Citizen Science System Assemblages: Toward Greater Understanding of Technologies to Support Crowdsourced Science

Nathan Prestopnik
Syracuse University
napresto@syr.edu

Kevin Crowston
Syracuse University
crowston@syr.edu

ABSTRACT

We explore the nature of technologies to support citizen science, a method of inquiry that leverages the power of crowds to collect and analyze scientific data. We evaluate these technologies as system assemblages, collections of interrelated functionalities that support specific activities in pursuit of overall project goals. The notion of system assemblages helps us to explain how different citizen science platforms may be comprised of widely varying functionalities, yet still support relatively similar goals. Related concepts of build vs. buy, support for science vs. support for participants, and web satisfiers vs. web motivators are used to explore how different citizen science functionalities may lead to successful project outcomes. Four detailed case studies of current citizen science projects encompassing a cross-section of varying project sizes, resource levels, technologies, and approaches to inquiry help us to answer the following research questions: 1) What factors influence the composition of a system assemblage for citizen science? 2) What do typical system assemblages for citizen science look like? 3) What effect does the assemblage composition have on scientific goals, participant support, motivation, and satisfaction? and 4) What are the design implications for the system assemblage perspective on citizen science technologies?

Author Keywords

Citizen Science, System Assemblages, Socially Intelligent Computing, Web Technology.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Citizen Science is a term used to describe research collaborations that enable non-scientist members of the public to assist with scientific investigations [1, 2]. Sometimes described as crowdsourced science, citizen science projects include (for example) those that ask participants to classify photographs of galaxies taken by space telescopes, report bird sighting data for ornithological research, or plant sunflowers and observe bee pollination at various times during the day. Additional examples abound.

While their scientific areas of inquiry and research questions vary widely, these projects increasingly have a common approach to data collection: large numbers of individuals are asked to submit observations or analyses via web- or mobile-based technologies. Despite this similarity, at the level of specific functionalities employed, citizen science projects vary widely. Some projects utilize off-the-shelf software while others hire contract or in-house developers to build custom platforms. Many include highly motivational elements such as games, social media, and impactful data visualizations. Supporting scientific goals is critical for virtually all citizen science projects, but some projects do this better than others, and some fall short in various ways.

When designing a citizen science project, the *system assemblage*, a collection of interrelated components that support various functionalities, is the ultimate goal. The assemblage may be loose, as when a variety of off the shelf software systems are used separately to support various project goals, or tight, as in a project website that presents all functionalities in a unified way. In any of its many potential variations, the citizen science system assemblage will support overall project goals via discrete functional components: data collection tools, visualization components, blogs, news feeds, email, forums, project information, etc. In addition, these components will support a social element, as scientists and participants cooperate toward scientific outcomes. There is no ideal assemblage that projects should aspire to, as a multitude of parameters make each assemblage unique. Nonetheless, achieving a smoothly functioning system assemblage is a key milestone for any successful citizen science project.

Over the course of system assemblage development, projects encounter a number of key challenges. The build vs. buy decision to either develop assemblage components for oneself or buy/acquire them is an important moment in the development of the assemblage. It is a decision that has repercussions for many other aspects of a citizen science project, including two additional challenges: the need to balance scientific goals against the needs of participants, and the need to motivate participation and ensure participant satisfaction.

We explore the system assemblages of four citizen science projects through a comparative case study approach. For each case study, we evaluate the assemblage in terms of its overall composition, build vs. buy decision-making, and its impact on scientific goals, participant support, motivation, and satisfaction. We address the following research questions: 1) What factors influence the composition of a system assemblage for citizen science? 2) What do typical system assemblages for citizen science look like? 3) What effect does the assemblage composition have on scientific goals, participant support, motivation, and satisfaction? and 4) What are the design implications for the system assemblage perspective on citizen science technologies?

BACKGROUND

No one functionality or technology can adequately support every one of a citizen science project's goals. Citizen science project developers ultimately mix and match, choosing different functionalities and technologies for different purposes. Furthermore, though project goals are often similar across citizen science projects, the specific functionalities and technologies that support them are often chosen for pragmatic reasons rather than scientific ones. To analyze this situation, we draw on the notion of a system assemblage [3-5].

System Assemblages

A basic feature of our analysis is a consideration of the technology support for a citizen science project as an as assemblage of different loosely coupled systems rather than a single system. We draw from Kling's [3-5] approach to computer technology evaluation that he calls "web models." Web models "conceive of a computer system as an assemblage of equipment, applications, and techniques with identifiable information-processing capabilities [4]." This approach is distinguished from "engineering models," or "discrete entity" analyses, which emphasize equipment and information processing capabilities only.

Most citizen science projects leverage multiple technologies to support multiple functionalities; rather than deploying single, self-contained systems, citizen science projects are more likely to choose from a "menu" of functionalities. One project may opt to include a web forum to support participant collaboration, while another may view this functionality as unnecessary. Because of the social nature of citizen science, many of the functionalities that make up a citizen science assemblage have some communication role: a forum enhances communication between participants and project scientists, content pages provide information or standardize training, blogs and news feeds reach out to participants, contact forms allow participants to address members of the project team. Even data collection utilities give participants a standardized avenue for communicating the work they have completed. Because most individuals' first encounter with a citizen science project is through its online presence, the citizen science assemblage also has a key role to play in motivating participation.

One issue in developing a citizen science assemblage is the build vs. buy decision. Some projects take a "build" mindset, developing most of their system components in-house, thereby retaining a great deal of control over component functionality and the ability to integrate components. This degree of control usually comes at the cost of increased development time and more money spent on professional design expertise. Forgoing this expertise may save on expenses, but can cost a project in other ways. For example, projects developed by volunteer, non-expert labor may, depending on the capabilities of the volunteer, have more limited or less reliable data collection instruments. Similarly, they may suffer through ugly, ineffective, or unusable design which can have an effect on participant motivation and satisfaction. Tailored approaches to development may also require that even professional system developers spend a large portion of build time on features other than data collection; basic add, edit, and delete functionality for information pages, as well as participant management features that must often be built from scratch.

A sizable minority of citizen science projects instead seek to leverage technologies developed by others, either by purchasing them or by finding free or open-source solutions. This "buy" mentality usually results in inexpensive and high quality support for some assemblage components, particularly those common to many web applications like participant management, content management, or site navigation. Much basic web functionality comes prepackaged with bought systems, so developers can dedicate their time to creating data collection instruments and other novel features, rather than recreating industry-standard functionality like participant account or content management tools. However, bought systems sometimes place restrictions on the augmenting features that may be designed into them. Some systems are not open technology, making additions very difficult or even impossible; more often, such systems are either too complex or too limited in their capabilities, increasing the difficulty of appending advanced functionality onto them or seamlessly integrating the various assemblage components.

The build vs. buy decision shapes the final character of a system assemblage, particularly as it relates to the support of scientific goals vs. the support of participants. Built systems tend to allow a level of flexibility and integration that is most useful when applied to the science support components of the assemblage (data collection or analysis interfaces in particular). Bought systems tend to emphasize conformity to existing standards, and often come standard with functionality such as managing user accounts, maintaining web content, and communicating with participants. Such functionality is important to the project, in that it supports participants and their expectations, but can be too rigid to easily support science goals "out of the box." The build vs. buy and supporting science vs. supporting participants dichotomies have further implications with regard to participant satisfaction and motivation as well.

Satisfiers vs. Motivators

Because citizen science efforts are predicated on the contributions of a large number of participants, motivation is a core issue for most citizen science efforts. If too few participants are motivated to join a citizen science project or if too many lose interest, the project will fail. Motivation to participate is multi-faceted: participants join citizen science projects for a variety of reasons, including their interest in the subject of scientific inquiry, the relevance of data collection efforts to particular interests or hobbies, how inherently fun a project is, or for altruistic reasons.

Zhang and von Dran [6] have argued that website features can be grouped into two distinct categories: *satisfiers* and *motivators*. Satisfying features are those whose absence will cause a participant to experience dissatisfaction with a website while motivating features are those which add motivational value to a website. Participants expect websites to maintain a certain level of quality, and substantial reduction of that quality will cause dissatisfaction. At the same time, the motivation to use and revisit a website comes from a potentially different set of features; in other words, "being usable and being likable are two different goals," [6].

In their research, Zhang and von Dran [6] identified a variety of common web features and clustered them into satisfier and motivator categories. Satisfiers include participant account and security features, basic usability, cross-browser compatibility, up-to-date information, and the quality of informational content. Motivating features include cognitive outcomes such as learning new knowledge or skills, the fun of exploring a site, the presence of multimedia, level of challenge, the ability to control interaction, and the presence of eye-catching visuals. Zhang and von Dran's [6] research did not specifically look at the citizen science context. However, the concepts of satisfier and motivator should be relevant to citizen science, as good usability and highly motivating design are important to scientific and participant-oriented outcomes. Citizen science websites usually address highly specific scientific interests, so these projects must convince average citizens that site content and activities are interesting, enjoyable, and easy to use; failure to attract or retain participants will also result in the failure of ultimate scientific goals. This is to say that citizen science projects must meet a high bar in terms of both satisfiers and motivational features.

The build mindset, which allows for flexible and unique data collection interfaces to support science goals, often results in more unique or custom functionalities that emphasize motivational elements over satisfiers. This can include highly visual interfaces, games or game-like features, high scores, participant statistics, and data visualizations. Bought components, such as content management systems (CMS), on the other hand, tend to emphasize standard functionalities like modifying page content, or managing users. These functionalities fall well into the satisfier category, and off the shelf software typically excels at supporting them. With either the build or buy mindset, it is possible to develop a highly motivating *and* highly satisfying project; for those who primarily build, however, extra attention must be paid to satisfying elements such as usability, organization, and content, while for those who primarily buy, extra attention must be paid to motivators like making a site fun, including multimedia, and visual design.

CASE STUDIES

We used a case study approach to explore the notion of system assemblages, buy vs. build, support for science vs. support for participants, and satisfiers vs. motivators in the citizen science context, first undertaking a heuristic analysis of citizen science projects to identify common features and functionalities, and then further evaluating four of these projects through in-depth reviews of their technological implementations and interviews with their science and development staff.

Case study projects included *eBird*, *Galaxy Zoo*, the *Great Sunflower Project*, and *What's Invasive*, citizen science undertakings in the scientific domains of ornithology, astronomy, apiology, and phenology respectively. The chosen case sites span a range of technological sophistication and development approaches. Data for these case studies was triangulated from three distinct sources of data: interviews, project documentation, and website analysis.

Scientists and system developers were interviewed about project histories, scientific and outreach goals, supporting technologies, and the citizen science phenomenon as it relates to their particular research efforts. Ten separate interviews of nine scientists and system developers were conducted in total. Interviews lasted for approximately one hour each. In some cases, follow-up email questions were sent and responses to them were collected.

In most cases, interview participants from the four projects were willing to provide additional documentation in the form of email correspondence, technical documentation, planning documentation, web traffic information, and participation data. These various sources of information were reviewed, and key data points were extracted for inclusion in the case studies.

Finally, the websites (and, in the case of the *What's Invasive* project, mobile applications) for the four projects were scrutinized in detail for the specific functionalities that they include, as well as information about the technologies and design strategies utilized in their development.

Additionally, the researchers conducted a review of website functionalities on a cross-section of citizen science project websites. The 27 websites included in this review were from the sample developed by Wiggins and Crowston [2] using the landscape sampling method described in [7]. A list of key functionalities was iteratively generated by visiting each of the 27 citizen science websites and comprehensively evaluating their publically available content. The resulting list of 33 functionalities included items like "project information," "forums," "registration forms," "submit text data," blogs," etc. Websites were flagged as either containing or lacking these features, and the resulting data set was stored in a database developed for this research.

These three sources of primary data: interviews, project documentation, and website feature reviews were used to compile four detailed case study write-ups. Case studies were used to make comparisons and draw conclusions about system assemblages, motivators, and satisfiers in citizen science development.

CASE 1: EBIRD

eBird is joint project of the Cornell Lab of Ornithology and National Audubon Society. It is a citizen science project that engages a global network of birdwatchers to submit their observations via web-based tools. The service is free and serves a primary role for gathering large quantities (2 million observations per month) of scientific and conservation-oriented bird data, which are used by a broad spectrum of research and conservation initiatives.

System Assemblages in *eBird*

The system assemblage perspective can be applied to *eBird* at a variety of levels. For example, several projects are supported by *eBird's* Oracle database, forming a suite of related projects that can be thought of as one large system assemblage. These projects share some common features at the data level (for example, they all orient around bird observation), but are otherwise discrete. Each *Lab of Ornithology* project is also an individual system assemblage that supports varying features, functionality, and participants depending on its overall scientific goals and the birding activities that should be supported.

The current instantiation of *eBird* is a hybrid of the build vs. buy decision, with some content portions of the project website running off a CMS system, while data collection interfaces and the database itself are built in-house and are highly customized. The hybrid approach has helped *eBird* to address the various challenges identified through this research. For example, using off the shelf CMS software allows for easy management of certain basic content, freeing developer time for more complex challenges in database design, data collection, and providing feedback to participants. In general, *eBird* has adopted the strategy that basic functionality can be off the shelf, while more advanced tools should be custom. There is a trend away from off the shelf software on this project, however, and soon *eBird* may be an entirely custom system assemblage.

eBird supports its scientific goals through tailored applications that allow participants to submit locations and bird observations. These tools also give participants the ability to view their prior submissions, maintain "life lists" of birds, and organize their work in various ways. In this sense, the custom development on *eBird* also supports a variety of participant-oriented functions, which makes sense given that *eBird's* development style is weighted more heavily toward a build mentality than a buy one. However, the CMS system is targeted at basic participant functionality, such as communicating with participants and organizing various site content.

Motivation vs. Satisfaction in *eBird*

eBird leverages an existing hobby—birding—to capture and retain a pool of willing and motivated participants. Thus, an unusually high base level of motivation is already present among participants of the system. This is an infrequent occurrence within the citizen science domain, where data collection tasks are not always tightly bound to inherently fun activities. Many citizen science projects attempt to turn mundane tasks into fun activities through games or other mechanisms. *eBird* takes the opposite stance, using an already fun activity that is enjoyed by millions to generate large amounts of scientific data. Capitalizing on this built-in motivational effect, the *eBird* website uses custom interfaces to provide participants with exactly the kinds of tools they desire for viewing, visualizing, and exploring the data that they and others have submitted. *eBird* is a highly motivating web experience for birders because custom development has made it their ideal. At the same time, the *eBird* designers have also emphasized satisfaction oriented features such as informative content, adequate privacy management, and well-designed functionality that works properly and as expected. This need for participant satisfaction has been emphasized both in their own custom design work and in the CMS system they use to augment it.

One risk of using a hobby or special interest population for data collection is that support for science goals may become conflated with support for unique participant requirements. The data collected by *eBird* addresses interesting scientific problems for the ornithology community, but this goal must always be balanced against the need to support birding activities through continued tool improvement, system maintenance, and periodic technology or design refreshes, lest participant desire to use *eBird* begin to wane. In the case of *eBird*, the relationship between science and participation is often mutually beneficial, as better tools for birders often equate to better tools for scientific data collection. This may not be the case on all citizen science projects, however. It is important that a citizen science assemblage directed at a hobbyist population does not become too heavily weighted toward supporting the hobby and the participants over scientific investigation.

CASE 2: GALAXY ZOO

Galaxy Zoo is a citizen science project that has individuals aid with scientific inquiry through the online annotation of the millions of astronomical photographs collected by the Hubble Space Telescope, the Sloan Digital Sky Survey, and others. *Galaxy Zoo* asks individuals to identify various galaxy features (the number of spiral arms, how round or elliptical they are, etc.) in these photographs, and this information is used to paint a more detailed picture of the universe we live in. *Galaxy Zoo* is a project within the *Zooniverse*, a collection of citizen science projects maintained and developed by the *Citizen Science Alliance*.

System Assemblages in *Galaxy Zoo*

Most *Zooniverse* projects are annotation based; instead of having participants go into the field to collect and report observations, these projects have participants work with a computer interface to generate metadata (annotations) about assets (photographs of galaxies, scans of old ship's logs, photos of moon craters, etc.).

The *Zooniverse* system assemblages are more integrated than most other citizen science efforts. They are built, rather than leveraging off the shelf software, and their designers spend a great deal of time on the design process. The build decision is directly related to support for science. According to one developer, "We had pretty grand designs when I joined, and we still do.... *Galaxy Zoo 2* was the first of a new breed for us. We're now eight or nine projects down since then. I think going the bespoke route allowed us to really be pretty opinioned about what we thought was the right approach.... Trying to run a *Drupal* CMS and just kind of hacking in some code to make a decision tree appear on the page just wouldn't have felt right. It wasn't grand enough for our plans."

Galaxy Zoo is designed around participants annotating assets, so specialized interface functionality is often required to present the participant with an asset and capture the resulting metadata. *Galaxy Zoo* uses a visual decision tree format to elicit information from participants. Other *Zooniverse* projects use different approaches, for example, the project *Moon Zoo* has participants draw outlines around moon craters, while the project *Old Weather* asks participants to highlight sections of a scanned ship's log and type in the information it contains. Each of these interfaces is unique, though other elements are more similar across projects. The tailored approach to development allows *Zooniverse* system assemblages to feel completely integrated at the website level, even as some features are used on multiple websites while others are deployed only in one place. This approach does require that the *Zooniverse* design team spend extra effort on basic usability, management of content, and management of users, but this tradeoff has been managed in a unique way.

Traditional programming practice dictates that code be reused as much as possible, to save time on new development. For *Zooniverse* projects, an extensive code library was developed during the *Galaxy Zoo* implementation. However, the code library became a source of tradeoffs very similar to those found in the build vs. buy decision. In essence the code library became a "bought" system that was complex and sometimes time consuming for newly hired programmers to use effectively. It offered a great deal of functionality, but sometimes more than was needed. Interestingly, it was difficult for newer developers to gain a sense of ownership over their work when the code they were working with came largely from this library.

The *Zooniverse* design team adopted a new development paradigm, where previous code is retained or centralized for identical or highly similar tasks (i.e. satisfier functionalities such as participant management), but each *Zooniverse* site is thereafter approached like an independent design project. This efficient approach to developing a system assemblage allows developers to direct their time toward developing motivating functionalities that support scientific goals, while simultaneously ensuring that previously developed and successful functionalities with a satisfier role are effectively reused.

Motivation vs. Satisfaction in *Galaxy Zoo*

One important motivational element identified in the *Galaxy Zoo* case study is the notion of "rules for ethical citizen science." These rules are simple, suggesting that citizen science participants 1) be treated as collaborators, 2) not have their time wasted, and 3) not be asked to undertake tasks that could be better undertaken by a computer. These rules have influenced many of the components included on *Zooniverse* system assemblages, including tools for participant collaboration, web sections to release scientific data and papers, and acknowledgement or the assignment of co-authorship on various academic papers. In addition, these rules impact the kinds of data that are collected and the custom interfaces by which they are submitted.

The rules for ethical citizen science stem from a desire for scientific outreach and education. *Galaxy Zoo* and the other *Zooniverse* projects have a core desire to teach participants what it is to do science. This outreach effort emphasizes scientific process—the scientific method and how scientific inquiry, regardless of topic, is conducted. Such outreach has a motivational element—the ability to learn new things—as evidenced by *Galaxy Zoo* participants who have undertaken independent scientific inquiry outside the aegis of specific *Galaxy Zoo* research questions. In one example, participants identified unique astronomical objects of interest in the *Galaxy* Zoo photos (the "Green Peas," so called for their vibrant green color) and started a data gathering and analysis project of their own design to identify what they might be.

Providing feedback to participants and encouraging them to collaborate can be important ways to foster a sense of community and to encourage continued participation. Feedback in the form of credit for work completed, acknowledgement in published research, praise for continued participation, and involvement in more challenging modes of inquiry are viewed as especially motivating by *Zooniverse* scientists. Most of these motivating features are strongly tied to scientific goals and have been addressed through custom systems.

Galaxy Zoo and the *Zooniverse* projects also do well with satisfier functionality; "simplicity" is a mantra for the *Zooniverse* design team, and their work is positively impacted by this viewpoint. By favoring the build side of the build vs. buy equation, *Zooniverse* developers provide themselves with as much control as possible over both motivation and satisfier elements.

CASE 3: THE GREAT SUNFLOWER PROJECT

The *Great Sunflower Project* is a citizen science project to collect information on bee pollination service throughout the United States. Participants in this project use a *Drupal*-based website to submit information about sunflower gardens that they have established, as well as the number of times bees visit sunflowers within a given a period of time. Aggregated, this data shows where bees are providing good pollination service, as well as places where pollination may be suffering. The ultimate goal of the *Great Sunflower Project* is to explore how ecosystem services like pollination are affected by environmental factors.

System Assemblages in the *Great Sunflower Project*

Limited project resources were highly influential in the build vs. buy decision for the *Great Sunflower Project* system assemblage. While other case study projects in this research had the resources to attempt custom development, the *Great Sunflower Project* adopted a buy mentality and used the open source CMS *Drupal* to deploy a project at much lower cost. The developer for the *Great Sunflower Project* was highly supportive of open-source, CMS-based development generally, saying, "When you embrace a project like a content management system, you get the benefit of years of different people working on different sites, and all the experience behind that built into your project. Frankly, I would not embark upon a project like this building it from scratch."

Building highly visual data interfaces (e.g. custom development with a motivational effect) was not possible on a tight budget, so *Great Sunflower Project* data collection takes place through simpler HTML forms programmed by its developer. All visual design on this project was kept as simple as possible because of the expense of custom design work. Furthermore, the *Great Sunflower Project* has also deployed only those functionalities that are required to support project goals; resources and personnel are not available to support a wide variety of functionalities that may not provide much additional value to the project. This has had the effect of making technology-based support for *Great Sunflower Project* participants both inexpensive and easy to deploy. The *Drupal* CMS handles user account management, website content, and a host of other support tasks. A free forum is used to manage participant communication. Email and web comments are also used for some communication tasks. The *Great Sunflower Project* incurred most of its technology-related expenses in the support of science goals, namely in hiring a programmer to develop the HTML forms used to collect data, where more custom development was required.

Motivation vs. Satisfaction in the *Great Sunflower Project*

Despite lacking very many motivational elements, the *Great Sunflower Project* case teaches that designing for participant satisfaction is deeply worthwhile. This is especially so in the citizen science domain, where improved usability can sometimes lead to effects that may otherwise result from motivational elements: increased participation, better participant retention, and better data validity. For example, the *Great Sunflower Project's* programmer and its principle investigator concentrated on usability issues surrounding data collection forms and noted specific improvements in the quality of submitted data when those issues were resolved. In addition, the participant population for this project is predominantly in an older and less web experienced age demographic, so good usability ensures that participants can find the information they need and easily navigate the project website. This has direct implications for participation in the project, which is high (approximately 80,000 registered users) despite aforementioned resource limitations.

Designing easy-to-use interfaces even for simple tasks can be more challenging than it seems; principle investigators on citizen science projects most often come from the natural sciences and rarely have expertise in information architecture. They are often unsure how to create the simplest, easiest, and best experience for their participants. This places project staff in the awkward position of rapidly learning a discipline's worth of design knowledge that typically falls well outside their field of expertise. Learning curves can be steep, and avoidable mistakes are often made. This is particularly problematic since in the citizen science domain, if usability is not at its best, data reliability may be called into question and participation may suffer.

Adopting a buy over build mindset helps to address these challenges, as off the shelf software ensures at least a minimum level of usability. For example, CMS systems and simple HTML forms address participant satisfaction because many of the major problems in these kinds of interface have been addressed through product development cycles and standardized web form specifications issued by the W3C. The *Great Sunflower Project* has leveraged such forms into a highly successful and highly cost effective approach to data collection.

CASE 4: WHAT'S INVASIVE

What's Invasive is a citizen science project developed by the *Center for Embedded Network Sensing* (CENS) at UCLA to collect information about invasive plant, animal, and insect species found within various locations throughout the United States. Participants in this project use mobile devices in the field to collect and submit information about the species they are observing, particularly their geographic location.

System Assemblages in *What's Invasive*

Because *What's Invasive* uses mobile technology, as a system assemblage it is different from others explored in this research. Mobile applications are, by their nature, more tightly integrated than many web applications, so at the application level, *What's Invasive* seems to be an exception to system assemblage perspective. However, the *What's Invasive* project also features a website to introduce people to the project and allow visitors to view and visualize various data. In addition, data from the mobile application is submitted to a database which feeds back to mobile devices and to the website. So at the project level, *What's Invasive* is a system assemblage like other citizen science efforts, with one component of the assemblage being a mobile application designed for data collection. This assemblage was developed in the build tradition, and the website, database, and mobile component are all built in a custom manner.

The *What's Invasive* case study highlighted how including a mobile component in a citizen science assemblage can be highly beneficial for scientific goals but may also place certain limitations on data collection. Mobile devices can be more difficult to use for supporting participants since all such management functionality must be custom developed into the mobile environment which, itself, must accommodate a variety of limitations (see below).

Citizen science projects benefit from the ability to bring a multi-purpose digital device into the field to collect GPS data, time data, photos, or video automatically. However, factors such as network connectivity, battery life, GPS limitations, and environmental conditions require much planning and preparation ahead of time. To function within an assemblage, mobile devices must be able to communicate effectively with databases and various other technologies despite these limits. Finally, because adverse environmental conditions such as wet weather, cold, or heat can impact performance or damage a mobile device, it is important to consider whether the scientific needs of a project may expose an individual's personal mobile device to damage, thereby inadvertently creating obstacles to participation.

Motivation vs. Satisfaction in *What's Invasive*

The *What's Invasive* development team has directed more attention toward motivational elements than satisfying ones, in large part because of the emphasis on mobile technology in this project. Mobile technologies are still relatively novel, and therefore have the potential to be highly motivating. However, simply releasing a mobile application is probably not enough to motivate participants by itself. One early project developed by CENS, *HAB Watch* (intended to monitor harmful algal blooms), showed how an application that is well designed from a technical and scientific standpoint may still fall short if functionality to attract and retain participants is not included. *HAB Watch* failed to attract much of a following because it concentrated entirely on data collection, but did not give participants other reasons for downloading or using the app. Vehicle parking information or news, for example, might have made this app more useful. Similarly a game or activity could also have increased participation. The *What's Invasive* mobile application has a much broader base of support and has not run into such serious difficulties, but motivating participation remains an important goal. Currently *What's Invasive* is undergoing updates to include game-like elements, scores, stats, and social activities for participants in an effort to enhance motivation and improve recruitment and retention of participants. In addition, both the mobile application and the website are undergoing updates to improve usability and other satisfier elements.

Games may be an especially powerful way to successfully flip the *eBird* approach to citizen science of capitalizing on an existing activity. For projects with no hobbies to leverage, games can produce the same sense of fun and enjoyment as "real world" activities, while still successfully linking participants to data collection efforts. For example, *What's Invasive* doesn't have any inherent interest group associated with it other than the relatively small number of people who are genuinely enthusiastic about invasive species reduction and park ecology. However, a large pool of potential participants for *What's Invasive* may be found among park-goers who are interested in hiking or walking, but do not have any particular interest in or awareness of ecology, biology, or park preservation. Mobile platforms are ideal for so-called pervasive gaming, where the game is infused into the physical environment. Such a game, if tied to a citizen science project like *What's Invasive*, might successfully wed outdoor enthusiasts to crowdsourced efforts toward invasive plant and insect species removal. This would directly stimulate science goals by increasing the amount (and, potentially, quality) of data collected.

DISCUSSION

Factors Influencing the System Assemblage

The build vs. buy paradigm has been discussed at length as one key influence on the composition of a system assemblage for citizen science. This decision is arrived at for a variety of reasons, including perceptions of how well off the shelf or custom software will support scientific and participatory roles.

Additionally, project resources and practical concerns often dictate the final shape of the system assemblage. The principle investigator for the *Great Sunflower Project* related that the web developer for the project, a professional programmer in the *Drupal* community, is a family relation; thus his services were accessible and obtained at a reasonable price. The developer himself indicated that his preference is for open platforms like *Drupal*. However, *Drupal* was specifically adopted not because he "evangelizes" for it in particular, but because it is the system that he is most familiar with. The *Great Sunflower Project's* path from scientific questions to technology implementation is similar to many successful citizen science deployments, in which the technologies of the system assemblage are chosen for largely pragmatic or convenience-based reasons, while functionality is purposed around broader project goals.

Galaxy Zoo offers a second example, where project developers explained that when they approached about designing new sites for other *Zooniverse* projects, they make a pragmatic decision about whether a project will be "done on the cheap" or "be a full project." This determination impacts the technologies that will be used, even as project goals may already be defined.

What's Invasive offers a third example: in this and other projects developed by CENS, a balance must be achieved between the computer science students who seek undergraduate development projects or graduate thesis topics, and project scientists, who seek to collect environmental data using mobile technologies. Science and technology goals are not always complementary, so a fair amount of pragmatism is introduced into the project selection and development process to ensure that computer science and environmental science goals are being met.

Composition of System Assemblages

It is difficult to envision what a "typical" system assemblage for citizen science would look like because of the varying details of these projects, but it is possible to characterize the most common components of these assemblages. From our review of 27 citizen science assemblages we identified a relatively stable set of common features and organized them into a continuum from virtually ubiquitous to highly optional. The following table shows this feature set, organized by overall count within the 27 reviewed projects:

Citizen Science Features	Count
Submit Data	27
Instructions	26
Project Information	26
Contact Information	23
Scientific Information	21
Registration	20
Collect Participant Information	18
Education	18
Project Data	17
Affiliates and Sponsors	16
Team/Staff Information	16
FAQ	15
Links	13
Sign In	13
News Feed	12
Blogs	11
Alerts	9
Forum	9
Donate	8
Practice and Testing	7
Email List	5
Participant Scores and Stats	5
Photo/Image Gallery	5
Published Papers	5
Calendar	2
Customized Participant Experience	2
Page Translation	2
Sales/Store	2
Contests	1

Table 1. Citizen science features.

This list of features suggests that, while motivational functionalities are often foremost in project scientist's minds during the design of a system assemblage, more mundane satisfiers such as task instructions, project information, or contact information are, in fact, equally or more common once a project hits the real world. The preponderance of such functionalities on this list suggests that project scientists and developers eventually do recognize the importance of these satisfier elements, even if they are not thought about as much as motivational features during early phases of design.

Effects of the Assemblage Composition

Data collection to support science goals typically requires highly specialized (and often highly complex) interfaces, along with relatively customized underlying data relationships. Illustrating the variability between systems, *Galaxy Zoo* features an icon-based question and answer system, *eBird* and the *Great Sunflower Project* employ a series of web forms for submitting birding data and bee counts respectively, and *What's Invasive* uses a mobile application for data collection. Other projects include drawing interfaces, transcription interfaces, or games. Wide variability in data collection approaches and technologies suggests that a tailored development approach may be appropriate if scientific goals—especially unique goals that cannot be supported by prepackaged technologies—are seen as preeminent.

Interview participants in this research also suggested that elements such as visualization technologies, games, and a high level of participant involvement with science and scientific outcomes could be highly motivating features within the citizen science context. Games and visualization technologies, in particular, are most likely to be successfully supported by tailored assemblages. For example, one of *eBird's* developers suggests (with regard to visualization technologies), "My biggest concern is that as the [visualization] technology matures, participants' expectations increase. Just being able to meet people's expectations, and to keep... something that they are going to want to go to, keep going back to, that's what my biggest concern is." One of the developers for *What's Invasive* further suggested that people who participate in citizen science projects want to know how their data is being used, and that updating the website to with information or visualizations about the submitted data would be especially valuable. Visualization and game technologies are complex, often requiring more design effort and more integration into a system assemblage than can be achieved through off the shelf software alone.

Assemblages composed predominantly of bought systems and based on a CMS, on the other hand, may be particularly suitable for projects that require an emphasis on participant and information management or those which have more limited requirements for data collection and other motivating features. CMS systems are specifically designed to support such functionalities, and projects with a information dissemination and retention focus will find that CMS systems often contain much of the desired feature set, especially satisfier functionalities that may impact participant retention.

Because science goals, motivation, and participation are important to all citizen science projects, whatever form the final system assemblage takes, appropriate attention should be paid to each of these elements. In an assemblage whose composition favors science support, more work may be required to support participant management tasks. In a one that favors participant support, extra attention should be directed toward science oriented functionalities.

Many citizen science developers and scientists appear to devote more of their attention to motivators than they do to satisfiers. So, for example, a number of projects feature innovative and visually appealing data collection interfaces, but are flawed in basic ways such as in their usability or site architecture. Others do, in fact, do well with satisfier functionalities, but have clearly placed most emphasis and effort on motivating features rather than satisfying ones.

This may be the result of the relatively limited web design knowledge held by many project scientists. One experienced *Galaxy Zoo* developer made the interesting point, when discussing project planning for new citizen science websites, that, "People typically think far more conservatively than you might imagine." This developer went on to explain that project scientists, with limited knowledge of the web's potential and its limitations, typically make one of two mistakes: they either assume that relatively achievable interactions with participants will be impossibly difficult, or they plan on complex or time consuming interactions that are not very feasible. It is rare for project scientists to truly understand what the technology will support and what it will not.

Possibly because of their interest in data and their limited knowledge of web design, most scientists emphasize data collection interfaces, which lend themselves to tailored, motivation-centered development approaches. However, considering only motivational features, which may be challenging or costly to develop, can be discouraging to scientists who wish to use a citizen science approach but believe it to be unrealistically difficult or expensive. This is unfortunate, because basic functionalities that satisfy participants are often equally or more prominent in the final assemblage, and designing an assemblage to support participant satisfaction is often easier because so many pre-built software packages support this kind of design. The importance and easily deployable nature of many satisfier features often goes unrecognized by project scientists as they begin planning citizen science efforts.

Design Implications

Citizen science projects have scientific goals which must be supported through technology. At the same time, these goals can become subordinate to a variety of factors, including project resources and the need for participants. Resource limitations force practical considerations to the forefront: to learn X, we must implement Y which will cost Z. Finding ways to implement Y at the lowest possible cost may involve searching for open-source or free software, using volunteer effort, or even revising scientific or project goals. It can be easy to assume that a cheaper component that is similar to Y may also do an adequate job of supporting research question X, but this is not always a safe assumption. Limited resources should be balanced against scientific goals to ensure that the science mission of a citizen science assemblage is not being compromised by tradeoffs in quality over cost.

Scientific outcomes are also impacted by participation. For science goals to be achieved, participants must be attracted to a project, motivated to involve themselves with it, and successfully managed so that their time is well spent. However, motivating, attracting, and retaining participants is a much larger design job than building a simple interface to collect data. Some functionalities support motivation, while others best support satisfaction. Planning an assemblage that can support both is highly important. Project scientists and developers should consider the full scope of a system assemblage to support citizen science, rather than concentrating only on its data collection components. Most of the project members we spoke with indicated that recruitment and managing participants occupies a much greater portion of their time than they had expected. Most had either already implemented or were seeking ways to implement various technologies to reduce the time spent on participant management duties or to more effectively attract participants to their project.

It is also important for project scientists and developers to acknowledge the importance of motivational and satisfier elements in a citizen science system assemblage. In the citizen science domain, motivating participation is valued; scientists who manage citizen science projects are very interested in understanding more about motivation and participation. However, there is much less overt enthusiasm for satisfier elements such as good usability, good organization, adequate privacy controls, or responsive communication to participants. These factors are recognized as necessary, but are sometimes undervalued in comparison to more motivational elements. Because citizen science projects are system assemblages, however, emphasizing satisfiers is at least as important as emphasizing motivation. A poorly integrated assemblage that is difficult to use or has technical problems will likely fail no matter how motivating certain of its components are. Attracting participants to a project, only to have them grow unhappy and leave, provides little benefit to the project or the participants and does nothing to serve science goals. We suggest that satisfier elements should be considered as early as possible, especially on projects with limited resources, as they constitute a healthy foundation upon which motivational components can eventually flourish.

Finally, consideration of the assemblage as a whole is important. Many project scientists and developers, especially those combining various bought or off the shelf systems, think deeply about specific functionalities, but miss the bigger picture of assemblage integration. Often integration is dictated by what specific technologies—a CMS, a visualization module, a data collection form—will allow. Not all technologies selected for inclusion in an assemblage come with serious technological constraints, however. Thinking early and often about how the components of an assemblage should work together will have positive effects on how easy the assemblage is to use and how well it satisfies and retains participants.

CONCLUSION

The system assemblage view of citizen science projects suggests that rather than understanding technological instantiations as seamlessly integrated units, it makes more sense to think of them as collections of discrete functionalities assembled to work together toward overarching project goals. Specific decisions about what functionalities and technologies to include in an assemblage are determined by a variety of factors, including project resources, practical concerns, the desire to build or buy, support for science, support for participants, motivation, and participant satisfaction.

Typically, functionalities—the various activities that an assemblage must support—are selected based on higher order concerns such as scientific goals or participant motivation. Technologies—the specific programming languages, databases, and software suites that support selected functionalities—are chosen for more pragmatic reasons such as cost, ability to support intended activities, availability, or how well they integrate with other assemblage components.

Support for scientific goals can sometimes be in conflict with support for participants, as the technologies to support each vary considerably and often require different perspectives on design. Nonetheless, both science and participation are critical factors for a citizen science project, and a successful system assemblage in this domain should be shaped to adequately support both.

Keeping motivation and satisfaction in mind during the design of the assemblage is one way to accomplish this, as well executed interfaces to support scientific goals are often motivating while well designed participant support features help to create a satisfying participant experience. Frequently, scientific goals and motivation are thought of as preeminent, but satisfaction oriented functionality is at least as prevalent on established projects. This suggests that those who intend to begin a citizen science project would do well to spend their time thinking through issues of participant satisfaction: high quality information architecture and navigational elements, well organized content, and careful consideration of participant privacy.

Ultimately, there is no ideal system assemblage for citizen science. Rather, individual projects will have different goals, different resources, and different needs. By contemplating a project's technological instantiation as an assemblage of interrelated functionalities, however, project scientists and developers may have a better sense of the effects of various functionalities and technologies, as well as ways to identify which to include and which to avoid.

ACKNOWLEDGMENTS

Omitted for review.

REFERENCES

1. Cohn, J.P., *Citizen Science: Can Volunteers Do Real Research?* BioScience, 2008. **58**(3): p. 192-107.
2. Wiggins, A. and K. Crowston. *From Conservation to Crowdsourcing: A Typology of Citizen Science*. in *44th Hawaii International Conference on System Sciences*. 2011. Kauai, Hawaii.
3. Kling, R., *Defining the boundaries of computing across complex organizations.*, in *Critical issues in information systems*, R.B.a.R. Hirschheim, Editor. 1987, Wiley: London.
4. Kling, R., *Computerization and Social Transformations.* Science, Technology & Human Values, 1991. **16**(3): p. 342-367.
5. Kling, R. and W. Scacchi, *The Web of Computing: Computer Technology as Social Organization*, in *Advances in Computers*. 1982, Elsevier. p. 1-90.
6. Zhang, P. and G.M. von Dran, *Satisfiers and dissatisfiers: A two-factor model for Website design and evaluation.* Journal of the American Society for Information Science and Technology, 2000. **51**(14): p. 1253.
7. Bos, N., et al., *From Shared Databases to Communities of Practice: A Taxonomy of Collaboratories.* Journal of Computer-Mediated Communication, 2007. **12**(2): p. 652-672.