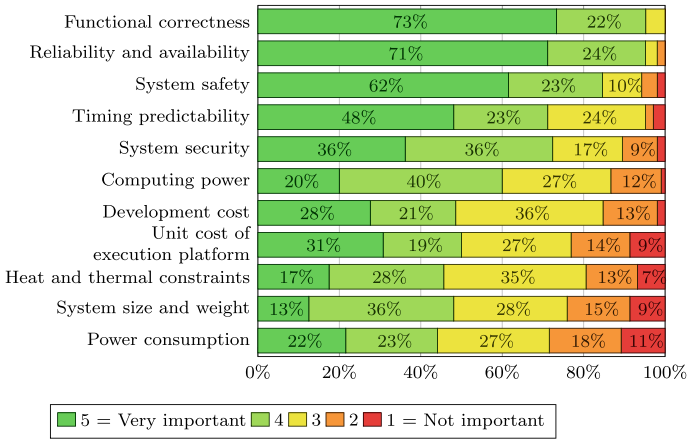
Question 5 Is (parts of) the considered system safety-critical? (**n = 107**)



Observations: Even though the response to Question 4 indicates broad domain coverage, a large majority (75%) of the systems considered had some part that was safety critical.

Of those respondents who selected Avionics in Question 4, 100% answered "Yes" to this question, compared to 91% of those who selected Automotive, and just 52% of those who selected Consumer Electronics.

Question 6 Give a score to the importance of different system aspects for the considered system. (**n = 107**)

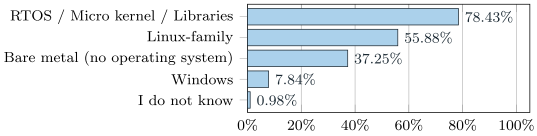


Observations: Timing predictability, although viewed as less important than functional correctness, reliability, and safety, was seen as more important than security, computing power, cost, and thermal considerations. (This is perhaps unsurprising, since the survey targeted those working on real-time systems).

Of the respondents who selected Avionics in Question 4, 87% thought that timing predictability was very important, compared to 48% of those who selected Automotive, and just 26% of those who selected Consumer Electronics. In contrast, unit cost of the execution platform was rated as very important by 45% of those respondents who selected Automotive in Question 4, 32% of those who selected Consumer Electronics, and just 7% of those who selected Avionics.

Hardware platform: This part of the survey asked questions about the hardware and software configurations of the considered system. Here, most questions allowed the selection of multiple options to capture the characteristics of complex systems with many components. Respondents were asked to select all options that apply to the system they were considering.

Question 7 What Operating Systems are running on the considered system? (**n = 103**)

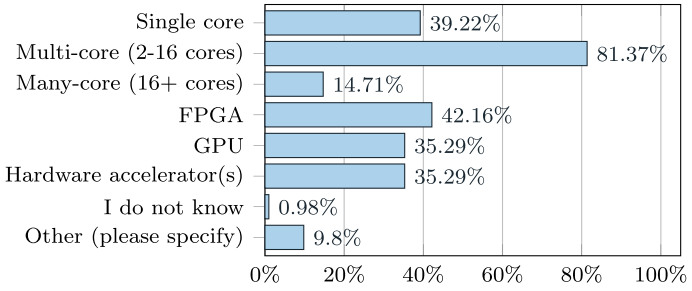


Observations: While 78% of respondents indicated that some parts of their system use an RTOS or Micro kernel, a significant minority (37%) had parts that use no Operating System (OS) at all. RTOS alone was selected by 22.5%, Linux alone by 7.8%, and Bare metal alone by 4.9%. None of the respondents used Windows alone. There were many systems that used more than one OS (62.7%). The largest overlaps were between RTOS and Linux (42.2%), Bare metal and RTOS (28.4%), and Bare metal and Linux (17.6%). The combination of Bare metal, RTOS, and Linux was used by 14.7% of respondents.

Of the respondents who indicated in Question 5 that their system contained some safety-critical components, 87% used an RTOS. This figure reduced to 50% of those who indicated no safety-critical components. By contrast, the corresponding figures for the use of Windows were 3% and 25%, respectively.

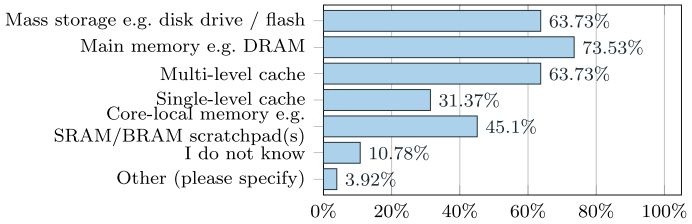
As an optional addition to this question, respondents were asked to name the operating systems that they were using. 32 respondents did so, many citing multiple operating systems. The following lists the operating systems named and the number of times they were cited: Autosar (8), QNX (8), VXWorks (4), OSEK (3), Redhat Linux (3), Free RTOS (2), Linux (2), PikeOS (2), Ubuntu (2), Yocto Linux (2), Arinc-653 (1), DEOS (1), EmbOS (1), Erika (1), Integrity (1), LynxOS (1), RTEMS (1), SafeRTOS (1), ThreadX (1), Windows (1), and Zephyr (1).

Question 8 Select the options that describe the processing hardware of the considered system. (**n = 103**)



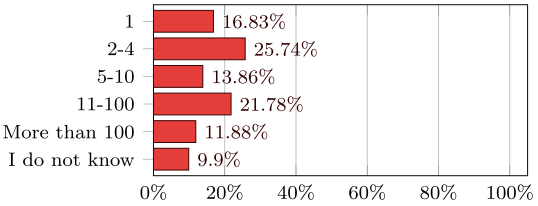
Observations: The majority of systems (81%) include multi-core components, while just under 40% include single core components. Similarly, 35% to 42% of systems include FPGAs, GPUs, and other hardware accelerators. Of the 10 respondents who indicated "Other", three specified DSP (i.e. 2.9%) and two specified System-on-Chip (i.e. 2.0%).

Question 9 Select the options that describe the memory hierarchy of the considered system. (**n = 103**)



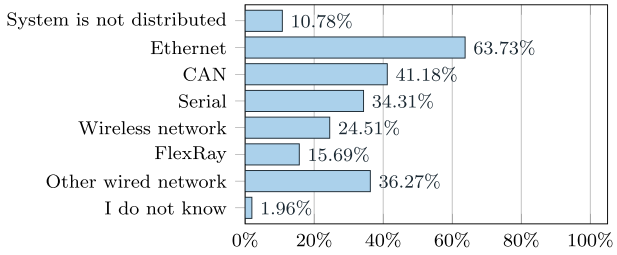
Observations: The majority of systems (over 63%) have elements of a complex memory hierarchy including mass storage devices, DRAM, and multiple levels of cache. Core local memory and single level caches are also prevalent.

Question 10 How many distributed nodes (e.g. ECUs) are there in the considered system? (**n = 102**)



Observations: The majority of systems are distributed (73%), with only 17% identified as having a single node (ECU).

Question 11 Which of the following options describe the connectivity within the (distributed) system? (**n = 103**)



Observations: Wireless networks were used in around 25% of systems, with Ethernet (64%) and CAN (41%) the most popular forms of wired network. Many systems (48%) used multiple types of wired network, with 34.3% using Ethernet and CAN, 27.5% Ethernet and Serial, 19.6% CAN and Serial, and 14.7% Ethernet, CAN, and Flexray. 9.8% of systems used Ethernet as the only wired network, while less

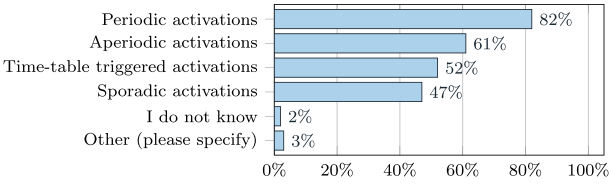
than 3% used CAN, Flexray, or Serial alone. Wireless was used as the only network by 8.8% of respondents, about one third of those using that technology.

Of the respondents who selected Automotive in Question 4, 74% used CAN, and 34% used Flexray, this reduced to 21% and 3% for those who selected Avionics. Flexray was only used by one respondent (1%) who did not select Automotive.

There was some inconsistency in what was understood by one node or ECU in Question 10 (17%) and what was understood by "not distributed" (11%) in Question 11. This could be because respondents were considering "Nodes" or "ECUs" in Question 10, and "connectivity" in Question 11. (A single node or ECU may contain multiple connected processing units).

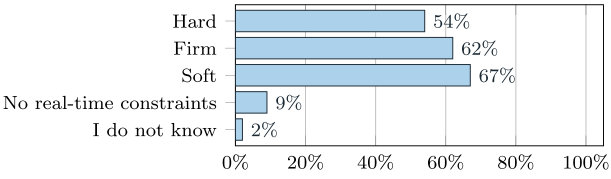
Timing Characteristics: This part of the survey asked questions about the timing characteristics of the considered system.

Question 12 Which of the following sentences are true about task activations in your system? (**n = 101**)



Observations: While periodic activation is the most common at 82%, over 60% of systems included aperiodic activations. 22% of responses indicated highly predictable behaviors (utilizing either periodic or time triggered activation) with no sporadic or aperiodic tasks, while 4% (and 2%) of respondents indicated purely sporadic (aperiodic) activations with no time-triggered or periodic tasks. Interestingly, 74% of respondents indicated at least two, and 25% all four types of activations.

Question 13 Which of the following timing constraints exist(s) in your system? (**n = 101**)

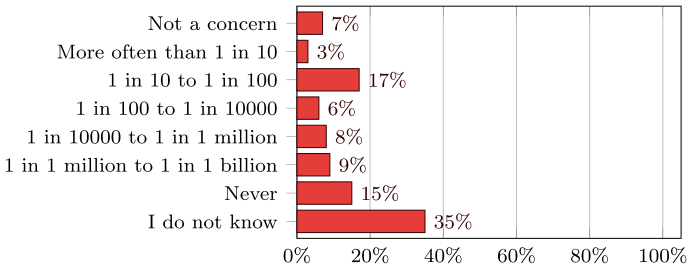


Note, a more detailed explanation of the terms used was provided in the survey. Hard implies that violating the timing constraint is considered a failure of the system. Firm implies that violating the timing constraint is highly undesirable. Soft means that occasionally violating the timing constraint is acceptable, but negatively impacts the perceived quality of the system.

Observations: Given the scope of the survey, it is unsurprising that just under 90% of respondents indicated that their system had some form of timing constraints. Many systems (62%) had a combination of two or more different constraints: Hard and Firm 38%, Hard and Soft 36%, Firm and Soft 42%, and all three 27%. In contrast, far fewer systems had only one type of timing constraint: Hard 5%, Firm 10%, and Soft 15%.

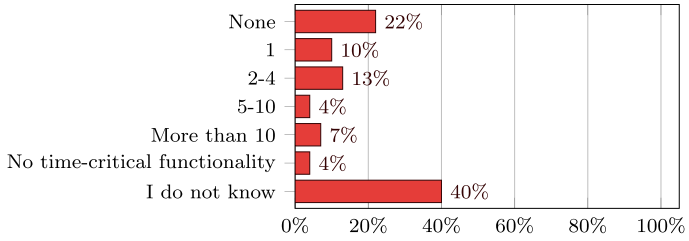
Of the respondents who selected Avionics in Question 4, 79% indicated Hard constraints, compared to 56% of those who selected Automotive, and only 27% of those who selected Consumer Electronics. Of the respondents who indicating in Question 5 that their system contained some safety-critical components, 64% indicated Hard constraints. This reduced to 21% of those who indicated no safety-critical components.

Question 14 For the most time-critical functions in the system, roughly how frequently can the deadline of a function be missed without causing a system failure. (**n = 101**)



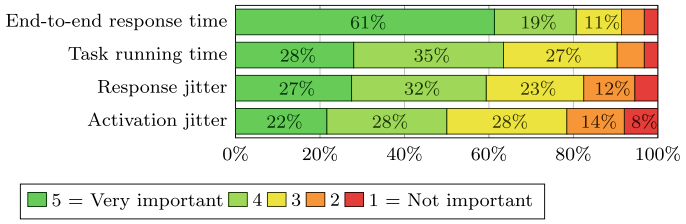
Observations: A substantial number of respondents (35%) were unable to give a specific answer to this question, and answered "I do not know". Only a small proportion (15%) of systems were considered strictly hard real-time, with deadlines that must never be missed. By contrast, 45% of respondents indicated that the most time critical functions in the system could miss some deadlines, and 20% indicated that deadline misses more often than 1 in 100 could be tolerated.

Question 15 What is the largest number of consecutive deadline misses that could be tolerated, assuming that such a blackout does not reoccur for a very long time. (**n = 101**)



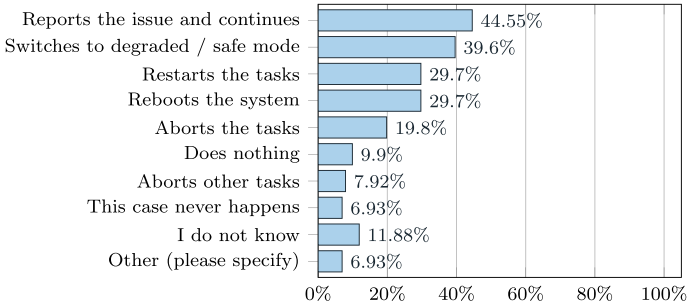
Observations: The responses to this question follow a similar pattern to those of Question 14, with 34% of respondents indicating that the system can tolerate blackout periods in the range of 1 to more than 10 deadline misses. Here, only about 60% of respondents were able to give a specific answer, with 40% answering "I do not know".

Question 16 What are relevant timing constraints to your system? (**n = 99**)



Observations: End-to-end response time was considered the most important timing constraint, with the largest percentage of "very important" scores and the highest average score of 4.3. However, task running time (3.78), response jitter (3.64) and activation jitter (3.42) also need to be considered. 72.7% of respondents rated end-to-end response time highest or equal highest. For task running time, response jitter, and activation jitter, this was the case for 45.5%, 35.4%, and 32.3% of respondents, respectively.

Question 17 How does the considered system react if tasks miss deadlines? (**n = 102**)



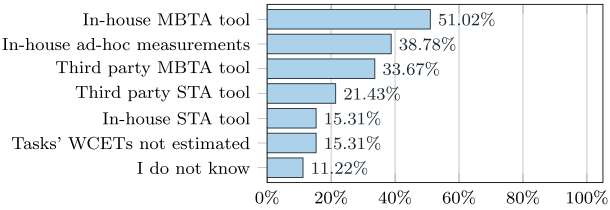
Observations: The most common (45%) reaction to a missed deadline is to report the issue and continue, while 10% of systems do nothing. Other systems take actions on a deadline miss, including 30% rebooting, and 30% restarting tasks. Further, although 15% said that a deadline miss may never occur (Question 14) only 7% trust their system enough to state that "This case never happens".

Of the respondents who indicated in Question 5 that their system contained some safety-critical components, 36% indicated "Reboots the system". This reduced to

just 8% of those who indicated no safety-critical components. By contrast, the figures for "Does nothing" were 6% and 21%, respectively.

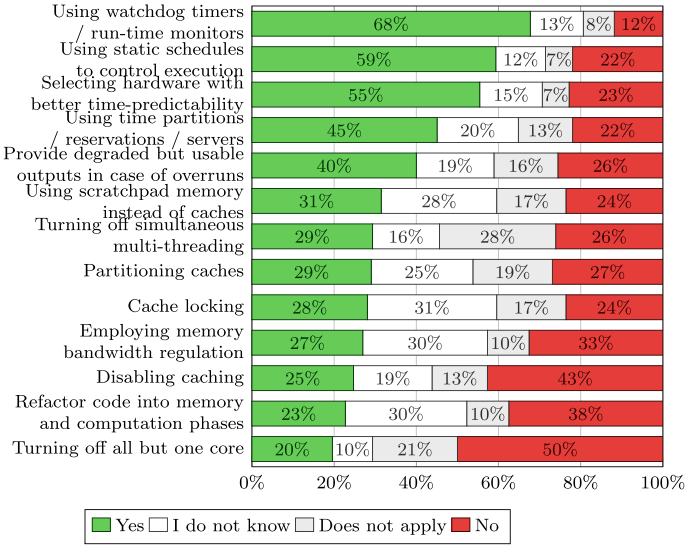
Managing timing behavior: This part of the survey asked questions about the methods used to analyze and influence the timing behavior of the system.

Question 18 Which methods are used for Worst-Case Execution Time (WCET) estimation in the considered system? (**n = 99**)



Observations: Measurement-Based Timing Analysis (MBTA) tools are used by substantially more respondents than Static Timing Analysis (STA) tools, with more than 50% using in-house MBTA tools compared to 15% for in-house STA tools. This distinction in less stark when it comes to third party solutions with 34% using third party MBTA tools and 21% using third party STA tools. Overall, 67.4% of respondents used some form of MBTA tool, 33.7% used some form of STA tool, and 23.5% used both.

Question 19 What steps are taken to help increase timing predictability? (**n = 97**)

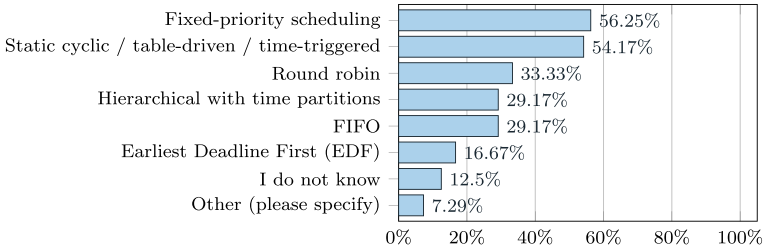


Observations: While more than 50% of respondents use watchdog timers, static scheduling, and appropriate hardware selection, it is clear that there is no "silver bullet" to improving timing predictability. Each of the wide range of different

techniques is used by at least 20% of respondents, and 46% of respondents answered "Yes" to at least 5 of the techniques listed. Some of the techniques that are least frequently employed are, however, those that have the largest impact on average-case execution times (e.g. disabling caching and turning off all but one core).

There was considerable uncertainty in answering parts of this question, reflected in approximately 30% of respondents answering "I don't know" with respect to the use of scratchpads, cache locking, memory bandwidth regulation, and refactoring code into memory and computation phases.

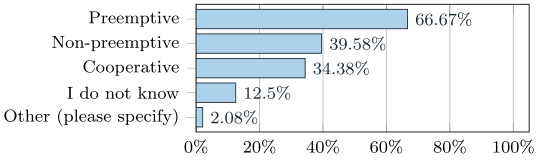
Question 20 Which task scheduling policy/policies are used in the considered system? (**n = 97**)



Observations: The most popular scheduling policies were fixed priority and static cyclic/table driven, with each used by more than half of the respondents. Round-robin and FIFO, which are not traditionally viewed as real-time policies, were employed in around 30% of systems, while EDF was employed in less than 17% of systems, less than one third as often as fixed priority scheduling.

Of the respondents who selected Automotive in Question 4, 27% used EDF scheduling, compared to just 3% of those who selected Avionics, and 11% of those who selected Consumer Electronics.

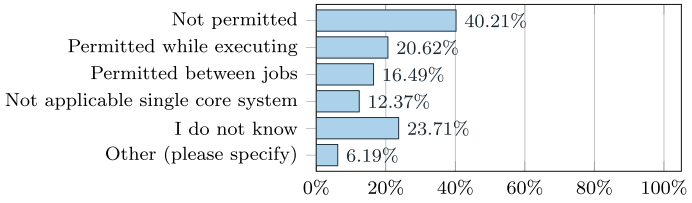
Question 21 Indicate the types of preemption that are supported in the considered system. (**n = 97**)



Note, a more detailed explanation of the terms used was provided in the survey. Preemptive implies that task execution can be preempted by other tasks at any time, non-preemptive implies that task execution cannot be preempted by other tasks before completion, and cooperative means that task execution can be preempted by other tasks, but only at predefined preemption points.

Observations: While preemptive scheduling is the most popular choice, used in two thirds of systems, both non-preemptive and co-operative scheduling are used in more than one third of systems.

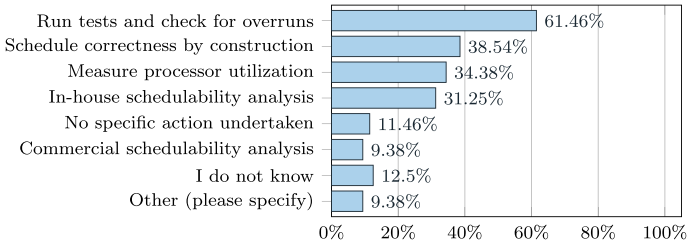
Question 22 Indicate how task migration can take place between different cores in the considered system. (**n = 98**)



Observations: Although timing predictability is typically easier to achieve without task migration, the proportion of systems permitting migration (37%) is similar to the proportion that do not permit it (40%).

Of the respondents who selected Avionics in Question 4, only 7% indicated that task migration is permitted while the task is executing. By comparison, this figure was 27% for those who selected Automotive, and 30% for those who selected Consumer Electronics.

Question 23 How do you ensure that the functions in the considered system respect their deadlines? (**n = 97**)



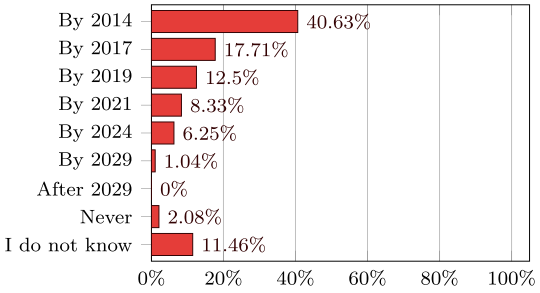
Observations: Less than 10% of respondents are using commercial schedulability analysis tools, while more than 30% use in-house solutions. The main off-line approach is schedule correctness by construction, using a static schedule and checking that execution time budgets hold. However, the most common approach overall is to run tests and check for overruns (61%), with a similar proportion to those that use watchdogs timers/run-time monitors (see Question 19).

None of the respondents who selected Avionics in Question 4, indicated "no specific action undertaken", compared to 12% of those who selected Automotive, and 16% of those who selected Consumer Electronics. Of the respondents who indicating in Question 5 that their system contained some safety-critical components, 7%

answered “no specific action undertaken”. The corresponding figure was 29% of those who indicated no safety-critical components.

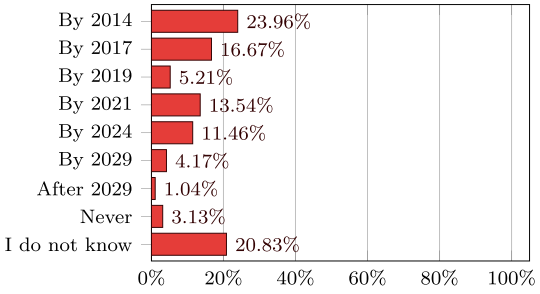
Timelines for hardware adoption: This part of the survey asked questions about timelines for hardware adoption.

Question 24 By which year did or do you expect development projects for real-time embedded systems in your department to begin using multi-core embedded processors (i.e. processors with 2 to 16 cores)? (**n = 97**)



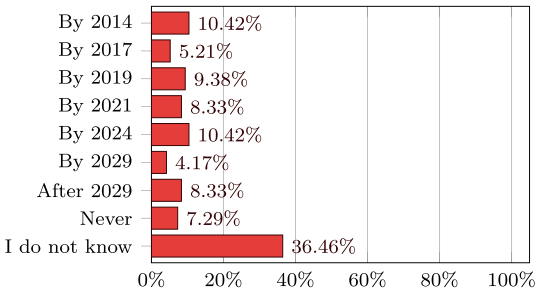
Observations: Multi-core systems are already widely used in current developments, with 80% of respondents indicating their use by 2021, and only 10% answering “I do not know”.

Question 25 By which year did or do you expect development projects for real-time embedded systems in your department to begin using heterogeneous multi-cores with different types of CPUs, GPUs, and other accelerators? (**n = 97**)



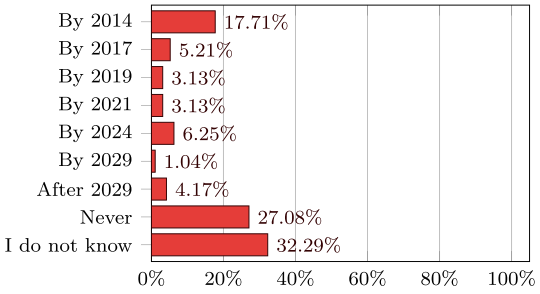
Observations: The uptake of more complex multi-core systems lags behind simpler multi-core systems (Question 24), but nevertheless just under 60% of respondents indicate their use by 2021, with 20% answering “I do not know”.

Question 26 By which year did or do you expect development projects for real-time embedded systems in your department to begin using many-core embedded processors (i.e. processors with more than 16 cores)? (**n = 97**)



Observations: The uptake of many-core systems is less certain, with 36% of respondents answering "I do not know", 33% indicating take up by 2021, and 48% take up by by 2029.

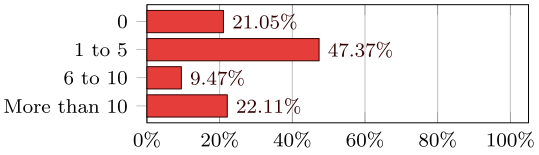
Question 27 By which year did or do you expect new development projects for real-time embedded systems in your department to stop using single-core embedded processors (i.e. processors with one core)? (**n = 97**)



Observations: Although the proportion of respondents expecting to use single-core devices drops in future years, a substantial minority (31%) still expect to use single cores after 2029. Interestingly, this is the case for respondents who indicated each of the Automotive, Avionics, and Consumer Electronics domains in Question 4, with 30%, 34.5%, and 30%, respectively, expecting to use single-cores after 2029.

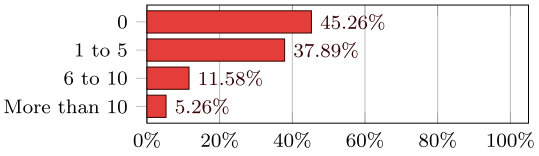
Familiarity with real-time systems research: This part of the survey asked questions about familiarity with the real-time systems research community and its results.

Question 28 How many research publications (e.g. conference or journal papers) in the real-time systems field have you read in the last year? (**n = 96**)



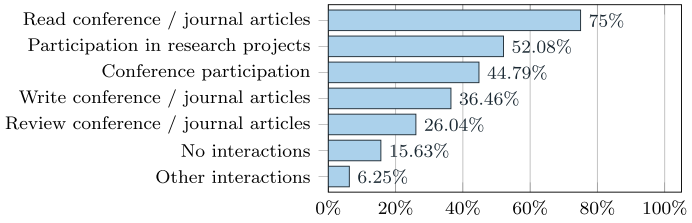
Observations: Around 79% of respondents read at least one research publication in the past year.

Question 29 How many real-time systems research publications (e.g. conference or journal papers) have you published as a (co-)author in the last 5 years? (**n = 96**)



Observations: Around 55% of respondents contributed to research publications in the past 5 years.

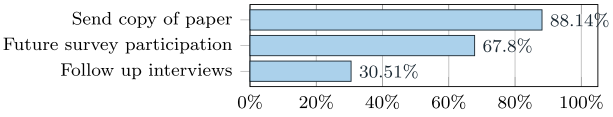
Question 30 How do you interact with the real-time research community? (**n = 97**)



Observations: Only 16% of respondents have no interactions with the real-time research community. “Other interaction” (6%) included: research internships, co-supervisions, and interacting with researchers directly.

Follow Up: The final part of the survey asked questions about following up on this survey and general remarks.

Question 31 Indicate the purposes for which we may contact you again, if any. (**n = 60**)



Observations: 48 respondents provided their email addresses for subsequent follow up.

Question 32 Enter feedback or remarks (**n = 23**)

Observations: 23 respondents provided feedback. The most common comments were complementary remarks about the survey and a desire to see the results.

5 Analysis and discussion

In this section, we present the results of using a standard statistical tool to generalize our findings by estimating parameters (e.g. the proportion using some feature *F*) for the population from which the sample data was taken. Using *statistical inference*, we derive the *confidence intervals* of our main findings at a *confidence level* γ of 95%. Confidence intervals provide a useful estimate of population parameters, since their calculation tends to produce intervals that contain the true value of the parameter. There are a number of common misconceptions about confidence intervals (Greenland et al. 2016). For example, it is not correct to assume that there is a probability of γ (e.g. 95%) that the confidence interval will actually contain the true parameter value. Rather, confidence intervals are such that if the sampling process were repeated a large number of times, then the true value of the population parameter would be expected to fall within the confidence intervals computed for those samples γ (e.g. 95%) of the time (Neyman and Jeffreys 1937). The confidence interval represents a range of values for the population parameter for which the difference between the parameter and the estimate from the sample is not statistically significant at the $(1-\gamma)$ level (Cox and Hinkley 1979). Hence, if the true value does not fall within the confidence interval, then it means that a sampling event has occurred that had a probability of $(1-\gamma)$ (e.g. 5%) or less of happening by chance.

Below, we revisit the five objectives set out in Sect. 1. For each objective, we use statistical inference to extend the results of the survey to the *effective population* (see Sect. 3.3). For each objective, we list a set of *propositions*. Each proposition is expressed as a statement (in bold) that is derived from generalizations of the survey results that follow. Each generalization gives the data from the sample, followed by a confidence interval for the population parameter (e.g. "*C* (count) of *S* (sample size), CI [L% to U%] (confidence interval), use feature *F*"). The confidence intervals were calculated assuming a 95% confidence level using the StatKey online tool[6] and selecting "CI for single proportion". The confidence intervals were computed excluding "I do not know" responses, since we assumed that such responses to factual questions were genuine attempts by respondents to complete the survey to the best of their knowledge.

Objective O1 Establish whether timing predictability is of concern to the real-time embedded systems industry.

[6] http://www.lock5stat.com/StatKey/index.html.

Proposition 1 Although timing predictability is important, it is only one of many system design aspects (**Question** 6).

97 *of* 105, *CI* [87%, 97%], *consider timing predictability to be no more important than functional correctness*.

96 *of* 105, *CI* [86%, 96%], *than reliability and availability*.

90 *of* 105, *CI* [79%, 92%], *than system safety*.

69 *of* 105, *CI* [56%, 75%], *than system security*.

54 *of* 105, *CI* [42%, 61%], *than development cost*.

53 *of* 105, *CI* [40%, 60%], *than unit cost*.

48 *of* 105, *CI* [36%, 55%], *than size and weight*.

40 *of* 105, *CI* [30%, 48%], *than power consumption*.

Objective O2 Identify relevant industrial problem contexts, including hardware platforms, middleware, and software.

Proposition 2 Hardware platforms are complex and distributed (**Questions** 8, 9, 10, 11).

91 *of* 101, *CI* [84%, 96%], *use multi-cores or many-cores*.

68 *of* 101, *CI* [57%, 77%], *have FPGAs and/or GPUs and/or hardware accelerators*.

48 *of* 91, *CI* [42%, 63%], *include mass storage*, *main memory*, *and multi-level caches*.

65 *of* 100, *CI* [56%, 74%], *use two or more types of network*.

48 *of* 91, *CI* [42%, 63%], *include* 5 *or more distributed nodes*.

Proposition 3 Multiple different types of Operating System (OS) are used, often within the same system (**Question** 7).[7]

80 *of* 101, *CI* [71%, 87%], *use an RTOS*.

[7] Note most systems are distributed and have multiple nodes.

38 *of* 101, *CI* [28%, 48%], *use bare metal* (*i.e. no OS*).

57 *of* 101, *CI* [46%, 67%], *use Linux*.

60 *of* 101, *CI* [49%, 69%], *use at least two of*: *bare metal*, *RTOS*, *Linux in the same system*.

Proposition 4 Deadlines are not sacrosanct (**Questions** 14, 15).

44 *of* 66, *CI* [54%, 79%], *consider that the most time critical functions in their systems can miss some deadlines*.

20 *of* 66, *CI* [19%, 40%], *can miss deadlines more often than* 1 *in* 100.

24 *of* 57, *CI* [29%, 55%], *can tolerate two or more consecutive deadline misses*.

Proposition 5 Different types of timing constraints are present in the same system (**Question** 13).[8]

62 *of* 98, *CI* [54%, 73%], *work on systems with a mix of at least two different types of timing constraint* (*i.e. hard*, *firm*, *and soft*).

Objective O3 Determine which methods and tools are used to achieve timing predictability.

Proposition 6 Measurement-based timing analysis is more prevalent than static timing analysis, but both are used (**Question** 18).

66 *of* 87, *CI* [66%, 85%], *use measurement-based timing analysis*.

33 *of* 87, *CI* [27%, 49%], *use static timing analysis*,

23 *of* 87, *CI* [17%, 36%], *use both*.

Proposition 7 Both static and dynamic methods of improving timing predictability are widely used (**Question** 19).

76 *of* 95, *CI* [71%, 88%], *use at least one static approach* (*e.g. static schedules*, *time partitions*, *turn off multi-threading*, *partitioned caches*, *cache locking*, *disable caching*, *refactor code into memory and computation phases*, *turn off all but one core*).

74 *of* 95, *CI* [69%, 87%], *use at least one dynamic mechanism* (*e.g. watchdog timers*, *degraded outputs on overrun*, *memory bandwidth regulation*, *and run-time monitors*).

[8] Note most systems are distributed and have multiple nodes.

Proposition 8 Systems often take mitigating actions in the event of timing violations (**Question** 17).

40 *of* 89, *CI* [34%, 56%], *switch to degraded/safe mode*.

44 *of* 89, *CI* [39%, 60%], *abort or restart tasks*.

30 *of* 89, *CI* [23%, 44%], *reboot the system to mitigate missed deadlines*.

Proposition 9 Some systems use only highly predictable task activation patterns (**Question** 12).

21 *of* 99, *CI* [13%, 30%], *use exclusively periodic and/or time-triggered forms of task activation*.

Objective O4 Establish which techniques and tools are used to satisfy real-time requirements.

Proposition 10 Many different scheduling policies are used in the same system[7], some of which are not "real-time" (**Question** 20).

54 *of* 84, *CI* [53%, 74%], *use Fixed Priority scheduling*,

52 *of* 84, *CI* [51%, 73%], *use static cyclic scheduling*,

32 *of* 84, *CI* [28%, 49%], *use Round-robin*,

28 *of* 84, *CI* [23%, 44%], *use FIFO*,

16 *of* 84, *CI* [10%, 28%], *use EDF*.

56 *of* 84, *CI* [57%, 77%], *use at least two of the above policies in the same system*.

Proposition 11 Many different preemption strategies are used in the same system[7] (**Question** 21).

65 *of* 85, *CI* [67%, 85%], *use preemptive scheduling*.

56 *of* 85, *CI* [55%, 77%], *use cooperative and/or non-preemptive scheduling*.

36 *of* 85, *CI* [31%, 53%], *use both preemptive and cooperative/non-preemptive scheduling in the same system*[7].

28 *of* 85, *CI* [23%, 43%], *use exclusively preemptive scheduling*.

19 *of* 85, *CI* [14%, 32%], *use exclusively cooperative and non-preemptive scheduling*.

Proposition 12 Some systems permit task migration between cores (Question 22).

13 *of* 58, *CI* [12%, 33%], *always permit task migration either between or during jobs*.

17 *of* 58, *CI* [17%, 42%], *do not permit any form of task migration*.

30 *of* 58, *CI* [39%, 64%], *permit migrations for some parts and do not permit task migrations for other parts*.

Proposition 13 The most common way to verify timing requirements is to run tests and check for overruns (Question 23).

37 *of* 84, *CI* [33%, 55%], *use static schedules*.

37 *of* 84, *CI* [33%, 55%], *use schedulability analysis tools*.

59 *of* 84, *CI* [60%, 80%], *run tests and check for overruns*.

Objective O5 Determine trends for future real-time systems.

Proposition 14 Multi-core and complex heterogeneous multi-core processors are being adopted, as are many-cores (Questions 24, 25, 26).

77 *of* 86, *CI* [82%, 96%], *expect to use multi-cores by* 2021,

57 *of* 77, *CI* [63%, 84%], *expect to use complex heterogeneous multi-cores by* 2024, *and*

32 *of* 62, *CI* [38%, 65%], *expect to use many-cores by* 2029.

Proposition 15 Single-cores continue to be used (Question 27).

30 *of* 65, *CI* [33%, 59%], *expect to still be using single-cores after* 2029.

The results of the survey show that many respondents work for companies that are active in multiple application domains. This real-world complexity prevented a fully stratified analysis, comparing and contrasting the characteristics of different application domains.

6 Are the survey findings common knowledge?

During the review process for the preliminary version of this survey (Akesson et al. 2020b), it was suggested that the survey findings might be common knowledge among the real-time systems community. In the best selling book *Factfulness*, Rosling-Ronnlund et al. (2018) note that when asked simple questions about global or aggregate information, experts typically systematically get the answers wrong; often so wrong that they can be outperformed by random guesswork. Reflecting on this, we sought to determine if the results of this survey represent common knowledge among the real-time systems community. (Specifically two cohorts: academic researchers and industry practitioners, since we were also interested in finding out if there were any significant differences between the two groups). If the survey information was already well-known, then that would diminish its value. Alternatively, if the information was not well-known or worse the antithesis was somehow assumed, then that would highlight the value of this empirical survey-based research to the community.

To test the hypothesis that the real-time systems community already has substantial up-to-date knowledge of industry practice, we devised a simple quiz comprising 13 multiple choice questions, each of which could be answered correctly based on the results presented in Sect. 4. We selected quiz questions covering roughly half of the survey questions, with a focus on those survey questions where there was little ambiguity. For example, we did not ask quiz questions about the findings of Questions 14 and 15, since those questions had a large number of "I don't know" responses. There was a maximum of one quiz question for any given survey question. Each quiz question had three possible options to select from, hence selecting answers at random would be expected to score 33.3%, or on average 4.33 correct answers, with the number of correct answers following a binomial distribution.

We designed the quiz questions to avoid issues of bias that could be caused by differences in the distribution of practitioners from different industry domains responding to the survey. To achieve this, we choose questions where there was no statistically significant difference in answers between the domains considered in the survey. Further, we set the multiple choice answers to the quiz questions, so that small differences between reality and the survey results would not change which answers to the quiz questions were correct. This also prevents minimal misconceptions of reality from resulting in incorrect answers. Finally, we ensured that the correct answers to the quiz questions were always at the upper or lower end of the three options given. This was done to enable an examination of how often respondents selected the option furthest from the correct answer.

We considered two cohorts for the quiz: academic researchers and industry practitioners. As a representative subset of expert real-time systems researchers, we chose the Technical Program Committee (TPC) members from the past three editions of the top-tier real-time systems conferences: RTSS, RTAS, and ECRTS. We obtained the names of the TPC members from the conference websites, and found their email addresses either from our personal contacts or online. We

filtered the list removing any industry practitioners, thus leaving 183 academic researchers. We emailed these researchers asking them to complete the quiz to the best of their ability, using the knowledge that they had in their heads, i.e. not looking anything up. Out of 68 academic respondents, 21 had previously read the preliminary version of the survey, or had seen its presentation. We excluded these academics from the results, since we wanted to examine prior knowledge rather than test memory of the survey or presentation. This left 47 academics who had not seen the survey. We used the same approach to contacting industry practitioners that we had previously used for the survey, and asked them to complete the quiz. Out of 53 respondents, 16 had previously read the preliminary version of the survey, or had seen its presentation. We excluded these practitioners from the results, leaving 37 who had not seen the results of the survey. (Note, although these practitioners may have contributed to the survey individually, they were not aware of the overall aggregate results on which the quiz questions and answers were based).

Table 2 gives the overall quiz results by cohort, with the distribution of scores illustrated in Fig. 1. The primary data from the quiz has been made available in aggregate form (i.e. the distribution of answers to each question by cohort) for others to use (Akesson et al. 2020a).

Observation 1: Academic researchers fared worse than random guesswork on the quiz, with an average of 3.51 correct answers to the 13 questions (27%), while random guesswork averages 4.33 correct answers (33.3%). Industry practitioners fared slightly better, with an average of 4.57 correct answers (35.1%). Further, academic researchers chose the middle option, which was incorrect but closer to the correct answer, 43% of the time, and the least correct option 30% of the time. For industry practitioners, these figures were 33.9% and 31% respectively. Again, random guesswork would give 33.3%.

Observation 2: Taking a *wisdom of crowds* (Surowiecki 2004) approach, choosing the answer to each question that was most popular, resulted in a score of 2 out of 13 for the sample of academic researchers. The most popular answer given by academic researchers was the furthest from being correct in 5 out of 13 cases, and the least popular answer was correct in 7 out of 13 cases. Industry practitioners fared better with a wisdom of crowds score of 4 out of 13. The most popular answer given by the sample of industry practitioners was the furthest from being correct in 3 out of 13 cases, and the least popular answer was correct in 2 out of 13 cases.

Observation 3: None of the quiz questions were answered correctly by 50% or more of the academic researchers, while 3 were answered correctly by 50% or more of the industry practitioners. Further, only 3 questions were answered correctly by 33.3% or more of the academic researchers, while 5 were answered correctly by 33.3% or more of the industry practitioners.

Table 2 Summary of quiz results by cohort

	Academic researchers	Industry practitioners	Random guesses
Average score	3.51 (27%)	4.57 (35.1%)	4.33 (33.3%)
Most popular answer is correct	2/13	4/13	–
Most popular answer is middle option	6/13	6/13	–
Most popular answer is least correct	5/13	3/13	–
Least popular answer is correct	7/13	2/13	–
Least popular answer is middle option	0/13	5/13	–
Least popular answer is least correct	6/13	6/13	–
Questions with $\geq$ 50% correct answers	0/13	3/13	–
Questions with $\geq$ 33% correct answers	3/13	5/13	–

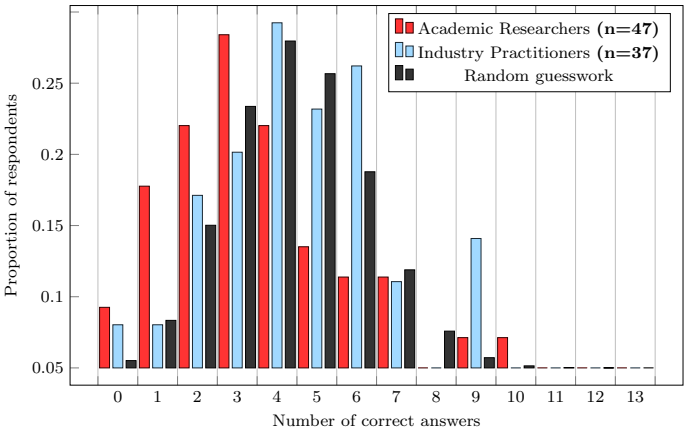


Fig. 1 Distribution of the number of correct answers to the quiz questions for three cohorts: academic researchers, industry practitioners, and random guesswork

Observation 4: Quiz questions 6, 8, and 9 were the most difficult ones and answered correctly by fewer than 20% of both academic researchers and industry practitioners. The correct answers to these questions could therefore be regarded as the most surprising.

Observation 2 is particularly striking, since it implies that taking a poll of academic respondents opinions on the aggregate picture of the state of industry practice and trends in real-time systems design and development, and then applying a *contrarian* view that the least popular answer is the one that is most likely correct would result in a much better match to the survey results (7/13 correct) than giving credence to the actual poll results themselves (2/13 correct).

Clearly, there is a substantial gap between the ideal of perfect scores on the quiz, and how the samples of academic researchers and industry practitioners performed. Below, we discuss some of the possible reasons for this.

- *Out-of-date world-view*: Just like health experts (Rosling-Ronnlund et al. 2018), both academic researchers and industry practitioners working on real-time systems establish their *world-view* of industry practice in real-time systems over time. This happens via direct experience working on industry projects or with industry partners, and indirect experience reading research papers, listening to keynotes, and discussing real-time systems with their peers. This world-view evolves over time, and may only be updated relatively slowly with respect to what was first learnt many years ago. In the meantime, the world moves forward in a way that seems to be incremental when observed on an annual basis, but which nevertheless results in large changes over a time frame of decades.
- *Specialist vs. aggregate knowledge*: Most academic research papers are about a specific aspect of a real-time system. This is typically a sub-problem, such as the scheduling of tasks on processors. The context for such work is only a small part of the reality of complex, distributed, industrial real-time systems. With this specialized view, systems may seem simpler and less varied than they actually are. Industry practitioners may have excellent in-depth knowledge of their own specific systems and other similar ones in their domain; however, similar to academics, before this survey was produced, there was no easy way for them to obtain a thorough quantitative overview of the entire field.
- *Survey as a valid baseline*: It might be argued that the survey results are not a valid baseline.

  It is true that the results of this survey are not yet a well-established fact, and that there could be issues with just how representative the sample of industry practitioners that responded to the survey was with respect to the population of such practitioners as a whole. That said, we took steps, set out in Sect. 3, to mitigate threats to validity in the survey design. We also selected the quiz questions to avoid issues of bias that could be caused by differences in the distribution of practitioners in different industry domains, and set the multiple choice options so that small differences between reality and the survey results would not change which answers were correct. Further, we reached out to the same set of industry practitioners for the quiz as we did for the survey, hence the industry practitioners who responded to the quiz are likely to be strongly correlated with, and representative of, those who contributed to the survey.
- *Quiz not taken seriously*: Could respondents have rushed through the quiz giving essentially random answers? The distribution of times spent answering the quiz questions indicates that respondents are likely to have taken the quiz seriously, and tried to answer the questions as requested, i.e. to the best of their ability without looking anything up. The distribution of overall times was as follows: 25-percentile: 4 min 51 s, median: 7 min 10 s, 75-percentile: 10 min 3 s. The minimum time was 2 min 33 s, and at the upper end of the range there were a small number of times that were over 20 min, indicative of respondents leaving the quiz open while they focused on something else. To assess whether the minimum time was realistic, the authors read through the quiz and made decisive choices. The times obtained indicated that the minimum of 2 min 33 s although fast, is not an unreasonable time for valid completion of the quiz. Considering only those 15 respondents that took less than 4 minutes the mean score was 30%

and the standard deviation was 16%, whereas for the remaining respondents that took 4 minutes or more the mean score was 31% and the standard deviation was 17%, indicating that the time taken to complete the quiz was not a significant factor in the scores obtained.

The results of the quiz are consistent with the view put forward in the *Factfulness* book (Rosling-Ronnlund et al. 2018), namely that experts may have excellent specific knowledge about their own specialities and systems, but when asked about the state of practice and trends in terms of an aggregate view, they are unable to provide reliable information. This paper aims at addressing that problem.

7 Conclusions

An absence of any systematic studies into industry practice increases the risk that academic research will diverge from areas that are crucial to the development of future industrial systems. This may lead to research that is less relevant, less likely to be adopted, and has a lower potential for impact. While empirical survey-based research is well-established in software and system engineering, there were previously no such studies in the real-time systems field. This paper addresses that omission by presenting the results of a survey, containing 32 questions related to methods, tools, and trends in industrial real-time systems development. The survey was completed by 120 industry practitioners from a variety of different organizations, countries, and application domains.

The survey results show that industry recognizes the importance of timing predictability, but that other design aspects are of equal or greater importance, such as functional correctness and reliability/availability. Hence, it is important for real-time systems research to be cognizant of its impact on these aspects.

Many real-time systems today are distributed systems that use multi-core processors, and have complex memory hierarchies. Further, multiple different operating systems and networking technologies are typically utilized within the same system, as are different types of timing constraints and task activation patterns. Many respondents did not consider timing constraints to be sacrosanct, with even the most time-critical functions allowed to miss some deadlines.

There is no silver bullet to manage timing behavior in complex real-time systems. Instead, the survey reveals that a wide range of different tools, techniques, and policies are used for timing analysis, scheduling, and to increase timing predictability. There is no one size fits all solution.

The trends suggest that single-core systems are still widely used today, and are expected to remain relevant for new developments for at least the next ten years. However, more complex (heterogeneous) multi- and many-core systems are already prevalent and their adoption is expected to increase significantly during the 2020s.

The results of the quiz, discussed in Sect. 6, show that the *aggregate* findings of the survey are not common knowledge among industry practitioners or academic researchers, with the average scores of each cohort differing little from what could be achieved via random guesswork. Hence, the survey fulfills a very

important and necessary purpose, to help bring members of the real-time systems community up-to-date in terms of their aggregate view of current industry practice in real-time systems design and development. This knowledge is valuable in: (i) motivating further research on both new and old topics; (ii) assessing where research is likely to have wide-ranging and lasting impact; and (iii) shaping the landscape for future research funding. We hope that this survey will encourage others in the community to engage in further empirical research in this area, including replication studies of this work.

Finally, we would like to end by encouraging members of the academic real-time systems research community to absorb and interpret the information presented in this survey in the context of their specific research topics, and reflect on how they can address the variety and complexity of future industrial real-time systems in their own research. With this purpose in mind, the primary data from the survey has been made available in aggregate form for others to use (Akesson et al. 2020a). Readers may also be interested in viewing a video presentation of this survey available at https://www.akesson.nl/files/videos/akesson20-rtss_video.mp4 and also the associated industry panel discussion https://www.akesson.nl/files/videos/akesson20-rtss_panel.mp4.

Appendix: Quiz questions

This appendix reproduces the quiz, including instructions, questions, and finally a list of the answers. Note, the version used online (https://www.surveymonkey.com/r/quiz_real-time) had some additional demographic questions that have been omitted here.

Industry Practice in Real-time Systems Quiz

These quiz questions and their answers are based on the findings in the paper "An Empirical Survey-based Study into Industry Practice in Real-time Systems" published in RTSS 2020. You may take the quiz regardless of whether you have read the paper/seen the presentation or not. We simply ask that you complete the quiz fairly, to the best of your ability, using the information and knowledge that you have in your head, i.e. without referring to the survey paper, or looking up any answers. Please complete all questions. Note that answers will be recorded anonymously and the results will be used in a research paper in an aggregated form only. There are 14 questions in total, and the quiz should typically take about 7-10 minutes to complete. The answers will be provided once you have completed all of the questions.

Question 1 Have you read the paper "An Empirical Survey-based Study into Industry Practice in Real-time Systems" published in RTSS 2020 or viewed its presentation?

- Yes
- No

In the following when we refer to "real-time systems", we mean the set of real-time systems that the industry practitioners surveyed in the paper "An Empirical

Survey-based Study into Industry Practice in Real-time Systems", published in RTSS 2020, work on (or expect to be working on, for questions involving future trends). Note the industry practitioners came from a variety of sectors, including automotive, avionics, consumer electronics, industrial automation, semiconductors, healthcare, space, and telecommunications.

Question 2 Approximately what proportion of the real-time systems include processors that run software "bare metal" i.e. without an operating system?

A 10%
B 20%
C 40%

Question 3 What is the ratio between the number of real-time systems that contain multi-core processors and those that contain single-core processors?

A About twice as many contain multi-cores as contain single cores
B The prevalence of multi-cores and single cores is about the same
C About twice as many contain single cores as contain multi-cores

Question 4 Approximately what proportion of the real-time systems have elements of a complex memory hierarchy (e.g. mass storage devices, DRAM, multiple levels of cache)?

A 20%
B 40%
C 60%

Question 5 Approximately what proportion of the real-time systems are distributed systems (i.e. made up of processors connected via external communication links (i.e. wired or wireless networks or buses)?

A 30%
B 60%
C 90%

Question 6 How many of the different task scheduling policies, listed below in order of prevalence, are used either alone or in combination with other policies in at least 25% of the real-time systems?

(1) Static Cyclic,
(2) Fixed Priority,
(3) Round Robin,
(4) Hierarchical with Time Partitions,
(5) FIFO,

(6) EDF.

A 1
B 3
C 5

Question 7 Approximately what proportion of the real-time systems have only hard real-time constraints?

A 5%
B 15%
C 45%

Question 8 There are two main methods used for Worst-Case Execution Time (WCET) estimation in the real-time systems: Measurement Based Timing Analysis (MBTA) and Static Timing Analysis (STA). What is the ratio of use in the real-time systems between the two?

A About four times as many use MBTA as use STA
B About three times as many use MBTA as use STA
C About twice as many use MBTA as use STA

Question 9 How many of the different methods of improving timing predictability, listed below in order of prevalence, are used either alone or in combination with other methods in at least 25% of the real-time systems?

(1) Using watchdog timers/run-time monitors,
(2) Using static schedules to control execution,
(3) Selecting hardware with better time-predictability,
(4) Using time partitions/reservations/servers,
(5) Provide degraded but usable outputs in case of overruns,
(6) Using scratchpad memory instead of caches,
(7) Turning off simultaneous multi-threading,
(8) Partitioning caches,
(9) Cache locking,
(10) Employing memory bandwidth regulation,
(11) Disabling caching,
(12) Refactor code into memory, and computation phases,
(13) Turning off all but one core.

A 3
B 7
C 11

Question 10 Approximately what proportion of the industry practitioners expect development projects for real-time embedded systems in their department to be using multi-cores by 2021?

A 40%
B 60%
C 80%

Question 11 Approximately what proportion of the industry practitioners expect development projects for real-time embedded systems in their department to be using heterogeneous multi-cores (with different types of CPUs, GPUs, and other accelerators) by 2021?

A 20%
B 40%
C 60%

Question 12 What is the latest year by which at least 30% of industry practitioners expect development projects for real-time embedded systems in their department to still be using single core processors?

A 2021
B 2024
C 2029

Question 13 How many of the different types of task activation listed below are used in at least 40% of the real-time systems?

(1) Periodic activations,
(2) Aperiodic activations,
(3) Time-table triggered activations, and
(4) Sporadic activations.

A 2
B 3
C 4

Question 14 Approximately what proportion of the real-time systems use both preemptive and non-preemptive/cooperative task scheduling?

A 10%
B 25%
C 40%

The answers to the quiz questions can be found in this footnote[9]

Acknowledgements The research described in this paper is funded, in part, by the Innovate UK HICLASS Project (113213) and by the Netherlands Organisation for Applied Scientific Research TNO. The authors would like to thank all of the respondents who took the time and effort to complete the survey and also to provide feedback on it, and also those who completed the quiz. Without them, this paper would not have been possible.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Akesson B, Nasri M, Nelissen G, Altmeyer S, Davis RI (2020a) Data set real-time systems survey. https://doi.org/10.21942/uva.13117709.v1

Akesson B, Nasri M, Nelissen G, Altmeyer S, Davis RI (2020b) An empirical survey-based study into industry practice in real-time systems. In: IEEE real-time systems symposium (RTSS 2020)

Audsley NC, Burns A, Davis RI, Tindell KW, Wellings AJ (1995) Fixed priority pre-emptive scheduling: an historical perspective. Real-Time Syst 8(2–3):173–198. https://doi.org/10.1007/BF01094342

Broy M, Kirstan S, Krcmar H, Schätz B, Zimmermann J (2012) What is the benefit of a model-based design of embedded software systems in the car industry? In: Emerging technologies for the evolution and maintenance of software models, pp 343–369

Burns A, Davis RI (2017) A survey of research into mixed criticality systems. ACM Comput Surv 50(6):1–37

Buttazzo GC, Bertogna M, Yao G (2013) Limited preemptive scheduling for real-time systems. a survey. IEEE Trans Ind Inform 9(1):3–15

Cox DR, Hinkley DV (1979) Theoretical statistics. Chapman and Hall, New York

Davis RI, Burns A (2011) A survey of hard real-time scheduling for multiprocessor systems. ACM Comput Surv 43(4):1–44. https://doi.org/10.1145/1978802.1978814

Davis RI, Cucu-Grosjean L (2019a) A survey of probabilistic schedulability analysis techniques for real-time systems. LITES: Leibniz Transactions on Embedded Systems, pp 1–53

Davis RI, Cucu-Grosjean L (2019b) A survey of probabilistic timing analysis techniques for real-time systems. LITES: Leibniz Transactions on Embedded Systems, pp 1–60

Forward A, Lethbridge TC (2008) Problems and opportunities for model-centric versus code-centric software development: a survey of software professionals. In: Proceedings of the 2008 international workshop on Models in software engineering, ACM, pp 27–32

Greenland S, Senn SJ, Rothman KJ, Carlin JB, Poole C, Goodman SN, Altman DG (2016) Statistical tests, p values, confidence intervals, and power: a guide to misinterpretations. Eur J Epidemiol 31(4):337–350. https://doi.org/10.1007/s10654-016-0149-3

[9] All of the answers to the quiz questions are C, except for questions 3 and 7, where A is correct.

Hermans F, Pinzger M, Van Deursen A (2009) Domain-specific languages in practice: a user study on the success factors. In: International conference on model driven engineering languages and systems, Springer, pp 423–437

Höfer A, Tichy WF (2007) Status of empirical research in software engineering. In: Empirical software engineering issues. Critical assessment and future directions, Springer, pp 10–19

Hutchinson J, Whittle J, Rouncefield M (2014) Model-driven engineering practices in industry: social, organizational and managerial factors that lead to success or failure. Sci Comput Program 89:144–161

Kitchenham BA, Pfleeger SL (2008) Personal opinion surveys. In: Guide to advanced empirical software engineering, Springer, pp 63–92

Kuhn A, Murphy GC, Thompson CA (2012) An exploratory study of forces and frictions affecting large-scale model-driven development. In: International conference on model driven engineering languages and systems, Springer, pp 352–367

Liebel G, Marko N, Tichy M, Leitner A, Hansson J (2018) Model-based engineering in the embedded systems domain: an industrial survey on the state-of-practice. Softw Syst Model 17(1):91–113

Maiza C, Rihani H, Rivas JM, Goossens J, Altmeyer S, Davis RI (2019) A survey of timing verification techniques for multi-core real-time systems. ACM Comput Surv 52(3):1–38

Mohagheghi P, Dehlen V (2008) Where is the proof? A review of experiences from applying MDE in industry. Lecture Notes Comput Sci 5095:432–443

Muller G (2013) Systems engineering research methods. Procedia Comput Sci 16:1092–1101

Neyman J, Jeffreys H (1937) Outline of a theory of statistical estimation based on the classical theory of probability. Philos Trans R Soc Lond Ser A 236(767):333–380. https://doi.org/10.1098/rsta.1937.0005

Rosling-Ronnlund A, Rosling H, Rosling O (2018) Factfulness: ten reasons we're wrong about the world—and why things are better than you think. Flatiron Books, https://www.gapminder.org/factfulness-book/

Sha L, Abdelzaher T, Årzén KE, Cervin A, Baker T, Burns A, Buttazzo G, Caccamo M, Lehoczky J, Mok AK (2004) Real time scheduling theory: a historical perspective. Real-Time Syst 28(2–3):101–155. https://doi.org/10.1023/B:TIME.0000045315.61234.1e

Singh AK, Dziurzanski P, Mendis HR, Indrusiak LS (2017) A survey and comparative study of hard and soft real-time dynamic resource allocation strategies for multi-/many-core systems. ACM Comput Surv 50(2):1–40. https://doi.org/10.1145/3057267

Surowiecki J (2004) The wisdom of crowds: why the many are smarter than the few and how collective wisdom shapes business, economies, societies and nations. Doubleday Anchor, New York

Torchiano M, Tomassetti F, Ricca F, Tiso A, Reggio G (2013) Relevance, benefits, and problems of software modelling and model driven techniques—a survey in the Italian industry. J Syst Softw 86(8):2110–2126

van der Sanden B, Li Y, van den Aker J, Akesson B, Bijlsma T, Hendriks M, Triantafyllidis K, Verriet J, Voeten J, Basten T (2021) Model-driven system-performance engineering for cyber-physical systems. In: Embedded systems week (ESWEEK)

Vetro A, Bohm W, Torchiano M (2015) On the benefits and barriers when adopting software modelling and model driven techniques—an external, differentiated replication. In: 2015 ACM/IEEE international symposium on empirical software engineering and measurement (ESEM), IEEE, pp 1–4

Whittle J, Hutchinson J, Rouncefield M (2014) The state of practice in model-driven engineering. IEEE Softw 31(3):79–85

Whittle J, Hutchinson J, Rouncefield M, Burden H, Heldal R (2017) A taxonomy of tool-related issues affecting the adoption of model-driven engineering. Softw Syst Model 16(2):313–331

Wilhelm R, Engblom J, Ermedahl A, Holsti N, Thesing S, Whalley D, Bernat G, Ferdinand C, Heckmann R, Mitra T et al (2008) The worst-case execution-time problem—overview of methods and survey of tools. ACM Trans Embed Comput Syst 7(3):36

Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in software engineering. Springer, New York

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

B. Akesson received his Ph.D. from Eindhoven University of Technology, the Netherlands in 2010. Since then, he has been employed as a Researcher at Eindhoven University of Technology, Czech Technical University in Prague, and CISTER/INESC TEC Research Unit in Porto. He is a Professor by Special Appointment at the University of Amsterdam since 2019. Currently, Prof. Akesson is working as a Senior Research Fellow at ESI (TNO) in Eindhoven. His research interest is design methodologies for cyber-physical systems, in particular model-based engineering and real-time systems.

M. Nasri is an Assistant Professor in the Electronic Systems Group at the Eindhoven University of Technology, the Netherlands. She received her Ph.D. from the University of Tehran, in 2015. Before that, she was an assistant professor at the Delft University of Technology, the Netherlands, a postdoc fellow at the Max Planck Institute for Software Systems, Germany, and a postdoc researcher at TU-Kaiserslautern, Germany. Her research interests include scheduling, timing analysis, and model-based design of real-time systems.

G. Nelissen is an Assistant Professor in the IRIS group of the Mathematics and Computer Science department of the Eindhoven University of Technology (TU/e), the Netherlands. Prior to joining TU/e, Geoffrey was a researcher at the Polytechnic Institute of Porto where he was a member of the CISTER research unit. He received his Ph.D. from the Université Libre de Bruxelles (ULB), Belgium in 2013. His research interests span all theoretical and practical aspects of real-time embedded systems design, including the analysis and configuration of real-time parallel applications on multicore and distributed platforms.

S. Altmeyer is Full Professor at the University of Augsburg, Germany and head of the Chair for Embedded Systems. He received his Ph.D. from Saarland University, Germany in 2012. Before he joined the University of Augsburg, he has been employed at the University of Amsterdam, The Netherlands, and the University of Luxembourg. Sebastian's research aims at the design and development of safe, reliable and performant embedded systems. This includes in particular real-time scheduling, worst-case execution time analysis and processor architectures for real-time systems.

R. I. Davis is a Reader in the Real-Time Systems Research Group at the University of York, UK. Robert received his Ph.D. in Computer Science from the University of York in 1995. Since then he has founded three start-up companies, all of which have succeeded in transferring real-time systems research into commercial products. Robert's research interests include real-time scheduling and schedulability analyses for single-core, multi-core, and networked systems.