



Quality Control in Crowdsourcing Systems

Issues and Directions

**Mohammad Allahbakhsh, Boualem Benatallah,
and Aleksandar Ignjatovic** • *University of New South Wales*

Hamid Reza Motahari-Nezhad • *Hewlett-Packard Laboratories*

Elisa Bertino • *Purdue University*

Schahram Dustdar • *Vienna University of Technology*

As a new distributed computing model, crowdsourcing lets people leverage the crowd's intelligence and wisdom toward solving problems. This article proposes a framework for characterizing various dimensions of quality control in crowdsourcing systems, a critical issue. The authors briefly review existing quality-control approaches, identify open issues, and look to future research directions.

Crowdsourcing has emerged as an effective way to perform tasks that are easy for humans but remain difficult for computers.^{1,2} For instance, Amazon Mechanical Turk (MTurk; www.mturk.com) provides on-demand access to task forces for micro-tasks such as image recognition and language translation. Several organizations, including DARPA and various world health and relief agencies, are using platforms such as MTurk, CrowdFlower (<http://crowdflower.com>), and Ushahidi (<http://ushahidi.com>) to crowdsource work through multiple channels, including SMS, email, Twitter, and the World Wide Web. As Internet and mobile technologies continue to advance, crowdsourcing can help organizations increase productivity, leverage an external (skilled) workforce in addition to a core workforce, reduce training costs, and improve core and support processes for both public and private sectors.

On the other hand, the people who contribute to crowdsourcing might have different levels of skills and expertise that are sometimes

insufficient for doing certain tasks.³ They might also have various and even biased interests and incentives.^{1,4} Indeed, in recent years, crowdsourcing systems have been widely subject to malicious activities such as collusive campaigns to support people or products, and fake reviews posted to online markets.⁵ Additionally, ill-defined crowdsourcing tasks that don't provide workers with enough information about the tasks and their requirements can also lead to low-quality contributions from the crowd.⁶ Addressing these issues requires fundamentally understanding the factors that impact quality as well as quality-control approaches being used in crowdsourcing systems.

Categorizing Quality Control

To crowdsource a task, its owner, also called the *requester*, submits the task to a crowdsourcing platform. People who can accomplish the task, called *workers*, can choose to work on it and devise solutions. Workers then submit these *contributions* to the requester via the crowdsourcing platform.

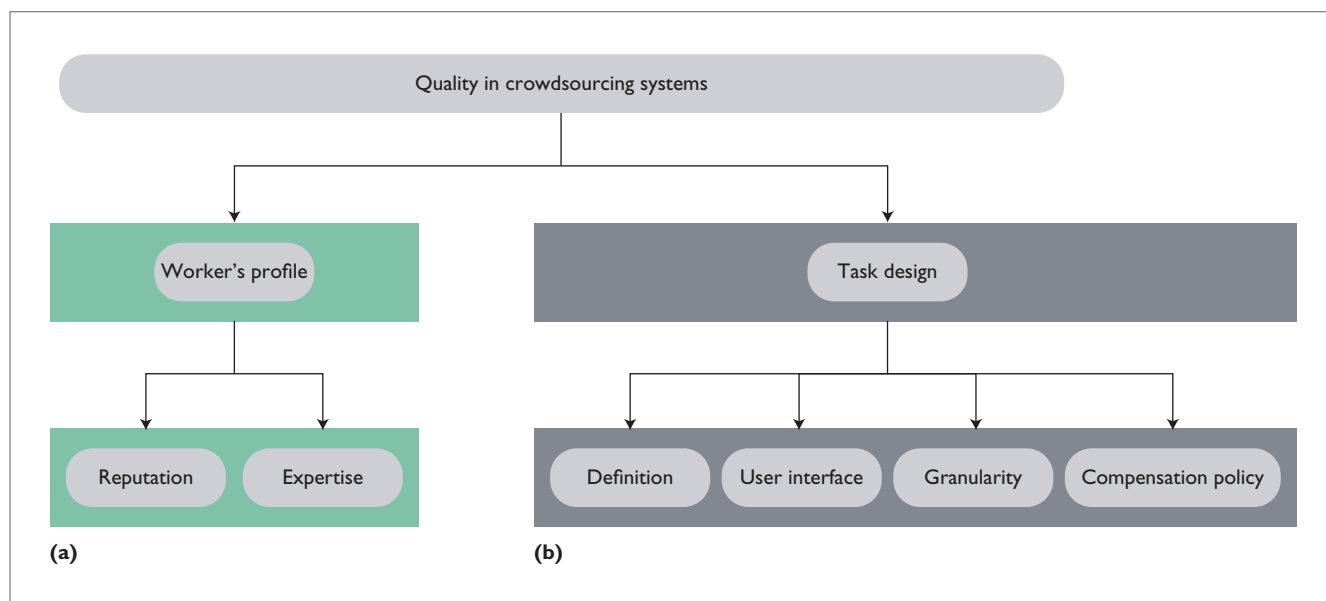


Figure 1. Taxonomy of quality in crowdsourcing systems. We characterize quality along two main dimensions: (a) worker profiles and (b) task design.

The requester assesses the posted contributions' quality and might reward those workers whose contributions have been accepted. This reward can be monetary, material, psychological, and so on.⁷ A task's outcome can be one or more individual contributions or a combination of accepted ones. The requester should choose contributions that reach his or her accepted level of quality for the outcome.

Quality is a subjective issue in general. Some efforts have proposed models and metrics to quantitatively and objectively assess quality along different dimensions of a software system, such as reliability, accuracy, relevancy, completeness, and consistency.⁸ In this survey, we adopt Crosby's definition of quality as a guide to identify quality-control attributes, including dimensions and factors.⁹ This definition emphasizes "conformance to requirements" as a guiding principle to define quality-control models. In other words, we define the quality of outcomes of a crowdsourced task as

"the extent to which the provided outcome fulfills the requirements of the requester."

The overall outcome quality depends on the definition of the task that's being crowdsourced and the contributing workers' attributes.^{1,2} We characterize quality in crowdsourcing systems along two main dimensions: *worker profiles* and *task design*. We propose a taxonomy for quality in crowdsourcing systems, as Figure 1 illustrates.

Worker Profiles

The quality of a crowdsourced task's outcome can be affected by workers' abilities and quality.² As Figure 1a shows, a worker's quality is characterized by his or her reputation and expertise. Note that these attributes are correlated: a worker with high expertise is expected to have a high reputation as well. We distinguish them because reputation is more general in nature. In addition to workers' expertise (which is reflected in the quality of their contributions), we might compute reputation based on several other parameters, such as the worker's timeliness or the quality of evaluators. Also, reputation is a public, community-wide metric, but expertise is task-dependent. For example, a Java expert with a high

reputation score might not be qualified to undertake a SQL-related task.

Reputation. The trust relationship between a requester and a particular worker reflects the probability that the requester expects to receive a quality contribution from the worker. At the community level, because members might have no experience or direct interactions with other members, they can rely on reputation to indicate the community-wide judgment on a given worker's capabilities.¹⁰

Reputation scores are mainly built on community members' feedback about workers' activities in the system.¹¹ Sometimes, this feedback is explicit – that is, community members explicitly cast feedback on a worker's quality or contributions by, for instance, rating or ranking the content the worker has created. In other cases, feedback is cast implicitly, as in Wikipedia, when subsequent editors preserve the changes a particular worker has made.

Expertise. A worker's expertise demonstrates how capable he or she is at doing particular tasks.⁴ Two types of indicators point to worker

expertise: credentials and experience. Credentials are documents or evidence from which the requesters or crowdsourcing platform can assess a worker's capabilities as regards a particular crowdsourced task. Information such as academic certificates or degrees, spoken languages, or geographical regions that a worker is familiar with can be credentials. Experience refers to knowledge and skills a worker has gained while working in the system as well as through support and training. For instance, in systems such as MTurk and Stack Overflow, workers can improve their skills and capabilities over time with shepherding and support.¹²

Task Design

Task design is the model under which the requester describes his or her task; it consists of several components. When the requester designates a task, he or she provides some information for workers. The requester might put a few criteria in place to ensure that only eligible people can do the task, or specify the evaluation and compensation policies. We identify four important factors that contribute to quality as regards this dimension (see Figure 1b): task definition, user interface, granularity, and compensation policy.

Task definition. The task definition is the information the requester gives potential workers regarding the crowdsourced task. A main element is a short description of the task explaining its nature, time limitations, and so on.⁶ A second element is the qualification requirements for performing the task. These specify the eligibility criteria by which the requester will evaluate workers before accepting their participation. For example, in MTurk, requesters can specify that only workers with a specified percentage of accepted works (for example, larger than 90 percent) can participate, or that only

those workers living in the US can take part in a particular survey. Previous studies show that the quality of the provided definition (such as its clarity or the instructions' usefulness) for a task affects outcome quality.⁶

User interface. The task UI refers to the interface through which the workers access and contribute to the task. This can be a Web UI, an API, or any other kind of UI. A user-friendly interface can attract more workers and increase the chance of a high-quality outcome. A simple interface, such as one with nonverifiable questions, makes it easier for deceptive workers to exploit the system.¹ On the other hand, an unnecessarily complicated interface will discourage honest workers and could lead to delays.

Granularity. We can divide tasks into two broad types: simple and complex. Simple tasks are the self-contained, appropriately short tasks that usually need little expertise to be solved, such as tagging or describing.¹³ Complex tasks usually need to be broken down into simpler subtasks. Solving a complex task (such as writing an article) might require more time, costs, and expertise, so fewer people will be interested or qualified to perform it. Crowds solve the subtasks, and their contributions are consolidated to build the final answer.¹³ A complex task workflow defines how these simple subtasks are chained together to build the overall task.¹⁴ This workflow can be iterative, parallel, or a combination.^{14,15}

Designing workflows for complex tasks greatly affects outcome quality.^{1,2,5} For instance, one study demonstrated that designing a poor outline for an essay that the crowd will write can result in a low-quality essay.¹³ Improving the quality of an outline using crowd contributions increases the corresponding written essay's quality.

Incentives and compensation policy.

Choosing suitable incentives and a compensation policy can affect the crowd's performance as well as outcome quality.^{7,12} Knowing about evaluation and compensation policies helps workers align their work based on these criteria and produce contributions with higher quality.¹² We broadly categorize incentives into two types: intrinsic incentives, such as personal enthusiasm or altruism, and extrinsic incentives, such as monetary reward. Intrinsic incentives in conjunction with extrinsic ones can motivate honest users to participate in the task. Moreover, in some cases, the intrinsic incentives' positive effect on the outcome's quality is more significant than the impact of the extrinsic incentives.¹⁶

Looking at monetary rewards, which are common, the reward amount attracts more workers and affects how fast they accomplish the task, but increasing the amount doesn't necessarily increase outcome quality.¹⁶ Some research also shows that the payment method might have a bigger impact on outcome quality than the payment amount itself.^{6,16} For example, in a requested task that requires finding 10 words in a puzzle, paying per puzzle will lead to more solved puzzles than paying per word.¹³

Quality-Control Approaches

Researchers and practitioners have proposed several quality-control approaches that fall under the aforementioned quality dimensions and factors. We broadly classify existing approaches into two categories: design-time (see Table 1) and run-time (see Table 2). These two categories aren't mutually exclusive. A task can employ both approaches to maximize the possibility of receiving high-quality outcomes.

At design time, the requesters can leverage techniques for preparing a

Table 1. Existing quality-control design-time approaches.

Quality-control approach	Subcategories	Description	Sample application
Effective task preparation	Defensive design	Provides an unambiguous description of the task; task design is defensive — that is, cheating isn't easier than doing the task; defines evaluation and compensation criteria	References 1,3,6,12
Worker selection	Open to all	Allows everybody to contribute to the task	ESP Game, Thredless.com
	Reputation-based	Lets only workers with prespecified reputation levels contribute to the task	MTurk, Stack Overflow, 4
	Credential-based	Allows only workers with prespecified credentials to do the task	Wikipedia, Stack Overflow, 4

Table 2. Existing quality-control runtime approaches.

Quality-control approach	Description	Sample application
Expert review	Domain experts check contribution quality.	Academic conferences and journals, Wikipedia, 3
Output agreement	If workers independently and simultaneously provide the same description for an input, they are deemed correct.	ESP Game
Input agreement	Independent workers receive an input and describe it to each other. If they all decided that it's a same input, it's accepted as a quality answer.	Tag-A-Tune
Ground truth	Compares answers with a gold standard, such as known answers or common sense facts to check the quality.	CrowdFlower, MTurk
Majority consensus	The judgment of a majority of reviewers on the contribution's quality is accepted as its real quality.	TurKit, Threadless.com, MTurk
Contributor evaluation	Assesses a contribution based on the contributor's quality.	Wikipedia, Stack Overflow, MTurk
Real-time support	Provides shepherding and support to workers in real time to help them increase contribution quality.	Reference 12
Workflow management	Designs a suitable workflow for a complex task; workflow is monitored to control quality, cost, and so on, on the fly.	References 13,14

well-designed task and just allow a suitable crowd to contribute to the task. Although these techniques increase the possibility of receiving high-quality contributions from the crowd, there is still a need to control the quality of contributions at runtime. Even high-quality workers might submit low-quality contributions because of mistakes or misunderstanding. Therefore, requesters must still put in place runtime quality-control approaches when the task is running as well as when the crowd contributions are being collected and probably aggregated to build the final task answer. We discuss both

design-time and runtime approaches in more detail in the Web appendix at <http://doi.ieeecomputersociety.org/10.1109/MIC.2013.20>.

Although researchers have proposed and used several quality-control approaches so far, many open issues and challenges remain for defining, measuring, and managing quality in crowdsourcing systems, and these issues require further research and investigation.

One serious limitation of existing approaches is their reliance on primitives and hard-wired

quality-control techniques. These approaches are typically embedded in their host systems, and requesters can't customize them based on their specific requirements. Defining new approaches is another challenge that requesters struggle with. Although some tools — such as TurKit — that rely on current crowdsourcing systems let users define some quality-control processes, using these tools requires programming skills such as Java or C++.

Endowing crowdsourcing services with customizable, rich, and robust quality-control techniques is key to crowdsourcing platforms'


wide ranging success – whether it's supporting micro and commodity tasks or high-value processes (such as business processes or intelligence data gathering). Requesters can achieve this functionality using a generic quality-control framework that lets them define new quality-control approaches and reuse or customize existing ones. Such a framework should also be capable of being seamlessly integrated with existing crowdsourcing platforms to let requesters benefit from both crowdsourcing and quality-control systems simultaneously. Building such a framework can be an interesting future direction for research in the crowdsourcing arena.

Another major limitation of existing quality-control approaches comes from the subjective nature of quality, particularly in crowdsourcing systems. The quality of a task's outcome might depend on several parameters, such as requesters' requirements, task properties, crowd interests and incentives, and costs. Currently, quality-control techniques are domain-specific – that is, a technique that performs well for some tasks might perform poorly on new and different ones. For instance, approaches that are suitable for checking a written essay's quality are different from those to control quality in an image-processing task. Finding a suitable approach based on a particular situation is a challenge that needs more investigation.

One solution to this limitation is a recommender system, which gives requesters a list of adequate quality-control approaches. Such a recommender could use machine learning techniques to provide more precise recommendations. It should offer the requester a list of approaches that best suit the situation based on the requester's profile (social relations, history, interests, and so on), the task's type and attributes,

the history of the existing crowd, and the quality requirements of the task, along with many more options. Designing such a system can be a suitable direction for further study.

Thanks to Web 2.0 technologies and the rapid growth of mobile computing in the form of smartphones, tablets, and so on, a tremendous amount of human computation power is available for accomplishing jobs almost for free. On the other hand, artificial intelligence and machine learning are fast-growing areas in computer science. We envision that, in the near future, combining the crowd and machines to solve problems will be easily feasible.¹⁷ This will raise some interesting research challenges. Topics such as machine versus human trustworthiness, workflow design for such tasks, and conflict resolution between human and machine judgments will all need to be addressed.

Moreover, people are at the core of crowdsourcing systems. However, they're also distributed among separated online communities, and a requester can't easily employ crowds from several communities. We foresee that this will be simplified in the near future via service composition middleware. Building such middleware will require addressing several issues, including how to share people-quality indicators such as reputation and expertise between different communities and how to build a public global picture for each individual based on his or her available history of activities in different possible crowd communities. Addressing these issues will be another interesting research direction for crowdsourcing systems. 

References

1. A. Kittur, E. Chi, and B. Suh, "Crowdsourcing User Studies with Mechanical Turk," *Proc. 26th Ann. SIGCHI Conf.*

Human Factors in Computing Systems, ACM, 2008, pp. 453–456.

2. R. Khazankin, S. Daniel, and S. Dustdar, "Predicting QoS in Scheduled Crowdsourcing," *Advanced Information Systems Eng.*, vol. 7328, J. Ralyté et al., eds., Springer, 2012, pp. 460–472.
3. A.J. Quinn and B.B. Bederson, "Human Computation: A Survey and Taxonomy of a Growing Field," *Proc. 2011 Ann. Conf. Human Factors in Computing Systems*, ACM, 2011, pp. 1403–1412.
4. D. Schall, F. Skopik, and S. Dustdar, "Expert Discovery and Interactions in Mixed Service-Oriented Systems," *IEEE Trans. Services Computing*, vol. 5, no. 2, 2012, pp. 233–245.
5. G. Wang et al., "Serf and Turf: Crowdturfing for Fun and Profit," *Proc. 21st Int'l Conf. World Wide Web*, ACM, 2012, pp. 679–688.
6. J.J. Chen, N. Menezes, and A. Bradley, "Opportunities for Crowdsourcing Research on Amazon Mechanical Turk," *Proc. CHI 2011 Workshop Crowdsourcing and Human Computation*, 2011; <http://crowdresearch.org/chi2011-workshop/papers/chen-jenny.pdf>.
7. O. Scekic, H. Truong, and S. Dustdar, "Modeling Rewards and Incentive Mechanisms for Social BPM," *Business Process Management*, vol. 7481, A. Barros et al., eds., Springer, 2012, pp. 150–155.
8. E. Agichtein et al., "Finding High-Quality Content in Social Media," *Proc. Int'l Conf. Web Search and Web Data Mining*, ACM, 2008, pp. 183–194.
9. P. Crosby, *Quality is Free*, McGraw-Hill, 1979.
10. A. Jøsang, R. Ismail, and C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision," *Decision Support Systems*, vol. 43, no. 2, 2007, pp. 618–644.
11. L.D. Alfaro et al., "Reputation Systems for Open Collaboration," *Comm. ACM*, vol. 54, no. 8, 2011, pp. 81–87.
12. S.P. Dow et al., "Shepherding the Crowd Yields Better Work," *Proc. 2012 ACM Conf. Computer Supported Cooperative Work (CSCW 12)*, ACM, 2012, pp. 1013–1022.

13. A. Kittur et al., "CrowdForge: Crowdsourcing Complex Work," *Proc. 24th Ann. ACM Symp. User Interface Software and Technology*, ACM, 2011, pp. 43–52.
14. A. Kulkarni, M. Can, and B. Hartmann, "Collaboratively Crowdsourcing Workflows with Turkomatic," *Proc. 2012 ACM Conf. Computer Supported Cooperative Work (CSCW 12)*, ACM, 2012, pp. 1003–1012.
15. G. Little et al., "TurKit: Human Computation Algorithms on Mechanical Turk," *Proc. 23rd Ann. ACM Symp. User Interface Software and Technology*, ACM, 2010, pp. 57–66.
16. W. Mason and D.J. Watts, "Financial Incentives and the 'Performance of Crowds,'" *SIGKDD Explorations Newsletter*, vol. 11, 2010, pp. 100–108.
17. H.L. Truong, S. Dustdar, and K. Bhat-tacharya, "Programming Hybrid Services in the Cloud," *Service-Oriented Computing*, vol. 7636, C. Liu et al., eds., Springer, 2012, pp. 96–110.

Mohammad Allahbakhsh is a PhD candidate in the School of Computer Science and Engineering at the University of New South Wales, Australia. His research focuses on quality control in crowdsourcing systems. Allahbakhsh has an MS in software engineering from Ferdowsi University of Mashhad. He's a student member of IEEE. Contact him at mallahbakhsh@cse.unsw.edu.au.

Boualem Benatallah is a professor of computer science at the University of New South Wales, Australia. His research interests include system and data integration, process modeling, and service-oriented architectures. Benatallah has a PhD in computer science from Grenoble University, France. He's a member of IEEE. Contact him at boualem@cse.unsw.edu.au.


Aleksandar Ignjatovic is a senior lecturer in the School of Computer Science and Engineering at the University of New South Wales, Australia. His current research interests include applications

of mathematical logic to computational complexity theory, sampling theory, and online communities. Ignjatovic has a PhD in mathematical logic from the University of California, Berkeley. Contact him at ignjat@cse.unsw.edu.au.

Hamid Reza Motahari-Nezhad is a research scientist at Hewlett-Packard Laboratories in Palo Alto, California. His research interests include business process management, social computing, and service-oriented computing. Motahari-Nezhad has a PhD in computer science and engineering from the University of New South Wales, Australia. He's a member of the IEEE Computer Society. Contact him at hamid.motahari@hp.com.

Elisa Bertino is a professor of computer science at Purdue University and serves as research director for the Center for Education and Research in Information Assurance and Security (CERIAS). Her main research interests include security, privacy, digital identity management systems, database systems, distributed systems, and multimedia systems. She's a fellow of IEEE and ACM and has been named a Golden Core Member for her service to the IEEE Computer Society. Contact her at bertino@cs.purdue.edu.

Schahram Dustdar is a full professor of computer science and head of the Distributed Systems Group, Institute of Information Systems, at the Vienna University of Technology. His research interests include service-oriented architectures and computing, cloud and elastic computing, complex and adaptive systems, and context-aware computing. Dustdar is an ACM Distinguished Scientist (2009) and IBM Faculty Award recipient (2012). Contact him at dustdar@dsg.tuwien.ac.at.

 Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

IEEE computer society

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field. Visit our website at www.computer.org.
OMBUDSMAN: Email help@computer.org.

Next Board Meeting: 13–14 June 2013, Seattle, WA, USA

EXECUTIVE COMMITTEE

President: David Alan Grier
President-Elect: Dejan S. Milojevic; **Past President:** John W. Walz; **VP, Standards Activities:** Charlene ("Chuck") J. Walrad; **Secretary:** David S. Ebert; **Treasurer:** Paul K. Joannou; **VP, Educational Activities:** Jean-Luc Gaudiot; **VP, Member & Geographic Activities:** Elizabeth L. Burd (2nd VP); **VP, Publications:** Tom M. Conte (1st VP); **VP, Professional Activities:** Donald F. Shafer; **VP, Technical & Conference Activities:** Paul R. Croll; **2013 IEEE Director & Delegate Division VIII:** Roger U. Fujii; **2013 IEEE Director & Delegate Division V:** James W. Moore; **2013 IEEE Director-Elect & Delegate Division V:** Susan K. (Kathy) Land

BOARD OF GOVERNORS

Term Expiring 2013: Pierre Bourque, Dennis J. Frailey, Atsuhito Goto, André Ivanov, Dejan S. Milojevic, Paolo Montuschi, Jane Chu Prey, Charlene ("Chuck") J. Walrad
Term Expiring 2014: Jose Ignacio Castillo Velazquez, David S. Ebert, Hakan Erdogmus, Gargi Keeni, Fabrizio Lombardi, Hironori Kasahara, Arnold N. Pears
Term Expiring 2015: Ann DeMarle, Cecilia Metra, Nita Patel, Diomidis Spinellis, Phillip Laplante, Jean-Luc Gaudiot, Stefano Zanero

EXECUTIVE STAFF

Executive Director: Angela R. Burgess; **Associate Executive Director & Director, Governance:** Anne Marie Kelly; **Director, Finance & Accounting:** John Miller; **Director, Information Technology & Services:** Ray Kahn; **Director, Membership Development:** Violet S. Doan; **Director, Products & Services:** Evan Butterfield; **Director, Sales & Marketing:** Chris Jensen

COMPUTER SOCIETY OFFICES

Washington, D.C.: 2001 L St., Ste. 700, Washington, D.C. 20036-4928
Phone: +1 202 371 0101 • **Fax:** +1 202 728 9614
Email: hq.ofc@computer.org
Los Alamitos: 10662 Los Vaqueros Circle, Los Alamitos, CA 90720 • **Phone:** +1 714 821 8380 • **Email:** help@computer.org
Membership & Publication Orders
Phone: +1 800 272 6657 • **Fax:** +1 714 821 4641 • **Email:** help@computer.org
Asia/Pacific: Watanabe Building, 1-4-2 Minami-Aoyama, Minato-ku, Tokyo 107-0062, Japan • **Phone:** +81 3 3408 3118 • **Fax:** +81 3 3408 3553 • **Email:** tokyo.ofc@computer.org

IEEE BOARD OF DIRECTORS

President: Peter W. Staecker; **President-Elect:** Roberto de Marca; **Past President:** Gordon W. Day; **Secretary:** Marko Delimar; **Treasurer:** John T. Barr; **Director & President, IEEE-USA:** Marc T. Apter; **Director & President, Standards Association:** Karen Bartleson; **Director & VP, Educational Activities:** Michael R. Lightner; **Director & VP, Membership and Geographic Activities:** Ralph M. Ford; **Director & VP, Publication Services and Products:** Gianluca Setti; **Director & VP, Technical Activities:** Robert E. Heber; **Director & Delegate Division V:** James W. Moore; **Director & Delegate Division VIII:** Roger U. Fujii

revised 22 Jan. 2013

