

Appendix – Event Heap Survey Material

By Brad Johanson

© Copyright 2003

This is a supplemental electronic appendix that is a companion to “Application Coordination Infrastructure for Ubiquitous Computing Rooms,” a dissertation by the author which was written in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering at Stanford University.

As discussed in Chapter 6 of the dissertation, two surveys were given about the Event Heap, and more generally about iROS, the meta operating system for which the Event Heap serves as the primary communication and coordination mechanism. The first survey was given to developers that have used the Event Heap, while the second one was given to people that have administered interactive workspaces using the iROS system (and by association the Event Heap itself). Both surveys were given using an on-line system intended primarily for course feedback surveys, but adapted to this purpose [1]. This appendix contains the survey material as it was presented to the participants along with the full results. It is presented without further comment as the survey is discussed in Chapter 6 of the dissertation.

Survey Directions

The survey was sent out to several mailing lists related to the interactive workspaces group at Stanford and also to the various mailing lists associated with the Open Source distribution of iROS hosted on the Source Forge site (<http://iros.sourceforge.net>). The email directions were as followed, and were the same for both developer and administrator surveys:

```
From: Brad Johanson (bjohanso@graphics.stanford.edu)
Subject: Event Heap Surveys (and FREE MOVIE TICKETS!)
```

Hello,

My dissertation is on the Event Heap, one of the components of the iROS infrastructure system. In order to go beyond anecdotal evidence of the success of the Event Heap as a coordination system for interactive workspace applications I've put together two user surveys, one for developers that have used the Event Heap, and another for people who have administered interactive workspace using iROS (and by extension the Event Heap).

So, if you fall into either of these two categories, I'd appreciate it if you would fill out the on-line survey in that particular category (directions attached). Even if your experience is limited, your feedback would be valuable! Each survey takes between 15 and 30 minutes to take. They must be filled out in one sitting. As added incentive, one in five people that fills in the survey will get a pair of movie tickets (even if

you live beyond the Bay Area I'll still try and work something out). [I know this isn't much incentive, but I hope you'll still do the survey in the interest of furthering science and helping me graduate!]

Finally, please complete the surveys by October 9th. I'll still look at surveys completed after that, but you won't be eligible for the raffle.

Please forward or mention the survey to anyone that may not have seen this email but does use the Event Heap either as a developer or an administrator.

Thanks for your time!

-Brad Johanson

=====
Survey Instructions
=====

To take the survey, go to the following URL to start:

<http://www.wcer.wisc.edu/salgains/student/default.asp>

It will prompt you for a course ID (I'm re-tasking a course evaluation system) and password which you will find in the section below for your survey. For your student ID you can just enter your email address (or to remain anonymous enter any other unique string). You can then fill in the on-line form to answer the questions. Note that some questions, marked 'EXPLANATORY: DON'T ANSWER' are only there to describe the next section and are questions due to limitations in the system I'm using to do the survey. Make sure to give me contact info in the last section if you want to be in the drawing for the movie tickets.

Part of the survey consists of ranking certain features of the Event Heap. While it can be done on-line, it will be easier if you print out the appropriate ranking sheet for the survey you will do (below with the survey info), work out your results by hand, and then enter them on-line.

Event Heap Developer Survey

~~~~~

The Course ID 714707641

The Course Password 'eheap'

Student ID Students enter their own (just enter your name or email account name== e.g. 'bjohanso')

Printable rankings sheet:

[http://graphics.stanford.edu/~bjohanso/eheap\\_survey/programmer\\_rankings.pdf](http://graphics.stanford.edu/~bjohanso/eheap_survey/programmer_rankings.pdf)

Event Heap Administrator Survey

~~~~~

The Course ID 714707642

The Course Password 'eheap'

Student ID Students enter their own (just enter your name or email account name== e.g. 'bjohanso')

Printable rankings sheet:

http://graphics.stanford.edu/~bjohanso/eheap_survey/administrator_rankings.pdf

Developer Survey

The first part of this appendix presents the material and results related to the developer survey.

Web Survey Questions

The questions that were presented on the web site to developers participating in the survey are shown on the succeeding pages. The actual pages shown were generated from a printout of the web site to a PDF file.

Student Assessment of Learning Gains Survey

Stanford University: Interactive Workspaces 1-1 1, 2002

1. [EXPLANATORY: DON'T ANSWER] The questions in this first section are related to your background.

- Yes
- No

2. In terms of my general knowledge, comfort, experience and ability as a general application developer, I would consider myself:

- No experience
- Novice
- Amateur
- Intermediate
- Master
- Expert

3. Comments

4. In terms of my general knowledge, comfort, experience and ability as an application developer for interactive workspaces, I would consider myself:

- No experience
- Novice
- Amateur
- Intermediate
- Master
- Expert

5. Comments

6. In terms of my general knowledge, comfort, experience and ability as an application developer for the Event Heap,

I would consider myself:

- No experience
- Novice
- Amateur
- Intermediate
- Master
- Expert

7. Comments**8. Please give a brief description of the Event Heap based applications you have done and the specific Event Heap functionality you have used:****9. [EXPLANATORY: DON'T ANSWER] The second section asks general questions about interactive workspaces not specific to developers or the Event Heap.**

- Yes
- No

10. 'Applications' for interactive workspaces can either be collections of small programs (ensembles) or monolithic applications which run across the whole space. Which of these two will be the more common case:

- Application Ensembles only
- Application Ensembles and an occasional monolithic application
- Both equally
- Mostly monolithic applications, but some ensembles
- Monolithic applications only
- Don't know.

11. Comments:

12. Interactive workspaces are likely to be dynamic on short time scales due to application and device failure and the entrance and departure of portable devices:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

13. Comments:

14. Which is more tolerable:

- Occasional total system failures which require ten or more minutes to restart
- More frequent faults which effect only a part of the system and require one or two minutes to reset that component
- Not sure

15. Comments:

16. Would you accept reduced performance (up to 1 s latency) from an interactive workspaces system infrastructure to get guaranteed message delivery?

- Yes
- No

17. Comments

18. Would you accept reduced performance (up to 1 s latency) from an interactive workspaces system infrastructure if the system guaranteed consistency (all components see the same events in the same order)?

- Yes
 No

19. Comments

20. Would you accept reduced performance (up to 1 s latency) from an interactive workspaces system infrastructure if it supported transactions among connected components?

- Yes
 No

21. Comments

22. [EXPLANATORY: DON'T ANSWER] Please rank the following six general features according to how important they are for an interactive workspace coordination infrastructure. Please assign a unique value for each feature, 1 being most important:

- Yes
 No

23. Support for heterogeneous devices and hardware platforms

- 1
- 2
- 3
- 4
- 5
- 6

24. Support for a variety of development platforms (Java, C++, .NET, etc.)

- 1
- 2
- 3
- 4
- 5
- 6

25. Ability to easily integrate legacy applications (COTS and In-house)

- 1
- 2
- 3
- 4
- 5
- 6

26. Provision for new hardware and software that may be released in the future.

- 1
- 2
- 3
- 4
- 5
- 6

27. Reliability and Stability

- 1
- 2
- 3
- 4
- 5
- 6

28. Security

- 1
- 2
- 3
- 4
- 5
- 6

29. Comments on the ranking of the features in the previous six questions:

30. [EXPLANATORY: DON'T ANSWER] Please rank the following ten developer features according to how important they are for an interactive workspace coordination infrastructure. Please assign a unique value for each feature, 1 being the most important:

- Yes
- No

31. Event expiration

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

32. Extensibility of application ensembles written with an infrastructure

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

33. Ability of infrastructure API to express different types of coordination

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

34. Simplicity and ease of learning the infrastructure API

- 1
- 2
- 3

- 4
- 5
- 6
- 7
- 8
- 9
- 10

35. Ease of debugging application ensembles written with the infrastructure

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

36. Portability of applications written with the infrastructure to new environments

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

37. Support for different communication patterns (point-to-point, broadcast, etc.)

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

38. Transparent communication (ability to monitor communications and easily interpret message contents)

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

39. Ordering of sent messages and at most once reception

- 1

- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

40. Support for transmission of complex object types

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

41. Comments on ranking developer features:**42. [EXPLANATORY: DON'T ANSWER] The following nine questions are general questions about developing for interactive workspaces:**

- Yes
- No

43. Do you foresee adapting your interactive workspaces applications to work with other applications in the future?

- Yes
- No, the application set will be written once and then left alone
- Not sure

44. Comments:

45. What kind of message routing fits best with the type of applications you usually write for interactive workspaces?

- Explicitly addressed messages sent to a specific named entity
- Content based routing where receiver applications choose which messages to accept based on their content

46. Comments:

47. What is more true in your application:

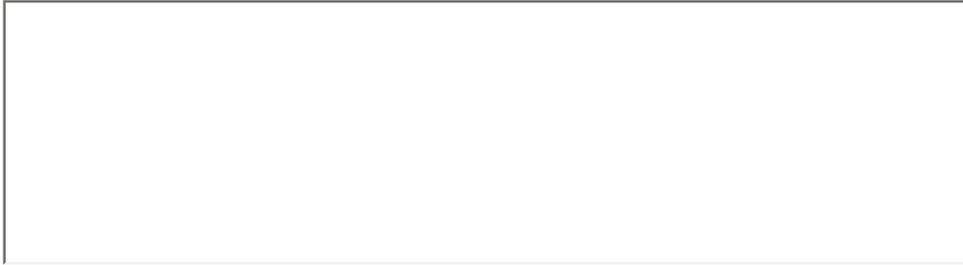
- The receiver of a message knows best whether or not it should receive the message
- The sender knows best who should receive its message

48. Comments:

49. How often do you use point-to-point routing in your applications:

- Very often
- Often
- Sometimes
- A Little
- Never

50. Comments:



51. How often do you use multicast routing (one sender to a group of receivers) in your applications:

- Very often
- Often
- Sometimes
- A Little
- Never

52. Comments:



53. How often do you use broadcast routing (one sender to all receivers in the environment) routing in your applications:

- Very often
- Often
- Sometimes
- A Little
- Never

54. Comments:



55. How often do you use anycast routing (one sender to exactly one of a group of receivers) in your applications:

- Very often
- Often
- Sometimes
- A Little
- Never

56. Comments:

57. Which type of delivery tends to work better for your applications?

- Pull based, where applications pull in messages as they need them
- Push based, where you register and the infrastructure gives you a callback each time a matching message is submitted by some sender

58. Comments:

59. Please list any distributed computing coordination techniques that you have used in the past: (e.g. Direct socket, RMI/RPC including Jini, publish-subscribe, etc.)

60. [EXPLANATORY: DON'T ANSWER] The following fourteen questions specifically relate to developing using the Event Heap:

- Yes

No

61. It was straightforward to adapt my previous programming techniques for use with the Event Heap:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

62. Comments:

63. The current Event Heap API is sufficient to express all interactive workspace coordination tasks I have needed to code in a reasonably straightforward manner:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

64. Comments:

65. Compared with other applications I have developed, Event Heap applications are very easy to debug:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

66. Comments:

67. The Event Heap is very stable and reliable:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

68. Comments:

69. It has been easier and quicker to develop stable applications and application sets using the Event Heap than it has been with previous systems I have used:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

70. Comments:

71. The current Event Heap object structure for events is sufficient for my application development needs:

- Strongly agree

- Agree
- Neutral
- Disagree
- Strongly disagree

72. Comments:**73. The extensible event format used by the Event Heap which allows extra fields to be added to events has been useful:**

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

74. Comments:**75. Downloading the iROS/Event Heap system and getting it working was easy:**

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

76. Comments:

77. The fact that the Event Heap is a centralized message exchange system running on a single server machine is an advantage:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

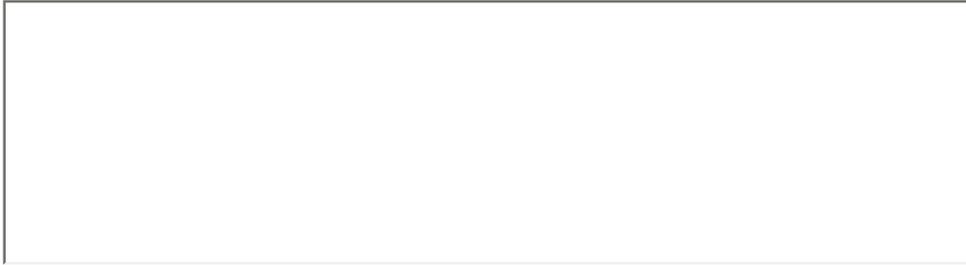
78. Comments:

79. How is the performance (bandwidth, throughput, latency) of the Event Heap for the applications you use?

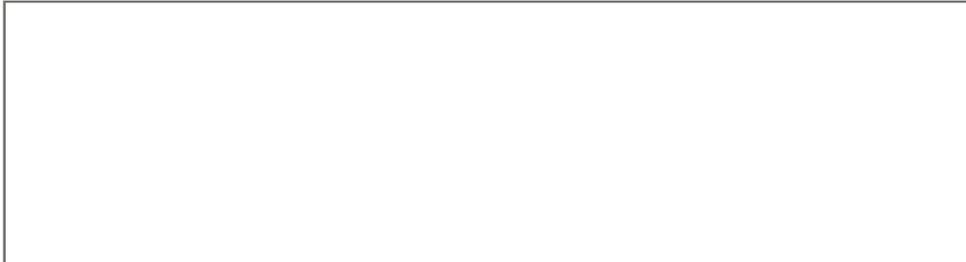
- Unusable
- Poor
- Tolerable
- Acceptable
- Better than expected
- Superior

80. Comments:

81. Are there any specific areas of Event Heap performance that you would like improved?



82. What are the typical event expiration times you use for your applications?



83. What platforms are important for the Event Heap to support (current and desired)? (e.g. C++, .NET, Perl, HTTP, Visual Basic, Cocoa, etc.)



84. What other systems did you consider using besides iROS/the Event Heap, if any? What factors made you decide in favor of the Event Heap/iROS (e.g. easy to download)?



85. [EXPLANATORY: DON'T ANSWER] Concluding Section

- Yes
 No

86. Any overall suggestions and/or comments?**87. Your name and contact information (optional)**

[Click here to submit your responses](#)

This site was created with funding courtesy of:

- [The Exxon Education Foundation](#)
- The NSF Chemistry Coalitions:
 - [New Traditions \(NT\)](#)
 - [ChemLinks](#)
 - [ModularChemistry \(MC2\)](#)
- [The NSF National Institute for Science Education](#)
- [The NSF SENCER Institutes](#)

Original Content Copyright ©1997 Elaine Seymour. All rights reserved. [Your comments](#) are welcome.

Ranking Assistance Sheet

The web based survey system did not provide a user friendly mechanism for ranking items relative to one another since only multiple choice and open ended questions were allowed. For the two questions that asked participants to rank features, the features were presented one by one in the survey with radio buttons to select their numeric ranking. Since this made for a somewhat awkward procedure, the sheet on the following page was provided for ranking by hand, and the participant could then copy their selections into the web based survey form.

Importance Ranking of Interactive Workspace Coordination Infrastructure Characteristics

Developer Survey

Please rank the features in each section in order of importance, with '1' being most important, '2' next most important and so on. This form is to simplify the ranking process only. Please enter your final choices into the on-line form at <http://www.wcer.wisc.edu/salgains/student/default.asp>. Please use course ID 714707641 and password 'krowi'. Use your email address as your identifier.

General Characteristics

Survey Question #	Platform Feature/Characteristic	Ranking
23	Support for heterogeneous devices and hardware platforms	
24	Support for a variety of development platforms (Java, C++, .NET, etc.)	
25	Ability to easily integrate legacy applications (COTS and In-house)	
26	Provision for new hardware and software that may be released in the future.	
27	Reliability and Stability	
28	Security	

Developer Specific Characteristics

Survey Question #	Platform Feature/Characteristic	Ranking
31	Event expiration	
32	Extensibility of application ensembles written with an infrastructure	
33	Ability of infrastructure API to express different types of coordination	
34	Simplicity and ease of learning the infrastructure API	
35	Ease of debugging application ensembles written with the infrastructure	
36	Portability of applications written with the infrastructure to new environments	
37	Support for different communication patterns (point-to-point, broadcast, etc.)	
38	Transparent communication (ability to monitor communications and easily interpret message contents)	
39	Ordering of sent messages and once only reception	
40	Support for transmission of complex object types	

Raw Results

The raw results, per respondent, are listed over the next pages. Names of participants have been removed to preserve anonymity. All other information is presented exactly as it was submitted on the survey web page. Questions are not repeated in the tables as they may be found earlier in this appendix in the ‘Web Survey Questions’ developer’s section. For legibility, the data is formatted in four tables, each containing three respondents.

	Respondent 1	Respondent 2	Respondent 3
Q1:	No Response	No Response	No Response
Q2:	Master	Master	Master
Q3:	NO ANSWER	NO ANSWER	I prefer proficient, master doesn't sound very modest :)
Q4:	Intermediate	Master	Intermediate
Q5:	NO ANSWER	NO ANSWER	NO ANSWER
Q6:	Intermediate	Intermediate	Amateur
Q7:	NO ANSWER	NO ANSWER	NO ANSWER
Q8:	The iButtons (first version). Simple event passing. I created my own event types and listeners for them.	-Time controller to broadcast time messages to applications - event heap communication add-ons to MS apps and proprietary 4D app	use eventheap to start a videoconferencing program on another computer in the iRoom.
Q9:	No Response	No Response	No Response
Q10:	Application Ensembles and an occasional monolithic application	Application Ensembles and an occasional monolithic application	Application Ensembles and an occasional monolithic application
Q11:	Not sure what is meant by "the whole space."	NO ANSWER	NO ANSWER
Q12:	Agree	Strongly agree	Agree
Q13:	NO ANSWER	NO ANSWER	NO ANSWER
Q14:	Occasional total system failures which require ten or more minutes to restart	More frequent faults which effect only a part of the system and require one or two minutes to reset that component	More frequent faults which effect only a part of the system and require one or two minutes to reset that component
Q15:	It is preferable to have a single procedure for restarting anything, for simplicity.	NO ANSWER	NO ANSWER
Q16:	No	No	Yes
Q17:	NO ANSWER	NO ANSWER	but not for events like mouse movements
Q18:	No	No	No
Q19:	NO ANSWER	NO ANSWER	but yes for database like apps (meeting capture, etc)
Q20:	No	No	No
Q21:	NO ANSWER	NO ANSWER	I don't know what "transaction among connected components" mean
Q22:	No Response	No Response	No Response
Q23:	3	6	1
Q24:	4	4	6
Q25:	2	3	2
Q26:	5	1	3
Q27:	1	2	4
Q28:	6	5	5
Q29:	NO ANSWER	NO ANSWER	if I am a user, I don't care how you wrote the code, so development platform ranks last.

Continued on next page

Continued from previous page

	Respondent 1	Respondent 2	Respondent 3
Q30:	No Response	No Response	No Response
Q31:	NO ANSWER	5	4
Q32:	NO ANSWER	1	7
Q33:	NO ANSWER	6	1
Q34:	1	2	10
Q35:	NO ANSWER	4	9
Q36:	NO ANSWER	3	8
Q37:	NO ANSWER	8	2
Q38:	NO ANSWER	10	3
Q39:	NO ANSWER	7	6
Q40:	NO ANSWER	9	8
Q41:	NO ANSWER	NO ANSWER	i ranked it based on the capability of the system, since an easy to learn system that doesn't do what you need to do is not useful.
Q42:	No Response	No Response	No Response
Q43:	NO ANSWER	Yes	Yes
Q44:	NO ANSWER	NO ANSWER	NO ANSWER
Q45:	NO ANSWER	Content based routing where receiver applications choose which messages to accept based on their content	Explicitly addressed messages sent to a specific named entity
Q46:	NO ANSWER	NO ANSWER	NO ANSWER
Q47:	NO ANSWER	The receiver of a message knows best whether or not it should receive the message	The sender knows best who should receive its message
Q48:	NO ANSWER	NO ANSWER	NO ANSWER
Q49:	No Response	Sometimes	Very often
Q50:	NO ANSWER	NO ANSWER	NO ANSWER
Q51:	No Response	Very often	Never
Q52:	NO ANSWER	NO ANSWER	NO ANSWER
Q53:	No Response	Very often	Never
Q54:	NO ANSWER	NO ANSWER	NO ANSWER
Q55:	No Response	Sometimes	Never
Q56:	NO ANSWER	NO ANSWER	NO ANSWER
Q57:	NO ANSWER	Push based, where you register and the infrastructure gives you a callback each time a matching message is submitted by some sender	Push based, where you register and the infrastructure gives you a callback each time a matching message is submitted by some sender
Q58:	NO ANSWER	NO ANSWER	NO ANSWER
Q59:	NO ANSWER	NO ANSWER	direct socket, shared memory
Q60:	No Response	No Response	No Response
Q61:	No Response	Strongly agree	Strongly agree
Q62:	NO ANSWER	NO ANSWER	NO ANSWER

Continued on next page

Continued from previous page

	Respondent 1	Respondent 2	Respondent 3
Q63:	No Response	Agree	Strongly agree
Q64:	NO ANSWER	NO ANSWER	NO ANSWER
Q65:	No Response	Agree	Agree
Q66:	NO ANSWER	NO ANSWER	NO ANSWER
Q67:	No Response	Agree	Neutral
Q68:	NO ANSWER	NO ANSWER	NO ANSWER
Q69:	No Response	Agree	Neutral
Q70:	NO ANSWER	NO ANSWER	NO ANSWER
Q71:	No Response	Agree	Agree
Q72:	NO ANSWER	NO ANSWER	NO ANSWER
Q73:	No Response	Strongly agree	Neutral
Q74:	NO ANSWER	NO ANSWER	NO ANSWER
Q75:	No Response	Neutral	Neutral
Q76:	NO ANSWER	NO ANSWER	I didn't like the java baggage :)
Q77:	No Response	Strongly agree	Strongly agree
Q78:	NO ANSWER	NO ANSWER	NO ANSWER
Q79:	NO ANSWER	Better than expected	Acceptable
Q80:	NO ANSWER	NO ANSWER	NO ANSWER
Q81:	NO ANSWER	NO ANSWER	NO ANSWER
Q82:	NO ANSWER	NO ANSWER	immediate
Q83:	NO ANSWER	NO ANSWER	c++, http
Q84:	NO ANSWER	NO ANSWER	NO ANSWER
Q85:	No Response	No Response	No Response
Q86:	sorry, i ran out of time to finish	NO ANSWER	NO ANSWER

Table 1 – Developer Survey Responses, Respondents One through Three

	Respondent 4	Respondent 5	Respondent 6
Q1:	No Response	No Response	No Response
Q2:	Intermediate	Intermediate	Master
Q3:	Was a master years ago, but am rusty.	In some areas (Mac OS X, Java) I might be a master, but in others (Windows) I'm an amateur.	NO ANSWER
Q4:	Intermediate	Intermediate	Intermediate
Q5:	Mostly involved at higher levels, but have done a lot of it.	I have been developing applications in the iRoom for about one year.	NO ANSWER
Q6:	Intermediate	Master	Intermediate
Q7:	Again, I've written little code myself, but have been involved in design of several application suites	see below	NO ANSWER
Q8:	Primarily example pieces, worked with actual implementers on the design of eHeap interfaces Both CIFE, and the iStuff and iWall HCI research work	I have written or helped write many applications the use the EventHeap (iPong, GroupBoard, iWall, PatchPanel) which, combined, use most of the functionality of the EventHeap (sending, receiving, logging, intercepting, etc.). I have also implemented a partial (only for sending) EventHeap API in C++, so I am experienced with the EventHeap Wire Protocol.	WorkspaceNavigator. We use the Event Heap to trigger capture at regular intervals on all the devices in the iroom. Also, iButtons send events which our application picks up.
Q9:	No Response	No Response	No Response
Q10:	Application Ensembles and an occasional monolithic application	Application Ensembles and an occasional monolithic application	Application Ensembles and an occasional monolithic application
Q11:	NO ANSWER	This is generally what I've seen in the iRoom. I don't want to rule out either, but the "UNIX model" of many small apps that work together seems to mesh well with dynamic, distributed environments.	NO ANSWER
Q12:	Agree	Agree	Strongly agree
Q13:	NO ANSWER	Generally true, but varies widely between different workspaces. It is possible to have a completely static workspace (in terms of entrance and departure of devices) that is also relatively stable (in terms of device failure).	NO ANSWER

Continued on next page

Continued from previous page

	Respondent 4	Respondent 5	Respondent 6
Q14:	More frequent faults which effect only a part of the system and require one or two minutes to reset that component	More frequent faults which effect only a part of the system and require one or two minutes to reset that component	Occasional total system failures which require ten or more minutes to restart
Q15:	NO ANSWER	Users learn to deal with "routine" problems, but have a hard time dealing with total catastrophic failures.	NO ANSWER
Q16:	No	Yes	No
Q17:	Not suitable for UI work	Guaranteed message delivery is important, but I wouldn't be too tolerant of extra latency (beyond TCP) caused by the interactive workspace infrastructure, since TCP is pretty fast and it already has some guarantees, which are often enough.	NO ANSWER
Q18:	No	Yes	No
Q19:	Similar. We're doing UI research, and need to reduce latency	Yes, but I'd like a way to give up consistency in exchange for less latency. Some applications really need low latency and don't care about consistency.	NO ANSWER
Q20:	No	No	No
Q21:	Would have to be an option, not all eHeap communication	I don't think I understand the question - what does an interactive workspaces infrastructure do besides support transactions among connected components? If you mean direct component-to-component transactions, I would expect it to be as fast as direct sockets.	NO ANSWER
Q22:	No Response	No Response	No Response
Q23:	2	2	2
Q24:	1	3	3
Q25:	3	6	6
Q26:	4	4	4
Q27:	5	1	1
Q28:	6	5	5

Continued on next page

Continued from previous page

	Respondent 4	Respondent 5	Respondent 6
Q29:	A stack ranking is hard. All of the flexibility components are important Reliability is important, but less expected of research software Security does matter, but not so much in HCI research	Support for a wide variety of hardware (both new and old) and development environments (my ranks 2,3,and 4) are all pretty much tied. Reliability and Stability is a #1 priority for anything that so many other applications will rely on. Security, though I ranked it as #5, can be very important in certain situations, perhaps even as high as #2 (it's not important in research, but in industry is can be required).	NO ANSWER
Q30:	No Response	No Response	No Response
Q31:	6	9	8
Q32:	5	3	1
Q33:	4	1	3
Q34:	9	6	9
Q35:	8	7	2
Q36:	7	4	7
Q37:	3	2	4
Q38:	2	5	5
Q39:	1	8	6
Q40:	10	10	10
Q41:	This was very hard to do, both physically (it's hard to keep track of the ordering and the questions when you have to scroll) and because some features simply have to be there, so it's not a stack ranking	The most important thing is for the API to support what needs to be done, by making it not only possible but easy to accomplish the required coordination. Specific features like event expiration, sequencing, and complex data types are important, but only if the theory of operation of the infrastructure requires them (Publish-Subscribe wouldn't have a notion of event expiration, for example) The ability to log and/or intercept communication is very important, though not as important as building the communicating pieces in the first place.	NO ANSWER
Q42:	No Response	No Response	No Response

Continued on next page

Continued from previous page

	Respondent 4	Respondent 5	Respondent 6
Q43:	Yes	Yes	Not sure
Q44:	NO ANSWER	Almost every app I wrote for the iRoom had to be adapted later for some other purpose, whether I planned it or not. It became very important to think about extensibility before writing new apps, so it would be easier to adapt when (not if) necessary.	NO ANSWER
Q45:	Content based routing where receiver applications choose which messages to accept based on their content	Content based routing where receiver applications choose which messages to accept based on their content	Content based routing where receiver applications choose which messages to accept based on their content
Q46:	Where "content" is event type. We need many-to-many, for an unbound number of many But, the content matching we use is relatively simple--name and id primarily	Though the routing I did was usually content based, it was typically based solely on the EventType (and sometimes on other field).	NO ANSWER
Q47:	The receiver of a message knows best whether or not it should receive the message	The receiver of a message knows best whether or not it should receive the message	The receiver of a message knows best whether or not it should receive the message
Q48:	The goal, actually, is to decouple the sender and receiver as much as possible. This way, we support generic input and generic receivers	see above (most routing I did was content based)	NO ANSWER
Q49:	A little	A little	Sometimes
Q50:	Basically PointRight	There is occasionally need for point-to-point, but it usually is required for interaction with an existing protocol, not because point-to-point is the best pattern.	NO ANSWER
Q51:	Often	Very often	Never
Q52:	NO ANSWER	This is perhaps the most important of often used feature of the EventHeap for the apps I have written.	NO ANSWER
Q53:	Very often	Never	Very often
Q54:	NO ANSWER	Sometimes I use very broad multicast, but never true broadcast.	NO ANSWER
Q55:	Sometimes	Sometimes	Never
Q56:	Not completely sure about this one	When this is needed, I often use a secondary field in the event to indicate which receiver is meant.	NO ANSWER

Continued on next page

Continued from previous page

	Respondent 4	Respondent 5	Respondent 6
Q57:	Push based, where you register and the infrastructure gives you a callback each time a matching message is submitted by some sender	Push based, where you register and the infrastructure gives you a callback each time a matching message is submitted by some sender	Pull based, where applications pull in messages as they need them
Q58:	NO ANSWER	Using registration and callbacks has the advantage of not blocking the listener. Quite often, an application wants to listen for certain kinds of messages for the lifetime of the applications, so registration is easiest.	NO ANSWER
Q59:	NO ANSWER	direct socket publish-subscribe distributed objects (Obj-C/Cocoa version of RMI/RPC)	NO ANSWER
Q60:	No Response	No Response	No Response
Q61:	Agree	Disagree	Agree
Q62:	Main disagreement is with error handling. The current API requires a very messy nesting to catch errors that really are best left uncaught. In general, applications should only catch errors that they can work around. Errors that mean that the system is corrupted or unstable are usually not worth catching, as there is nothing the application can do about it. Crashing is as good a response as anything, especially for research	Programming for the EventHeap involves thinking about problems that don't occur in other environments, but this has more to do with the applications being written than the EventHeap API. For example, the possibility of multiple users interacting with an application at the same time is an issue with EventHeap applications, because it is possible, while it generally isn't with a normal application.	NO ANSWER
Q63:	Strongly agree	Disagree	Agree
Q64:	NO ANSWER	While the EventHeap is good for many coordination tasks, there are some tasks it does not handle. Most notable are bulk data transfer and persistent state. There are other iROS APIs to handle some of these tasks, and how well they handle them is not what this survey is about - however, the point is that the EventHeap cannot stand alone.	NO ANSWER

Continued on next page

Continued from previous page

	Respondent 4	Respondent 5	Respondent 6
Q65:	Agree	Neutral	Agree
Q66:	Not much experience myself	EventHeap apps are not significantly easier to debug than other distributed and/or multithreaded apps. The ability to see Events on the EventHeap helps some, but there are often similar methods to debug non-EventHeap apps.	Though at first I didn't know about the application where you could see all the events going through.
Q67:	Neutral	Agree	Disagree
Q68:	In Java, yes. With the C++ API, no.	I have hardly ever seen the EventHeap server misbehave, and then only when I was actively trying to push it to its limits. The EventHeap client API does have some quirks (it has been known to hog the CPU when it can't find the server) but it generally runs very smoothly.	NO ANSWER
Q69:	Disagree	Strongly agree	Agree
Q70:	It's not the stability, it's the flexibility I admire	No system is the perfect development environment, but I would be hard pressed to find a better one. There are certainly many worse ones.	The system we developed would have been a nightmare without the Event Heap. It's not too stable however.
Q71:	Strongly agree	Agree	Strongly agree
Q72:	NO ANSWER	It is sufficient, but it would be convenient to be able to store more complex, hierarchical data types (like lists or hashes). Currently, these are usually flattened into a string representation in some arbitrary way. Having a standard way to do this might be beneficial to developers.	NO ANSWER
Q73:	Strongly agree	Strongly agree	Strongly agree
Q74:	NO ANSWER	All but the most simple messages I have created use extra fields. In fact, I hardly use the default fields at all, except for EventType and TimeToLive.	I used this extensively to add extra information that receivers need.
Q75:	Agree	Neutral	Agree
Q76:	NO ANSWER	I own a Mac, so I have not had much experience with the iROS install process. From a developer's standpoint, getting started with developing was a snap, since it only required the iROS.jar file.	NO ANSWER

Continued on next page

Continued from previous page

	Respondent 4	Respondent 5	Respondent 6
Q77:	Strongly agree	Agree	Agree
Q78:	For research systems, centralized is fine, and easier to debug and maintain.	Having all messages go through a single point has many advantages (most notably logging), but there are also disadvantages. Perhaps the option of a direct point-to-point could be added in such a way that logging is still possible (but sending messages both the other client and to the server, for example).	NO ANSWER
Q79:	Better than expected	Better than expected	Better than expected
Q80:	NO ANSWER	Performance has almost never been a problem. Only PointRight-over-EventHeap has raised concerns, and even then the EventHeap performed very well. Still, there is always room for performance improvements - the EventHeap will never be as efficient as direct sockets, but I believe there are still areas that can be squeezed.	NO ANSWER
Q81:	An even lower latency path would be useful, but this isn't just the eHeap	I think the primary performance improvements could be made with the size of the Event packets. Perhaps an option to not include many of the default fields?	NO ANSWER
Q82:	2 sec	From a few hundred milliseconds to several seconds (depending on the application, of course).	30 seconds
Q83:	HTTP, Java, C++, VB, Cocoa	Having a pure C reference implementation would make is possible to port the EventHeap API to many different platforms, since nearly all have C bindings. Almost every platform is potentially important, since each application will have its own unique platform and language requirements.	All my stuff is in java.
Q84:	It was there	None - I was developing specifically for the EventHeap/iRoom.	none

Continued on next page

Continued from previous page

	Respondent 4	Respondent 5	Respondent 6
Q85:	No Response	No Response	No Response
Q86:	The iROS has created an excellent infrastructure for HCI research in interactive environments. Most reliability problems seem to be with the C++ interface, or other parts of the iROS than the eHeap proper.	Better documentation. The javadocs are helpful, but a full-scale EventHeap tutorial explaining everything, including some of the more confusing quirks (ACTUAL/FORMAL/VIRTUAL, post vs. template type, etc.) would help many beginning developers. The currently posted example code is not enough for anything beyond the basics, and, besides, the link to the "sender" is broken.	a good tutorial for people just starting up.

Table 2 - Developer Survey Responses, Respondents Four through Six

	Respondent 7	Respondent 8	Respondent 9
Q1:	No Response	No Response	No Response
Q2:	Master	Intermediate	Expert
Q3:	NO ANSWER	NO ANSWER	NO ANSWER
Q4:	Amateur	Master	Expert
Q5:	NO ANSWER	NO ANSWER	NO ANSWER
Q6:	Intermediate	Expert	Expert
Q7:	NO ANSWER	Relatively speaking :)	NO ANSWER
Q8:	I have used the event heap to incorporate new physical user interfaces into the room. I use the event heap as a communications channel from the device proxy application to the device client application. I also have implemented a patch panel intermediary application that translates device events to application events so that users can specify the focus of a particular physical device. The event heap provides interesting qualities that allow for a software engineering technique that allows this decoupling and also provides for a higher grain of reuse.	I have written various point-point apps as well as multicast/broadcast type apps.	dataheap, smartpresenter, that sort of thing.
Q9:	No Response	No Response	No Response
Q10:	Application Ensembles and an occasional monolithic application	Application Ensembles and an occasional monolithic application	Application Ensembles and an occasional monolithic application
Q11:	NO ANSWER	NO ANSWER	NO ANSWER
Q12:	Agree	Agree	Agree
Q13:	NO ANSWER	NO ANSWER	I think they will eventually become dynamic on short time scales (where short is on the order of an hour or so). I think that's the right time scale because that's how long meetings tend to last. Once everyone is in the room, they tend to stay for awhile, and the system is pretty stable for the duration of the meeting. Device failures, etc occur, but I don't think this will occur often enough for change to occur on the order of minutes.

Continued on next page

Continued from previous page

	Respondent 7	Respondent 8	Respondent 9
Q14:	More frequent faults which effect only a part of the system and require one or two minutes to reset that component	More frequent faults which effect only a part of the system and require one or two minutes to reset that component	More frequent faults which effect only a part of the system and require one or two minutes to reset that component
Q15:	Even if it is the part of the system that you are working on 1 minute is a lot better than 10. But the degree of frequency can sway this answer the other direction.	The answer to 14 depends on how frequent "more frequent" is.	NO ANSWER
Q16:	Yes	Yes	No
Q17:	I would accept the reduced performance for additional features if I could choose the faster message delivery mechanism in a performance sensitive application. In other words, I wouldn't accept it if performance would be universally degraded.	16 is hard to answer. I answered yes because it appears that there would be cases where 1s extra latency is worth the guarantee. However, it would be nice if the penalty is only imposed on the messages where guaranteed delivery is needed.	NO ANSWER
Q18:	Yes	Yes	No
Q19:	same comment as above	Same as 17.	.
Q20:	Yes	No	No
Q21:	same comment as above	Transactions appear too heavyweight.	It might be acceptable for components which require transactional capability. If this latency is paid for by components that don't need those guarantees, that is unacceptable.
Q22:	No Response	No Response	No Response
Q23:	1	1	6
Q24:	3	4	5
Q25:	4	5	1
Q26:	5	6	4
Q27:	2	3	2
Q28:	6	2	3
Q29:	NO ANSWER	23 and 24 seem related, but my ranking is based on the following difference: I care about some language being supported for each platform, and not every language for each platform. it 26 is vague. Once the system goes beyond the "demo mode" and needs to be deployed, security and reliability become very important, although this is often overlooked by researchers.	We've been able to do a lot of interesting things with little support for heterogeneous devices/platforms. (we basically only run windows in the room, right?). The most important thing seems to have been integrating legacy software, because that's how we got most of our demo-functionality, right?

Continued on next page

Continued from previous page

	Respondent 7	Respondent 8	Respondent 9
Q30:	No Response	No Response	No Response
Q31:	10	9	2
Q32:	2	3	7
Q33:	8	1	9
Q34:	1	4	5
Q35:	7	5	8
Q36:	3	6	10
Q37:	4	2	1
Q38:	5	8	4
Q39:	9	7	3
Q40:	6	10	6
Q41:	NO ANSWER	Although I tried my best to order, I still feel there is some randomness here. Maybe this is not the right question to ask?	NO ANSWER
Q42:	No Response	No Response	No Response
Q43:	No, the application set will be written once and then left alone	Yes	Not sure
Q44:	I think that the infrastructure enables this to a higher degree with intermediation. Applications can send and receive generic application events that can be translated to different generic application events	NO ANSWER	NO ANSWER
Q45:	Explicitly addressed messages sent to a specific named entity	Explicitly addressed messages sent to a specific named entity	Explicitly addressed messages sent to a specific named entity
Q46:	NO ANSWER	NO ANSWER	NO ANSWER
Q47:	The receiver of a message knows best whether or not it should receive the message	The receiver of a message knows best whether or not it should receive the message	The sender knows best who should receive its message
Q48:	NO ANSWER	NO ANSWER	NO ANSWER
Q49:	Sometimes	Often	Very often
Q50:	NO ANSWER	NO ANSWER	NO ANSWER
Q51:	Sometimes	Sometimes	Very often
Q52:	NO ANSWER	NO ANSWER	i often use multicast or broadcast routing for a query, and unicast replies with the answer.
Q53:	Sometimes	Sometimes	Very often
Q54:	NO ANSWER	NO ANSWER	NO ANSWER
Q55:	Sometimes	Sometimes	Never
Q56:	NO ANSWER	NO ANSWER	NO ANSWER

Continued on next page

Continued from previous page

	Respondent 7	Respondent 8	Respondent 9
Q57:	Push based, where you register and the infrastructure gives you a callback each time a matching message is submitted by some sender	Push based, where you register and the infrastructure gives you a callback each time a matching message is submitted by some sender	Push based, where you register and the infrastructure gives you a callback each time a matching message is submitted by some sender
Q58:	NO ANSWER	NO ANSWER	NO ANSWER
Q59:	Direct socket	Direct socket, RMI/RPC, pub-sub	direct socket, rmi/rpc
Q60:	No Response	No Response	No Response
Q61:	Strongly agree	Agree	Strongly agree
Q62:	NO ANSWER	NO ANSWER	NO ANSWER
Q63:	Agree	Nuetral	Agree
Q64:	I wanted to implement RPC using events and I had to think about how I would do it. It is relatively simple, just not intuitive.	NO ANSWER	NO ANSWER
Q65:	Agree	Disagree	Nuetral
Q66:	NO ANSWER	I wouldn't say it is easier to debug with EventHeap.	NO ANSWER
Q67:	Agree	Nuetral	Agree
Q68:	NO ANSWER	NO ANSWER	agreed, with the exception of performance stability. I often have problems running > 50 msg/sec through the eheap, when I know it's supposed to be able to do more. Of course, that might be a perf. problem with my app ;)
Q69:	Agree	Agree	Agree
Q70:	NO ANSWER	NO ANSWER	certainly it's led to better fault isolation than when using something like RMI/RPC.
Q71:	Strongly agree	Agree	Agree
Q72:	NO ANSWER	Answer based on the Java API.	NO ANSWER
Q73:	Strongly agree	Agree	Strongly agree
Q74:	NO ANSWER	NO ANSWER	NO ANSWER
Q75:	Strongly agree	Nuetral	Nuetral
Q76:	NO ANSWER	I have a vested interest here :)	NO ANSWER
Q77:	Agree	Agree	Nuetral
Q78:	NO ANSWER	NO ANSWER	NO ANSWER
Q79:	Better than expected	Acceptable	Tolerable
Q80:	NO ANSWER	NO ANSWER	NO ANSWER
Q81:	-Expand API for RPC. -Put event heap on a network protocol that allows for prioritizing traffic. Event heap can't be disabled by extensive network traffic	NO ANSWER	NO ANSWER

Continued on next page

Continued from previous page

	Respondent 7	Respondent 8	Respondent 9
Q82:	10 sec	30s to a few minutes.	most msgs I expire in 1-5 seconds. Others I'll expire in much longer time-periods. it depends on what they are.
Q83:	Java, C++	C++, VB	NO ANSWER
Q84:	none	NO ANSWER	NO ANSWER
Q85:	No Response	No Response	No Response
Q86:	NO ANSWER	NO ANSWER	NO ANSWER

Table 3 - Developer Survey Responses, Respondents Seven through Nine

	Respondent 10	Respondent 11	Respondent 12
Q1:	No Response	No Response	No Response
Q2:	Intermediate	Intermediate	Master
Q3:	NO ANSWER	NO ANSWER	NO ANSWER
Q4:	Amateur	Intermediate	No experience
Q5:	NO ANSWER	NO ANSWER	NO ANSWER
Q6:	Amateur	Intermediate	No experience
Q7:	NO ANSWER	NO ANSWER	NO ANSWER
Q8:	I built a system to send user barcodes from the control panel machine to a smart board in the iRoom. I developed the barcode entry system and another person developed the UI component. Our applications only had to know what the other expected in terms of fields being sent. Other than that we developed our code independently and the first time we put it together it worked. It was really amazing to see it work for the first time.	Inter application communication; job scheduling, data reporting, control	n/a
Q9:	No Response	No Response	No Response
Q10:	Both equally	Application Ensembles and an occasional monolithic application	Application Ensembles and an occasional monolithic application
Q11:	NO ANSWER	NO ANSWER	NO ANSWER
Q12:	Agree	Agree	Agree
Q13:	NO ANSWER	NO ANSWER	NO ANSWER
Q14:	More frequent faults which effect only a part of the system and require one or two minutes to reset that component	More frequent faults which effect only a part of the system and require one or two minutes to reset that component	More frequent faults which effect only a part of the system and require one or two minutes to reset that component
Q15:	I am not expert but my experience is that usually small parts of the system break and you don't notice them until enough of the system stops working.	NO ANSWER	NO ANSWER
Q16:	Yes	No	No
Q17:	It depends on the type of application. I don't want any latency with streaming audio/video with another user but some other applications do matter	NO ANSWER	NO ANSWER

Continued on next page

Continued from previous page

	Respondent 10	Respondent 11	Respondent 12
Q18:	No	No	No
Q19:	You have to assume the developers of such systems can resolve this issue for themselves and it is not your job to hold their hands for this aspect of it.	NO ANSWER	NO ANSWER
Q20:	No	No	No
Q21:	Again, let the application programmer resolve this. An interactive workspace infrastructure needs to be a general purpose system that is FAST.	NO ANSWER	Transactions among supported components is a crucial feature. the CIFE infrastructure now uses the event heap as the distribution mechanism.
Q22:	No Response	No Response	No Response
Q23:	2	2	6
Q24:	1	2	4
Q25:	6	3	1
Q26:	3	2	3
Q27:	4	3	2
Q28:	5	2	5
Q29:	Actually I figure security it important but it is best to assume that the interactive workspace system exists in a safe environment.	NO ANSWER	NO ANSWER
Q30:	No Response	No Response	No Response
Q31:	8	3	NO ANSWER
Q32:	6	1	NO ANSWER
Q33:	7	3	NO ANSWER
Q34:	1	6	1
Q35:	2	4	2
Q36:	9	4	NO ANSWER
Q37:	3	2	3
Q38:	4	5	NO ANSWER
Q39:	10	3	NO ANSWER
Q40:	5	6	5
Q41:	Not to be picky but filling out this form is not the most intuitive thing. It is difficult to scroll between the 10 selections and keep them all in mind when assigning unique point values	NO ANSWER	NO ANSWER
Q42:	No Response	No Response	No Response
Q43:	Yes	Yes	Yes
Q44:	NO ANSWER	NO ANSWER	Everything changes...

Continued on next page

Continued on previous page

	Respondent 10	Respondent 11	Respondent 12
Q45:	Content based routing where receiver applications choose which messages to accept based on their content	Content based routing where receiver applications choose which messages to accept based on their content	Content based routing where receiver applications choose which messages to accept based on their content
Q46:	NO ANSWER	NO ANSWER	NO ANSWER
Q47:	The receiver of a message knows best whether or not it should receive the message	The receiver of a message knows best whether or not it should receive the message	The receiver of a message knows best whether or not it should receive the message
Q48:	NO ANSWER	NO ANSWER	NO ANSWER
Q49:	Never	Sometimes	Often
Q50:	NO ANSWER	NO ANSWER	NO ANSWER
Q51:	Never	Often	Very often
Q52:	NO ANSWER	NO ANSWER	NO ANSWER
Q53:	Very often	A little	Sometimes
Q54:	NO ANSWER	NO ANSWER	NO ANSWER
Q55:	No Response	Never	Sometimes
Q56:	NO ANSWER	NO ANSWER	NO ANSWER
Q57:	Push based, where you register and the infrastructure gives you a callback each time a matching message is submitted by some sender	Push based, where you register and the infrastructure gives you a callback each time a matching message is submitted by some sender	Push based, where you register and the infrastructure gives you a callback each time a matching message is submitted by some sender
Q58:	NO ANSWER	NO ANSWER	NO ANSWER
Q59:	RMI JINI	Socket	direct socket; Jini; publish-subscribe; database centric(all apps read/write same database)
Q60:	No Response	No Response	No Response
Q61:	Agree	Agree	No Response
Q62:	Programming the application was fine but as the eheap changed I had to adapt my code to work with it.	NO ANSWER	NO ANSWER
Q63:	Nuetral	Agree	No Response
Q64:	NO ANSWER	NO ANSWER	NO ANSWER
Q65:	Disagree	Agree	Disagree
Q66:	This is a major concern with developing any distributed-like application. Debugging is hard and I am sure this could be a good PhD topic for someone	NO ANSWER	NO ANSWER
Q67:	Nuetral	Agree	Agree
Q68:	All things break, but that doesn't take away from the fact that it is frustrating.	NO ANSWER	Architecture great. Early versions have had bugs. We sometimes have troubles dealing with network latency both when the net is busy and when the distance between app devices is great.

Continued on next page

Continued on previous page

	Respondent 10	Respondent 11	Respondent 12
Q69:	Nuetral	Strongly agree	Strongly agree
Q70:	NO ANSWER	NO ANSWER	NO ANSWER
Q71:	Agree	Agree	Nuetral
Q72:	NO ANSWER	NO ANSWER	It has been. It is hard to predict when we will want to send around first class objects.
Q73:	Agree	Agree	Agree
Q74:	NO ANSWER	NO ANSWER	NO ANSWER
Q75:	Disagree	Agree	Nuetral
Q76:	NO ANSWER	NO ANSWER	relatively easy, not absolutely, as in buying a can of soda from a vending machine...
Q77:	Disagree	Agree	Agree
Q78:	A distributed eHeap architecture would prove useful in many circumstances. I will not go into detail now but if you are interested you can email me to discuss it.	NO ANSWER	We will find the limits of this architecture.
Q79:	Better than expected	Acceptable	Acceptable
Q80:	NO ANSWER	NO ANSWER	We had latency problems running the iRoom in Gates and the E-H in the CIFE lab. probably, these problems were network related. Direct app connection (without using the E-H) would lower network utilization.
Q81:	No, it worked great for what I needed.	NO ANSWER	NO ANSWER
Q82:	Typically very short	10s	minutes
Q83:	Java, C++, Perl	java, cpp	C++, Java
Q84:	NO ANSWER	my own socket layer	NO ANSWER
Q85:	No Response	No Response	No Response
Q86:	NO ANSWER	NO ANSWER	NO ANSWER

Table 4 - Developer Survey Responses, Respondents Ten through Twelve

Compiled Results

The compiled results of the developer survey answers as presented in the previous section can be found below. For each question percentage breakdowns are given for multiple choice style questions, and a list of all responses is given for text questions. The compiled results were created automatically by the SALGains web site [1].

1. [EXPLANATORY: DON'T ANSWER] The questions in this first section are related to your background. (N=0)

2. In terms of my general knowledge, comfort, experience and ability as a general application developer, I would consider myself: (N=12)

No experience	0% (0)
Novice	0% (0)
Amateur	0% (0)
Intermediate	42% (5)
Master	50% (6)
Expert	8% (1)

3. Comments

I prefer proficient, master doesn't sound very modest :)
In some areas (Mac OS X, Java) I might be a master, but in others (Windows) I'm an amateur.
Was a master years ago, but am rusty.

4. In terms of my general knowledge, comfort, experience and ability as an application developer for interactive workspaces, I would consider myself: (N=12)

No experience	8% (1)
Novice	0% (0)
Amateur	17% (2)
Intermediate	50% (6)
Master	17% (2)
Expert	8% (1)

5. Comments

I have been developing applications in the iRoom for about one year.
Mostly involved at higher levels, but have done a lot of it.

6. In terms of my general knowledge, comfort, experience and ability as an application developer for the Event Heap, I would consider myself: (N=12)

No experience	8% (1)
Novice	0% (0)
Amateur	17% (2)
Intermediate	50% (6)
Master	8% (1)
Expert	17% (2)

7. Comments

Again, I've written little code myself, but have been involved in design of several application suites
Relatively speaking :)
see below

8. Please give a brief description of the Event Heap based applications you have done and the specific Event Heap functionality you have used:

Dataheap, smartpresenter, that sort of thing.
I built a system to send user barcodes from the control panel machine to a smart board in the iRoom. I developed the barcode entry system and another person developed the UI component. Our applications only had to know what the other expected in terms of fields being sent. Other than that we developed our code independently and the first time we put it together it worked. It was really amazing to see it work for the first time.
I have used the event heap to incorporate new physical user interfaces into the room. I use the event heap as a communications channel from the device proxy application to the device client application. I also have implemented a patch panel intermediary application that translates device events to application events so that users can specify the focus of a particular physical device. The event heap provides interesting qualities that allow for a software engineering technique that allows this decoupling and also provides for a higher grain of reuse.
I have written or helped write many applications the use the EventHeap (iPong, GroupBoard, iWall, PatchPanel) which, combined, use most of the functionality of the EventHeap (sending, receiving, logging, intercepting, etc.). I have also implemented a partial (only for sending) EventHeap API in C++, so I am experienced with the EventHeap Wire Protocol.
I have written various point-point apps as well as multicast/broadcast type apps.
Inter application communication; job scheduling, data reporting, control
n/a
Primarily example pieces, worked with actual implementers on the design of eHeap interfaces Both CIFE, and the iStuff and iWall HCI research work
The iButtons (first version). Simple event passing. I created my own event types and listeners for them.
-Time controller to broadcast time messages to applications -event heap communication add-ons to MS apps and proprietary 4D app
use eventheap to start a videoconferencing program on another computer in the iRoom.
WorkspaceNavigator. We use the Event Heap to trigger capture at regular intervals on all the devices in the iroom. Also, iButtons send events which our application picks up.

9. [EXPLANATORY: DON'T ANSWER] The second section asks general questions about interactive workspaces not specific to developers or the Event Heap. (N=0)

10. ‘Applications’ for interactive workspaces can either be collections of small programs (ensembles) or monolithic applications which run across the whole space. Which of these two will be the more common case: (N=12)

Application Ensembles only	0% (0)
Application Ensembles and an occasional monolithic application	92% (11)
Both equally	8% (1)
Mostly monolithic applications, but some ensembles	0% (0)
Monolithic applications only	0% (0)
Don't know.	0% (0)

11. Comments:

Not sure what is meant by "the whole space."
 This is generally what I've seen in the iRoom. I don't want to rule out either, but the "UNIX model" of many small apps that work together seems to mesh well with dynamic, distributed environments.

12. Interactive workspaces are likely to be dynamic on short time scales due to application and device failure and the entrance and departure of portable devices: (N=12)

Strongly agree	17% (2)
Agree	83% (10)
Neutral	0% (0)
Disagree	0% (0)
Strongly disagree	0% (0)
Average = 1.83, S.D. = 0.37	

13. Comments:

Generally true, but varies widely between different workspaces. It is possible to have a completely static workspace (in terms of entrance and departure of devices) that is also relatively stable (in terms of device failure).
 I think they will eventually become dynamic on short time scales (where short is on the order of an hour or so). I think that's the right time scale because that's how long meetings tend to last. Once everyone is in the room, they tend to stay for awhile, and the system is pretty stable for the duration of the meeting. Device failures, etc occur, but I don't think this will occur often enough for change to occur on the order of minutes.

14. Which is more tolerable: (N=12)

Occasional total system failures which require ten or more minutes to restart	17% (2)
More frequent faults which effect only a part of the system and require one or two minutes to reset that component	83% (10)
Not sure	0% (0)

15. Comments:

Even if it is the part of the system that you are working on 1 minute is a lot better than 10. But the degree of frequency can sway this answer the other direction.
I am not expert but my experience is that usually small parts of the system break and you don't notice them until enough of the system stops working.
It is preferable to have a single procedure for restarting anything, for simplicity.
The answer to 14 depends on how frequent "more frequent" is.
Users learn to deal with "routine" problems, but have a hard time dealing with total catastrophic failures.

16. Would you accept reduced performance (up to 1 s latency) from an interactive workspaces system infrastructure to get guaranteed message delivery? (N=12)

Yes	42% (5)
No	58% (7)

17. Comments

16 is hard to answer. I answered yes because it appears that there would be cases where 1s extra latency is worth the guarantee. However, it would be nice if the penalty is only imposed on the messages where guaranteed delivery is needed.
but not for events like mouse movements
Guaranteed message delivery is important, but I wouldn't be too tolerant of extra latency (beyond TCP) caused by the interactive workspace infrastructure, since TCP is pretty fast and it already has some guarantees, which are often enough.
I would accept the reduced performance for additional features if I could choose the faster message delivery mechanism in a performance sensitive application. In other words, I wouldn't accept it if performance would be universally degraded.
It depends on the type of application. I don't want any latency with streaming audio/video with another user but some other applications do matter
Not suitable for UI work

18. Would you accept reduced performance (up to 1 s latency) from an interactive workspaces system infrastructure if the system guaranteed consistency (all components see the same events in the same order)? (N=12)

Yes	25% (3)
No	75% (9)

19. Comments

but yes for database like apps (meeting capture, etc)
Same as 17.
same comment as above
Similar. We're doing UI research, and need to reduce latency
Yes, but I'd like a way to give up consistency in exchange for less latency. Some applications really need low latency and don't care about consistency.
You have to assume the developers of such systems can resolve this issue for themselves and it is not your job to hold their hands for this aspect of it.

20. Would you accept reduced performance (up to 1 s latency) from an interactive workspaces system infrastructure if it supported transactions among connected components? (N=12)

Yes	8% (1)
No	92% (11)

21. Comments

Again, let the application programmer resolve this. An interactive workspace infrastructure needs to be a general purpose system that is FAST.
I don't know what "transaction among connected components" mean
I don't think I understand the question - what does an interactive workspaces infrastructure do besides support transactions among connected components? If you mean direct component-to-component transactions, I would expect it to be as fast as direct sockets.
It might be acceptable for components which require transactional capability. If this latency is paid for by components that don't need those guarantees, that is unacceptable.
same comment as above
Transactions among supported components is a crucial feature. the CIFE infrastructure now uses the event heap as the distribution mechanism.
Transactions appear too heavyweight.
Would have to be an option, not all eHeap communication

22. [EXPLANATORY: DON'T ANSWER] Please rank the following six general features according to how important they are for an interactive workspace coordination infrastructure. Please assign a unique value for each feature, 1 being most important: (N=0)

23. Support for heterogeneous devices and hardware platforms (N=12)

1	25% (3)
2	42% (5)
3	8% (1)
4	0% (0)
5	0% (0)
6	25% (3)

24. Support for a variety of development platforms (Java, C++, .NET, etc.) (N=12)

1	17% (2)
2	8% (1)
3	25% (3)
4	33% (4)
5	8% (1)
6	8% (1)

25. Ability to easily integrate legacy applications (COTS and In-house) (N=12)

1	17% (2)
2	17% (2)
3	25% (3)
4	8% (1)
5	8% (1)
6	25% (3)

26. Provision for new hardware and software that may be released in the future. (N=12)

1	8% (1)
2	8% (1)
3	25% (3)
4	33% (4)
5	17% (2)
6	8% (1)

27. Reliability and Stability (N=12)

1	25% (3)
2	33% (4)
3	17% (2)
4	17% (2)
5	8% (1)
6	0% (0)

28. Security (N=12)

1	0% (0)
2	17% (2)
3	8% (1)
4	0% (0)
5	50% (6)
6	25% (3)

29. Comments on the ranking of the features in the previous six questions:

<p>23 and 24 seem related, but my ranking is based on the following difference: I care about some language being supported for each platform, and not every language for each platform. it 26 is vague. Once the system goes beyond the "demo mode" and needs to be deployed, security and reliability become very important, although this is often overlooked by researchers.</p>
<p>A stack ranking is hard. All of the flexibility components are important Reliability is important, but less expected of research software Security does matter, but not so much in HCI research</p>
<p>Actually I figure security it important but it is best to assume that the interactive workspace system exists in a safe environment.</p>
<p>if I am a user, I don't care how you wrote the code, so development platform ranks last.</p>
<p>Support for a wide variety of hardware (both new and old) and development environments (my ranks 2,3,and 4) are all pretty much tied. Reliability and Stability is a #1 priority for anything that so many other applications will rely on. Security, though I ranked it as #5, can be very important in certain situations, perhaps even as high as #2 (it's not important in research, but in industry is can be required).</p>
<p>We've been able to do a lot of interesting things with little support for heterogeneous devices/platforms. (we basically only run windows in the room, right?). The most important thing seems to have been integrating legacy software, because that's how we got most of our demo-functionality, right?</p>

30. [EXPLANATORY: DON'T ANSWER] Please rank the following ten developer features according to how important they are for an interactive workspace coordination infrastructure. Please assign a unique value for each feature, 1 being the most important: (N=0)

31. Event expiration (N=10)

1	0% (0)
2	10% (1)
3	10% (1)
4	10% (1)
5	10% (1)
6	10% (1)
7	0% (0)
8	20% (2)
9	20% (2)
10	10% (1)

32. Extensibility of application ensembles written with an infrastructure (N=10)

1	30% (3)
2	10% (1)
3	20% (2)
4	0% (0)
5	10% (1)
6	10% (1)
7	20% (2)
8	0% (0)
9	0% (0)
10	0% (0)

33. Ability of infrastructure API to express different types of coordination (N=10)

1	30% (3)
2	0% (0)
3	20% (2)
4	10% (1)
5	0% (0)
6	10% (1)
7	10% (1)
8	10% (1)
9	10% (1)
10	0% (0)

34. Simplicity and ease of learning the infrastructure API (N=12)

1	33% (4)
2	8% (1)
3	0% (0)
4	8% (1)
5	8% (1)
6	17% (2)
7	0% (0)
8	0% (0)
9	17% (2)
10	8% (1)

35. Ease of debugging application ensembles written with the infrastructure (N=11)

1	0% (0)
2	27% (3)
3	0% (0)
4	18% (2)
5	9% (1)
6	0% (0)
7	18% (2)
8	18% (2)
9	9% (1)
10	0% (0)

36. Portability of applications written with the infrastructure to new environments (N=10)

1	0% (0)
2	0% (0)
3	20% (2)
4	20% (2)
5	0% (0)
6	10% (1)
7	20% (2)
8	10% (1)
9	10% (1)
10	10% (1)

37. Support for different communication patterns (point-to-point, broadcast, etc.) (N=11)

1	9% (1)
2	36% (4)
3	27% (3)
4	18% (2)
5	0% (0)
6	0% (0)
7	0% (0)
8	9% (1)
9	0% (0)
10	0% (0)

38. Transparent communication (ability to monitor communications and easily interpret message contents) (N=10)

1	0% (0)
2	10% (1)
3	10% (1)
4	20% (2)
5	40% (4)
6	0% (0)
7	0% (0)
8	10% (1)
9	0% (0)
10	10% (1)

39. Ordering of sent messages and at most once reception (N=10)

1	10% (1)
2	0% (0)
3	20% (2)
4	0% (0)
5	0% (0)
6	20% (2)
7	20% (2)
8	10% (1)
9	10% (1)
10	10% (1)

40. Support for transmission of complex object types (N=11)

1	0% (0)
2	0% (0)
3	0% (0)
4	0% (0)
5	18% (2)
6	27% (3)
7	0% (0)
8	9% (1)
9	9% (1)
10	36% (4)

41. Comments on ranking developer features:

Although I tried my best to order, I still feel there is some randomness here. Maybe this is not the right question to ask?
i ranked it based on the capability of the system, since an easy to learn system that doesn't do what you need to do is not useful.
Not to be picky but filling out this form is not the most intuitive thing. It is difficult to scroll between the 10 selections and keep them all in mind when assigning unique point values
The most important thing is for the API to support what needs to be done, by making it not only possible but easy to accomplish the required coordination. Specific features like event expiration, sequencing, and complex data types are important, but only if the theory of operation of the infrastructure requires them (Publish-Subscribe wouldn't have a notion of event expiration, for example) The ability to log and/or intercept communication is very important, thought not as important as building the communicating pieces in the first place.
This was very hard to do, both physically (it's hard to keep track of the ordering and the questions when you have to scroll) and because some features simply have to be there, so it's not a stack ranking

42. [EXPLANATORY: DON'T ANSWER] The following nine questions are general questions about developing for interactive workspaces: (N=0)

43. Do you foresee adapting your interactive workspaces applications to work with other applications in the future? (N=11)

Yes	73% (8)
No, the application set will be written once and then left alone	9% (1)
Not sure	18% (2)

44. Comments:

Almost every app I wrote for the iRoom had to be adapted later for some other purpose, whether I planned it or not. It became very important to think about extensibility before writing new apps, so it would be easier to adapt when (not if) necessary.
Everything changes...
I think that the infrastructure enables this to a higher degree with intermediation. Applications can send and receive generic application events that can be translated to different generic application events

45. What kind of message routing fits best with the type of applications you usually write for interactive workspaces? (N=11)

Explicitly addressed messages sent to a specific named entity	36% (4)
Content based routing where receiver applications choose which messages to accept based on their content	64% (7)

46. Comments:

Though the routing I did was usually content based, it was typically based solely on the EventType (and sometimes on other field).

Where "content" is event type. We need many-to-many, for an unbound number of many But, the content matching we use is relatively simple--name and id primarily

47. What is more true in your application: (N=11)

The receiver of a message knows best whether or not it should receive the message	82% (9)
The sender knows best who should receive its message	18% (2)

48. Comments:

see above (most routing I did was content based)

The goal, actually, is to decouple the sender and receiver as much as possible. This way, we support generic input and generic receivers

49. How often do you use point-to-point routing in your applications: (N=11)

Very often	18% (2)
Often	18% (2)
Sometimes	36% (4)
A little	18% (2)
Never	9% (1)
Average = 2.82, S.D. = 1.19	

50. Comments:

Basically PointRight

There is occasionally need for point-to-point, but it usually is required for interaction with an existing protocol, not because point-to-point is the best pattern.

51. How often do you use multicast routing (one sender to a group of receivers) in your applications:
(N=11)

Very often	36% (4)
Often	18% (2)
Sometimes	18% (2)
A little	0% (0)
Never	27% (3)
Average = 2.64, S.D. = 1.61	

52. Comments:

i often use multicast or broadcast routing for a query, and unicast replies with the answer.
This is perhaps the most important of often used feature of the EventHeap for the apps I have written.

53. How often do you use broadcast routing (one sender to all receivers in the environment) routing in your applications: (N=11)

Very often	45% (5)
Often	0% (0)
Sometimes	27% (3)
A little	9% (1)
Never	18% (2)
Average = 2.55, S.D. = 1.56	

54. Comments:

Sometimes I use very broad multicast, but never true broadcast.

55. How often do you use anycast routing (one sender to exactly one of a group of receivers) in your applications: (N=10)

Very often	0% (0)
Often	0% (0)
Sometimes	60% (6)
A little	0% (0)
Never	40% (4)
Average = 3.8, S.D. = 0.98	

56. Comments:

Not completely sure about this one
When this is needed, I often use a secondary field in the event to indicate which receiver is meant.

57. Which type of delivery tends to work better for your applications? (N=11)

Pull based, where applications pull in messages as they need them	9% (1)
Push based, where you register and the infrastructure gives you a callback each time a matching message is submitted by some sender	91% (10)

58. Comments:

Using registration and callbacks has the advantage of not blocking the listener. Quite often, an application wants to listen for certain kinds of messages for the lifetime of the applications, so registration is easiest.

59. Please list any distributed computing coordination techniques that you have used in the past: (e.g. Direct socket, RMI/RPC including Jini, publish-subscribe, etc.)

Direct socket
direct socket publish-subscribe distributed objects (Obj-C/Cocoa version of RMI/RPC)
direct socket, rmi/rpc
Direct socket, RMI/RPC, pub-sub
direct socket, shared memory
direct socket; Jini; publish-subscribe; database centric(all apps read/write same database)
RMI JINI
Socket

60. [EXPLANATORY: DON'T ANSWER] The following fourteen questions specifically relate to developing using the Event Heap: (N=0)**61. It was straightforward to adapt my previous programming techniques for use with the Event Heap: (N=10)**

Strongly agree	40% (4)
Agree	50% (5)
Neutral	0% (0)
Disagree	10% (1)
Strongly disagree	0% (0)
Average = 1.8, S.D. = 0.87	

62. Comments:

Main disagreement is with error handling. The current API requires a very messy nesting to catch errors that really are best left uncaught. In general, applications should only catch errors that they can work around. Errors that mean that the system is corrupted or unstable are usually not worth catching, as there is nothing the application can do about it. Crashing is as good a response as anything, especially for research

Programming for the EventHeap involves thinking about problems that don't occur in other environments, but this has more to do with the applications being written than the EventHeap API. For example, the possibility of multiple users interacting with an application at the same time is an issue with EventHeap applications, because it is possible, while it generally isn't with a normal application.

Programming the application was fine but as the eheap changed I had to adapt my code to work with it.

63. The current Event Heap API is sufficient to express all interactive workspace coordination tasks I have needed to code in a reasonably straightforward manner: (N=10)

Strongly agree	20% (2)
Agree	50% (5)
Neutral	20% (2)
Disagree	10% (1)
Strongly disagree	0% (0)
Average = 2.2, S.D. = 0.87	

64. Comments:

I wanted to implement RPC using events and I had to think about how I would do it. It is relatively simple, just not intuitive.

While the EventHeap is good for many coordination tasks, there are some tasks it does not handle. Most notable are bulk data transfer and persistent state. There are other iROS APIs to handle some of these tasks, and how well they handle them is not what this survey is about - however, the point is that the EventHeap cannot stand alone.

65. Compared with other applications I have developed, Event Heap applications are very easy to debug: (N=11)

Strongly agree	0% (0)
Agree	55% (6)
Neutral	18% (2)
Disagree	27% (3)
Strongly disagree	0% (0)
Average = 2.73, S.D. = 0.86	

66. Comments:

EventHeap apps are not significantly easier to debug than other distributed and/or multithreaded apps. The ability to see Events on the EventHeap helps some, but there are often similar methods to debug non-EventHeap apps.

I wouldn't say it is easier to debug with EventHeap.

Not much experience myself

This is a major concern with developing any distributed-like application. Debugging is hard and I am sure this could be a good PhD topic for someone

Though at first I didn't know about the application where you could see all the events going through.

67. The Event Heap is very stable and reliable: (N=11)

Strongly agree	0% (0)
Agree	55% (6)
Neutral	36% (4)
Disagree	9% (1)
Strongly disagree	0% (0)
Average = 2.55, S.D. = 0.66	

68. Comments:

agreed, with the exception of performance stability. I often have problems running > 50 msg/sec through the eheap, when I know it's supposed to be able to do more. Of course, that might be a perf. problem with my app ;)
All things break, but that doesn't take away from the fact that it is frustrating.
Architecture great. Early versions have had bugs. We sometimes have troubles dealing with network latency both when the net is busy and when the distance between app devices is great.
I have hardly ever seen the EventHeap server misbehave, and then only when I was actively trying to push it to its limits. The EventHeap client API does have some quirks (it has been known to hog the CPU when it can't find the server) but it generally runs very smoothly.
In Java, yes. With the C++ API, no.

69. It has been easier and quicker to develop stable applications and application sets using the Event Heap than it has been with previous systems I have used: (N=11)

Strongly agree	27% (3)
Agree	45% (5)
Neutral	18% (2)
Disagree	9% (1)
Strongly disagree	0% (0)
Average = 2.09, S.D. = 0.9	

70. Comments:

certainly it's led to better fault isolation than when using something like RMI/RPC.
It's not the stability, it's the flexibility I admire
No system is the perfect development environment, but I would be hard pressed to find a better one. There are certainly many worse ones.
The system we developed would have been a nightmare without the Event Heap. It's not too stable however.

71. The current Event Heap object structure for events is sufficient for my application development needs: (N=11)

Strongly agree	27% (3)
Agree	64% (7)
Neutral	9% (1)
Disagree	0% (0)
Strongly disagree	0% (0)
Average = 1.82, S.D. = 0.57	

72. Comments:

Answer based on the Java API.
It has been. It is hard to predict when we will want to send around first class objects.
It is sufficient, but it would be convenient to be able to store more complex, hierarchical data types (like lists or hashes). Currently, these are usually flattened into a string representation in some arbitrary way. Having a standard way to do this might be beneficial to developers.

73. The extensible event format used by the Event Heap which allows extra fields to be added to events has been useful: (N=11)

Strongly agree	55% (6)
Agree	36% (4)
Neutral	9% (1)
Disagree	0% (0)
Strongly disagree	0% (0)
Average = 1.55, S.D. = 0.66	

74. Comments:

All but the most simple messages I have created use extra fields. In fact, I hardly use the default fields at all, except for EventType and TimeToLive.
I used this extensively to add extra information that receivers need.

75. Downloading the iROS/Event Heap system and getting it working was easy: (N=11)

Strongly agree	9% (1)
Agree	27% (3)
Neutral	55% (6)
Disagree	9% (1)
Strongly disagree	0% (0)
Average = 2.64, S.D. = 0.77	

76. Comments:

I didn't like the java baggage :)
I have a vested interest here :)
I own a Mac, so I have not had much experience with the iROS install process. From a developer's standpoint, getting started with developing was a snap, since it only required the iROS.jar file.
relatively easy, not absolutely, as in buying a can of soda from a vending machine...

77. The fact that the Event Heap is a centralized message exchange system running on a single server machine is an advantage: (N=11)

Strongly agree	27% (3)
Agree	55% (6)
Neutral	9% (1)
Disagree	9% (1)
Strongly disagree	0% (0)
Average = 2, S.D. = 0.85	

78. Comments:

A distributed eHeap architecture would prove useful in many circumstances. I will not go into detail now but if you are interested you can email me to discuss it.

For research systems, centralized is fine, and easier to debug and maintain.

Having all messages go through a single point has many advantages (most notably logging), but there are also disadvantages. Perhaps the option of a direct point-to-point could be added in such a way that logging is still possible (but sending messages both the other client and to the server, for example).

We will find the limits of this architecture.

79. How is the performance (bandwidth, throughput, latency) of the Event Heap for the applications you use? (N=11)

Unusable	0% (0)
Poor	0% (0)
Tolerable	9% (1)
Acceptable	36% (4)
Better than expected	55% (6)
Superior	0% (0)

80. Comments:

Performance has almost never been a problem. Only PointRight-over-EventHeap has raised concerns, and even then the EventHeap performed very well. Still, there is always room for performance improvements - the EventHeap will never be as efficient as direct sockets, but I believe there are still areas that can be squeezed.

We had latency problems running the iRoom in Gates and the E-H in the CIFE lab. probably, these problems were network related. Direct app connection (without using the E-H) would lower network utilization.

81. Are there any specific areas of Event Heap performance that you would like improved?

An even lower latency path would be useful, but this isn't just the eHeap

-Expand API for RPC. -Put event heap on a network protocol that allows for prioritizing traffic. Event heap can't be disabled by extensive network traffic

I think the primary performance improvements could be made with the size of the Event packets. Perhaps an option to not include many of the default fields?

No, it worked great for what I needed.

82. What are the typical event expiration times you use for your applications?

10 sec
10s
2 sec
30 seconds
30s to a few minutes.
From a few hundred milliseconds to several seconds (depending on the application, of course).
immediate
minutes
most msgs I expire in 1-5 seconds. Others I'll expire in much longer time-periods. it depends on what they are.
Typically very short

83. What platforms are important for the Event Heap to support (current and desired)? (e.g. C++, .NET, Perl, HTTP, Visual Basic, Cocoa, etc.)

All my stuff is in java.
c++, http
C++, Java
C++, VB
Having a pure C reference implementation would make is possible to port the EventHeap API to many different platforms, since nearly all have C bindings. Almost every platform is potentially important, since each application will have its own unique platform and language requirements.
HTTP, Java, C++, VB, Cocoa
Java, C++
Java, C++, Perl
java, cpp
NO ANSWER

84. What other systems did you consider using besides iROS/the Event Heap, if any? What factors made you decide in favor of the Event Heap/iROS (e.g. easy to download)?

It was there
my own socket layer
none
None - I was developing specifically for the EventHeap/iRoom.

85. [EXPLANATORY: DON'T ANSWER] Concluding Section (N=0)**86. Any overall suggestions and/or comments?**

a good tutorial for people just starting up.
Better documentation. The javadocs are helpful, but a full-scale EventHeap tutorial explaining everything, including some of the more confusing quirks (ACTUAL/FORMAL/VIRTUAL, post vs. template type, etc.) would help many beginning developers. The currently posted example code is not enough for anything beyond the basics, and, besides, the link to the "sender" is broken.
sorry, i ran out of time to finish
The iROS has created an excellent infrastructure for HCI research in interactive environments. Most reliability problems seem to be with the C++ interface, or other parts of the iROS than the eHeap proper.

Administrator Survey

The remainder of this appendix contains the material and results related to the administrator survey.

Web Survey Questions

The questions that were presented on the web site to administrators participating in the survey are shown on the succeeding pages. The actual pages shown were generated from a printout of the web site to a PDF file.

Student Assessment of Learning Gains

Instrument

Stanford University: Interactive Workspaces 2-2 2, 2002

[Go to your List of Courses](#)

[Log Out](#)

[Go to your Course Options](#)

[Go back](#)

Your students will see the questions as they appear on this page.

1. [EXPLANATORY: DON'T ANSWER] The first section consists of questions about your background.

- Yes
- No

2. In terms of my general knowledge, comfort, experience and ability with computers in general, I would consider myself:

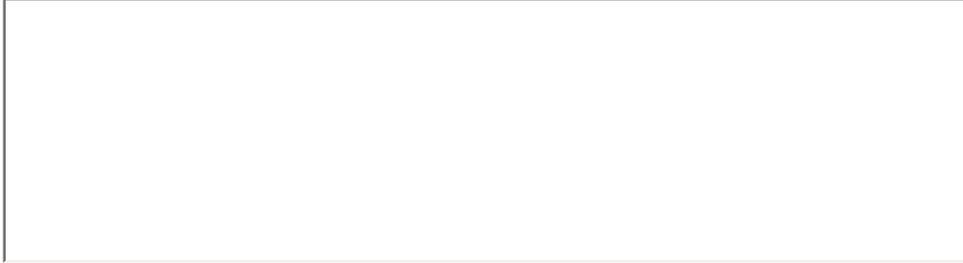
- No experience
- Novice
- Amateur
- Intermediate
- Master
- Expert

3. Comments

4. In terms of my general knowledge, comfort, experience and ability with interactive workspace type environments, I would consider myself:

- No experience
- Novice
- Amateur
- Intermediate
- Master
- Expert

5. Comments



6. In terms of my general knowledge, comfort, experience and ability with administering iROS and the Event Heap, I would consider myself:

- No experience
- Novice
- Amateur
- Intermediate
- Master
- Expert

7. Comments



8. Please give a brief description of the type of iROS based interactive workspace you have administered and the types of functionality being used in that space:



9. [EXPLANATORY: DON'T ANSWER] The second section contains questions not specific to administration or the iROS/Event Heap system.

- Yes
- No

10. 'Applications' for interactive workspaces can either be collections of small programs (ensembles) or monolithic applications which run across the whole space. Which of these two will be the more common case:

- Application Ensembles only
- Application Ensembles and an occasional monolithic application
- Both equally
- Mostly monolithic applications, but some ensembles
- Monolithic applications only
- Don't know.

11. Comments:**12. Interactive workspaces are likely to be dynamic on short time scales due to application and device failure and the entrance and departure of portable devices:**

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

13. Comments:**14. Which is more tolerable:**

- Occasional total system failures which require ten or more minutes to restart
- More frequent faults which effect only a part of the system and require one or two minutes to reset that component
- Not sure

15. Comments:

16. Would you accept reduced performance (up to 1 s latency) from an interactive workspaces system infrastructure to get guaranteed message delivery?

- Yes
 No

17. Comments

18. Would you accept reduced performance (up to 1 s latency) from an interactive workspaces system infrastructure if the system guaranteed consistency (all components see the same events in the same order)?

- Yes
 No

19. Comments

20. Would you accept reduced performance (up to 1 s latency) from an interactive workspaces system infrastructure if it supported transactions among connected components?

- Yes
 No

21. Comments

22. [EXPLANATORY: DON'T ANSWER] Please rank the following six general features according to how important they are for an interactive workspace coordination infrastructure. Please assign a unique value for each feature, 1 being the most important:

- Yes
- No

23. Support for heterogeneous devices and hardware platforms

- 1
- 2
- 3
- 4
- 5
- 6

24. Support for a variety of development platforms (Java, C++, .NET, etc.)

- 1
- 2
- 3
- 4
- 5
- 6

25. Ability to easily integrate legacy applications (COTS and In-house)

- 1
- 2
- 3
- 4
- 5
- 6

26. Provision for new hardware and software that may be released in the future.

- 1
- 2
- 3
- 4
- 5
- 6

27. Reliability and Stability

- 1
- 2
- 3
- 4
- 5
- 6

28. Security

- 1
- 2
- 3
- 4
- 5
- 6

29. Comments on the ranking of the features in the previous six questions:**30. [EXPLANATORY: DON'T ANSWER] Please rank the following five administration related features according to importance for an interactive workspace coordination infrastructure. Please assign a unique value for each feature, 1 being the most important:**

- Yes
- No

31. Ease of administration

- 1
- 2
- 3
- 4
- 5

32. Ability to bring in new applications and have them integrate with existing applications and devices in the interactive workspace

- 1
- 2
- 3
- 4
- 5

33. Ability to monitor communications among applications in the workspace and view their contents in human readable form

- 1
- 2
- 3
- 4
- 5

34. Ease of debugging

- 1
- 2
- 3
- 4
- 5

35. Portability of applications written for one interactive workspace to new environments

- 1
- 2
- 3
- 4
- 5

36. Comments on administrator features and their importance:

37. [EXPLANATORY: DON'T ANSWER] The following three questions are general questions on administering interactive workspaces:

- Yes
- No

38. What level of administration is acceptable for an interactive workspace?

39. What level of administration has been necessary for your interactive workspace so far?

40. What is or will be the primary usage of your interactive workspace?

41. [EXPLANATORY: DON'T ANSWER] The following eleven questions are questions about administration of the Event Heap/iROS specifically:

- Yes
- No

42. Compared to other setups of interactive workspaces I've worked with, ones based on the Event Heap/iROS are easy to maintain and administer:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

43. Comments:

44. The Event Heap/iROS applications and functionality deployed in our interactive workspace are easy to debug:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

45. Comments:

46. The Event Heap/iROS is very stable and reliable:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

47. Comments:

48. The Event Heap is implemented as a server that runs on a single machine through which devices and applications communicate. Overall, this centralization has been an advantage:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

49. Comments:

50. Downloading the iROS/Event Heap system and getting it working was easy:

- Strongly agree

- Agree
- Neutral
- Disagree
- Strongly disagree

51. Comments:

52. Currently the only security for iROS based interactive workspaces is to firewall the interactive workspace off from the rest of the network. This is an acceptable level of security for our interactive workspace:

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

53. Comments:

54. How is the performance (bandwidth, throughput, latency) of the Event Heap for the applications in use in your interactive workspace?

- Unusable
- Poor
- Tolerable
- Acceptable
- Better than expected
- Superior

55. Comments:

56. Are there any specific areas of Event Heap performance that you would like to see improved?

57. What level of administration has been necessary for the iROS specific components of your interactive workspace so far?

58. What other systems did you consider using besides iROS/the Event Heap, if any? What factors made you decide in favor of the Event Heap/iROS (e.g. easy to download)?

59. What platforms are important for the Event Heap to support (current and desired)? (e.g. C++, .NET, Perl, HTTP, Visual Basic, Cocoa, etc.)

60. [EXPLANATORY: DON'T ANSWER] Concluding Section

- Yes
 No

61. Any overall suggestions and/or comments?

62. Your name and contact information (optional)

This site was created with funding courtesy of:

- [The Exxon Education Foundation](#)
- The NSF Chemistry Coalitions:
 - [New Traditions \(NT\)](#)
 - [ChemLinks](#)
 - [ModularChemistry \(MC2\)](#)
- [The NSF National Institute for Science Education](#)
- [The NSF SENCER Institutes](#)

Student Assessment of Learning Gains -- Instrument

<http://www.wcer.wisc.edu/salgains/instructor/LearningGainsEntryForm...>

Original Content Copyright ©1997 Elaine Seymour. All rights reserved. [Your comments](#) are welcome.

Ranking Assistance Sheet

As with the developer survey presented earlier in this appendix, the ranking mechanism was awkward in the on-line survey so an off-line pen and paper based ranking sheet was provided for those that took the survey. The ranking sheet given to administrators is shown on the following page.

Importance Ranking of Interactive Workspace Coordination Infrastructure Characteristics

Administrator Survey

Please rank the features in each section in order of importance, with '1' being most important, '2' next most important and so on. This form is to simplify the ranking process only. Please enter your final choices into the on-line form at <http://www.wcer.wisc.edu/salgains/student/default.asp>. Please use course ID 714707642 and password 'krowi'. Use your email address as your identifier.

General Characteristics

Survey Question #	Platform Feature/Characteristic	Ranking
23	Support for heterogeneous devices and hardware platforms	
24	Support for a variety of development platforms (Java, C++, .NET, etc.)	
25	Ability to easily integrate legacy applications (COTS and In-house)	
26	Provision for new hardware and software that may be released in the future.	
27	Reliability and Stability	
28	Security	

Administrator Specific Characteristics

Survey Question #	Platform Feature/Characteristic	Ranking
31	Ease of administration	
32	Ability to bring in new applications and have them integrate with existing applications and devices in the interactive workspace	
33	Ability to monitor communications among applications in the workspace and view their contents in human readable form	
34	Ease of debugging	
35	Portability of applications written for one interactive workspace to new environments	

Raw Results

The raw results for the administrator survey, per respondent, are listed over the next pages. Names of participants have been removed to preserve anonymity. All other information is presented exactly as it was submitted on the survey web page. Questions are not repeated in the tables as they may be found earlier in this appendix in the ‘Web Survey Questions’ administrator’s section.

	Respondent 1	Respondent 2	Respondent 3
Q1:	No Response	No Response	No Response
Q2:	Intermediate	Expert	Master
Q3:	NO ANSWER	NO ANSWER	NO ANSWER
Q4:	Intermediate	Master	Master
Q5:	NO ANSWER	NO ANSWER	NO ANSWER
Q6:	Novice	Master	Intermediate
Q7:	NO ANSWER	NO ANSWER	NO ANSWER
Q8:	NO ANSWER	iRoom, plus ad-hoc workspaces on Mac OS X laptops for demos and testing	iLoft, Presentations, multibrowse, PointRight. Collaborative meetings, demonstrations.
Q9:	No Response	No Response	No Response
Q10:	Application Ensembles and an occasional monolithic application	Application Ensembles and an occasional monolithic application	Application Ensembles and an occasional monolithic application
Q11:	NO ANSWER	NO ANSWER	NO ANSWER
Q12:	Strongly agree	Agree	Agree
Q13:	NO ANSWER	NO ANSWER	NO ANSWER
Q14:	Not sure	More frequent faults which effect only a part of the system and require one or two minutes to reset that component	Occasional total system failures which require ten or more minutes to restart
Q15:	NO ANSWER	NO ANSWER	Needs a 3 minute restart max.
Q16:	Yes	No	Yes
Q17:	NO ANSWER	Since our work is in the area of user interfaces for interactive workspaces, latencies of up to 1s would not be tolerable in any case.	NO ANSWER
Q18:	Yes	No	Yes
Q19:	NO ANSWER	Since our work is in the area of user interfaces for interactive workspaces, latencies of up to 1s would not be tolerable in any case.	NO ANSWER
Q20:	Yes	No	Yes
Q21:	NO ANSWER	Since our work is in the area of user interfaces for interactive workspaces, latencies of up to 1s would not be tolerable in any case.	NO ANSWER

Continued on next page

Continued from previous page

	Respondent 1	Respondent 2	Respondent 3
Q22:	No Response	No Response	No Response
Q23:	5	1	5
Q24:	3	5	6
Q25:	4	3	3
Q26:	6	4	4
Q27:	1	2	1
Q28:	2	6	2
Q29:	NO ANSWER	security as low priority because of academic environment	NO ANSWER
Q30:	No Response	No Response	No Response
Q31:	1	2	4
Q32:	4	1	1
Q33:	3	5	3
Q34:	2	4	5
Q35:	5	3	2
Q36:	NO ANSWER	NO ANSWER	NO ANSWER
Q37:	No Response	No Response	No Response
Q38:	intermediate	not significantly more than would be required for the participating machines and networking without the iROS layer	Monthly maintenance session <= 2 hours. Weekly check <= 10 minutes. Major version upgrades approximately every 6 months that take 2 hours.
Q39:	intermediate to high	iRoom: more than the above; keeping distributed installations of demos etc intact has not been accomplished even with a dedicated 50% admin	When problems, mostly re-boot. Need to do an iROS update. Have to alter config files when moving to other iSpaces.
Q40:	discussing construction projects	meetings, plus research environment behind the scenes	- Informal collaborative meetings for design projects. - Presentations for visitors.
Q41:	No Response	No Response	No Response
Q42:	Strongly agree	Nuetral	Nuetral
Q43:	NO ANSWER	i have not installed others	NO ANSWER
Q44:	Strongly agree	Strongly disagree	Nuetral
Q45:	NO ANSWER	i am referring to the problem of "admin/user-level debugging" - getting all room features to work when they don't. This does not refer to developer-level debugging.	NO ANSWER
Q46:	Strongly agree	Agree	Agree
Q47:	NO ANSWER	NO ANSWER	Need an upgrade that hopefully will deal with some bugs.
Q48:	No Response	Nuetral	Agree
Q49:	NO ANSWER	NO ANSWER	NO ANSWER
Q50:	Strongly agree	Agree	Strongly agree
Q51:	NO ANSWER	I am referring to iROS X for Mac OS X here	NO ANSWER

Continued on next page

Continued from previous page

	Respondent 1	Respondent 2	Respondent 3
Q52:	No Response	Agree	Strongly disagree
Q53:	NO ANSWER	NO ANSWER	NO ANSWER
Q54:	Better than expected	Better than expected	Acceptable
Q55:	NO ANSWER	eheap itself seems fast enough, but the many intermediate entities (X10 controllers, JVMs, ...) create latencies that if added up are often not acceptable for UI research.	NO ANSWER
Q56:	not sure	see 55. Prioritization of certain events would be useful, although this may have to be done using lower-level hardware routers etc.	- Able to switch iSpaces with client more easily. - Security - PointRight as part of Event Heap
Q57:	not sure	Fairly little in the iRoom I believe (eheap always works, MB & PR mostly do). iROS X has not been used extensively yet.	Minimal
Q58:	none	iLand/BEACH (too Smalltalk-specific)	N/A
Q59:	C++, .NET, VB , Java	Generally, open or cross-platform standards (Java, HTTP) In our projects, also Cocoa (and its Objective-C) under Mac OS X How about ZeroConf?	C++, HTTP
Q60:	No Response	No Response	No Response
Q61:	very robust architecture and new receivers and senders can be plugged in very easily	The original event queue model was a step in OS architecture that was mainly aiming at improving the support for interactive applications. It now appears as if iROS could play a similar role in post-desktop, ubicomp environments. Therefore, take user interface development needs (latency etc.) into account when further developing iROS.	NO ANSWER

Table 5 - Administrator Survey Responses

Compiled Results

The compiled results of the administrator survey answers as presented in the previous section can be found below. For each question percentage breakdowns are given for multiple choice style questions, and a list of all responses is given for text questions. The compiled results were created automatically by the SALGains web site [1].

1. [EXPLANATORY: DON'T ANSWER] The first section consists of questions about your background. (N=0)

2. In terms of my general knowledge, comfort, experience and ability with computers in general, I would consider myself: (N=3)

No experience	0% (0)
Novice	0% (0)
Amateur	0% (0)
Intermediate	33% (1)
Master	33% (1)
Expert	33% (1)

3. Comments

NO ANSWER

4. In terms of my general knowledge, comfort, experience and ability with interactive workspace type environments, I would consider myself: (N=3)

No experience	0% (0)
Novice	0% (0)
Amateur	0% (0)
Intermediate	33% (1)
Master	67% (2)
Expert	0% (0)

5. Comments

NO ANSWER

6. In terms of my general knowledge, comfort, experience and ability with administering iROS and the Event Heap, I would consider myself: (N=3)

No experience	0% (0)
Novice	33% (1)
Amateur	0% (0)
Intermediate	33% (1)
Master	33% (1)
Expert	0% (0)

7. Comments

NO ANSWER

8. Please give a brief description of the type of iROS based interactive workspace you have administered and the types of functionality being used in that space:

iLoft, Presentations, multibrowse, PointRight. Collaborative meetings, demonstrations.

iRoom, plus ad-hoc workspaces on Mac OS X laptops for demos and testing

9. [EXPLANATORY: DON'T ANSWER] The second section contains questions not specific to administration or the iROS/Event Heap system. (N=0)

10. ‘Applications’ for interactive workspaces can either be collections of small programs (ensembles) or monolithic applications which run across the whole space. Which of these two will be the more common case: (N=3)

Application Ensembles only	0% (0)
Application Ensembles and an occasional monolithic application	100% (3)
Both equally	0% (0)
Mostly monolithic applications, but some ensembles	0% (0)
Monolithic applications only	0% (0)
Don't know.	0% (0)

11. Comments:

NO ANSWER

12. Interactive workspaces are likely to be dynamic on short time scales due to application and device failure and the entrance and departure of portable devices: (N=3)

Strongly agree	33% (1)
Agree	67% (2)
Neutral	0% (0)
Disagree	0% (0)
Strongly disagree	0% (0)
Average = 1.67, S.D. = 0.47	

13. Comments:

NO ANSWER

14. Which is more tolerable: (N=3)

Occasional total system failures which require ten or more minutes to restart	33% (1)
More frequent faults which effect only a part of the system and require one or two minutes to reset that component	33% (1)
Not sure	33% (1)

15. Comments:

Needs a 3 minute restart max.

16. Would you accept reduced performance (up to 1 s latency) from an interactive workspaces system infrastructure to get guaranteed message delivery? (N=3)

Yes	67% (2)
No	33% (1)

17. Comments

Since our work is in the area of user interfaces for interactive workspaces, latencies of up to 1s would not be tolerable in any case.

18. Would you accept reduced performance (up to 1 s latency) from an interactive workspaces system infrastructure if the system guaranteed consistency (all components see the same events in the same order)? (N=3)

Yes	67% (2)
No	33% (1)

19. Comments

Since our work is in the area of user interfaces for interactive workspaces, latencies of up to 1s would not be tolerable in any case.

20. Would you accept reduced performance (up to 1 s latency) from an interactive workspaces system infrastructure if it supported transactions among connected components? (N=3)

Yes	67% (2)
No	33% (1)

21. Comments

Since our work is in the area of user interfaces for interactive workspaces, latencies of up to 1s would not be tolerable in any case.

22. [EXPLANATORY: DON'T ANSWER] Please rank the following six general features according to how important they are for an interactive workspace coordination infrastructure. Please assign a unique value for each feature, 1 being the most important: (N=0)

23. Support for heterogeneous devices and hardware platforms (N=3)

1	33% (1)
2	0% (0)
3	0% (0)
4	0% (0)
5	67% (2)
6	0% (0)

24. Support for a variety of development platforms (Java, C++, .NET, etc.) (N=3)

1	0% (0)
2	0% (0)
3	33% (1)
4	0% (0)
5	33% (1)
6	33% (1)

25. Ability to easily integrate legacy applications (COTS and In-house) (N=3)

1	0% (0)
2	0% (0)
3	67% (2)
4	33% (1)
5	0% (0)
6	0% (0)

26. Provision for new hardware and software that may be released in the future. (N=3)

1	0% (0)
2	0% (0)
3	0% (0)
4	67% (2)
5	0% (0)
6	33% (1)

27. Reliability and Stability (N=3)

1	67% (2)
2	33% (1)
3	0% (0)
4	0% (0)
5	0% (0)
6	0% (0)

28. Security (N=3)

1	0% (0)
2	67% (2)
3	0% (0)
4	0% (0)
5	0% (0)
6	33% (1)

29. Comments on the ranking of the features in the previous six questions:

security as low priority because of academic environment
--

30. [EXPLANATORY: DON'T ANSWER] Please rank the following five administration related features according importance for an interactive workspace coordination infrastructure. Please assign a unique value for each feature, 1 being the most important: (N=0)

31. Ease of administration (N=3)

1	33% (1)
2	33% (1)
3	0% (0)
4	33% (1)
5	0% (0)

32. Ability to bring in new applications and have them integrate with existing applications and devices in the interactive workspace (N=3)

1	67% (2)
2	0% (0)
3	0% (0)
4	33% (1)
5	0% (0)

33. Ability to monitor communications among applications in the workspace and view their contents in human readable form (N=3)

1	0% (0)
2	0% (0)
3	67% (2)
4	0% (0)
5	33% (1)

34. Ease of debugging (N=3)

1	0% (0)
2	33% (1)
3	0% (0)
4	33% (1)
5	33% (1)

35. Portability of applications written for one interactive workspace to new environments (N=3)

1	0% (0)
2	33% (1)
3	33% (1)
4	0% (0)
5	33% (1)

36. Comments on administrator features and their importance:

NO ANSWER

37. [EXPLANATORY: DON'T ANSWER] The following three questions are general questions on administering interactive workspaces: (N=0)

38. What level of administration is acceptable for an interactive workspace?

intermediate
Monthly maintenance session <= 2 hours. Weekly check <= 10 minutes. Major version upgrades approximately every 6 months that take 2 hours.
not significantly more than would be required for the participating machines and networking without the iROS layer

39. What level of administration has been necessary for your interactive workspace so far?

intermediate to high
iRoom: more than the above; keeping distributed installations of demos etc intact has not been accomplished even with a dedicated 50% admin
When problems, mostly re-boot. Need to do an iROS update. Have to alter config files when moving to other iSpaces.

40. What is or will be the primary usage of your interactive workspace?

- Informal collaborative meetings for design projects. - Presentations for visitors.
discussing construction projects
meetings, plus research environment behind the scenes

41. [EXPLANATORY: DON'T ANSWER] The following eleven questions are questions about administration of the Event Heap/iROS specifically: (N=0)

42. Compared to other setups of interactive workspaces I've worked with, ones based on the Event Heap/iROS are easy to maintain and administer: (N=3)

Strongly agree	33% (1)
Agree	0% (0)
Neutral	67% (2)
Disagree	0% (0)
Strongly disagree	0% (0)
Average = 2.33, S.D. = 0.94	

43. Comments:

i have not installed others

44. The Event Heap/iROS applications and functionality deployed in our interactive workspace are easy to debug: (N=3)

Strongly agree	33% (1)
Agree	0% (0)
Neutral	33% (1)
Disagree	0% (0)
Strongly disagree	33% (1)
Average = 3, S.D. = 1.63	

45. Comments:

i am referring to the problem of "admin/user-level debugging" - getting all room features to work when they don't. This does not refer to developer-level debugging.

46. The Event Heap/iROS is very stable and reliable: (N=3)

Strongly agree	33% (1)
Agree	67% (2)
Neutral	0% (0)
Disagree	0% (0)
Strongly disagree	0% (0)
Average = 1.67, S.D. = 0.47	

47. Comments:

Need an upgrade that hopefully will deal with some bugs.

48. The Event Heap is implemented as a server that runs on a single machine through which devices and applications communicate. Overall, this centralization has been an advantage: (N=2)

Strongly agree	0% (0)
Agree	50% (1)
Neutral	50% (1)
Disagree	0% (0)
Strongly disagree	0% (0)
Average = 2.5, S.D. = 0.5	

49. Comments:

NO ANSWER

50. Downloading the iROS/Event Heap system and getting it working was easy: (N=3)

Strongly agree	67% (2)
Agree	33% (1)
Neutral	0% (0)
Disagree	0% (0)
Strongly disagree	0% (0)
Average = 1.33, S.D. = 0.47	

51. Comments:

I am referring to iROS X for Mac OS X here

52. Currently the only security for iROS based interactive workspaces is to firewall the interactive workspace off from the rest of the network. This is an acceptable level of security for our interactive workspace: (N=2)

Strongly agree	0% (0)
Agree	50% (1)
Neutral	0% (0)
Disagree	0% (0)
Strongly disagree	50% (1)
Average = 3.5, S.D. = 1.5	

53. Comments:

NO ANSWER

54. How is the performance (bandwidth, throughput, latency) of the Event Heap for the applications in use in your interactive workspace? (N=3)

Unusable	0% (0)
Poor	0% (0)
Tolerable	0% (0)
Acceptable	33% (1)
Better than expected	67% (2)
Superior	0% (0)

55. Comments:

eheap itself seems fast enough, but the many intermediate entities (X10 controllers, JVMs,) create latencies that if added up are often not acceptable for UI research.

56. Are there any specific areas of Event Heap performance that you would like to see improved?

- Able to switch iSpaces with client more easily. - Security - PointRight as part of Event Heap
 not sure
 see 55. Prioritization of certain events would be useful, although this may have to be done using lower-level hardware routers etc.

57. What level of administration has been necessary for the iROS specific components of your interactive workspace so far?

Fairly little in the iRoom I believe (eheap always works, MB & PR mostly do). iROS X has not been used extensively yet.

Minimal

not sure

58. What other systems did you consider using besides iROS/the Event Heap, if any? What factors made you decide in favor of the Event Heap/iROS (e.g. easy to download)?

iLand/BEACH (too Smalltalk-specific)

N/A

none

59. What platforms are important for the Event Heap to support (current and desired)? (e.g. C++, .NET, Perl, HTTP, Visual Basic, Cocoa, etc.)

C++, .NET, VB , Java

C++, HTTP

Generally, open or cross-platform standards (Java, HTTP) In our projects, also Cocoa (and its Objective-C) under Mac OS X How about ZeroConf?

60. [EXPLANATORY: DON'T ANSWER] Concluding Section (N=0)

61. Any overall suggestions and/or comments?

The original event queue model was a step in OS architecture that was mainly aiming at improving the support for interactive applications. It now appears as if iROS could play a similar role in post-desktop, ubicomp environments. Therefore, take user interface development needs (latency etc.) into account when further developing iROS.

very robust architecture and new receivers and senders can be plugged in very easily

List of References

1. Seymour, E., *Student Assessment of Learning Gains*, . 1997, University of Wisconsin
<http://www.wcer.wisc.edu/salgains/instructor/default.asp>.